أساسيات أمن الشبكات

تطبيقات ومعايير

أساسيات أمن الشبكات

تطبيقات ومعايير

تأليف

وليام ستولينج

نقله إلى العربية

الدكتور/ السيد محمد الألفي

الدكتور/ رضوان السعيد عبد العال

الدكتور/ علاء الدين أمين

كلية هندسة وعلوم الحاسب – جامعة الملك فهد للبترول والمعادن

العبيكان Obeikan

جامعة الملك فهد للبترول والمعادن
عمادة البحث العلمي

ح مكتبة العبيكان، 1432هـ

فهرسة مكتبة الملك فهد الوطنية أثناء النشر
ستولينج، وليام
أساسيات أمن الشبكات./ وليام ستولينج؛ رضوان السعيد عبدالعال.
- الرياض، 1432هـ
738ص؛ 16.5 × 24سم
ردمك: 978-603-503-157-8
1 - أمن المعلومات 2 - شبكات المعلومات
أ- عبدالعال، رضوان السعيد (مترجم) ب- العنوان
ديوي 005,8 1432/4960

رقم الإيداع: 1432/4960

تم دعم هذا المشروع من عمادة البحث العلمي بجامعة الملك فهد للبترول والمعادن تحت مشروع تأليف كتاب رقم ( ) بالتعاون مع شركة العبيكان للأبحاث والتطوير

الطبعة الأولى

1432هـ/2011م

حقوق الطباعة محفوظة للناشر

الناشر: العبيكان للنشر
الرياض - شارع العليا العام - جنوب برج المملكة
هاتف 2937574 / 2937581 فاكس 2937588
ص. ب 67622 الرمز 11517

التوزيع: مكتبة العبيكان
الرياض - العليا - تقاطع طريق الملك فهد مع العروبة
هاتف 4160018 /4654424 فاكس 4650129
ص. ب 62807 الرمز 11595

لا يسمح بإعادة إصدار هذا الكتاب أو نقله في أي شكل أو واسطة، سواء أكانت إلكترونية أو ميكانيكية، بما في ذلك التصوير بالنسخ «فوتوكوبي»، أو التسجيل، أو التخزين والاسترجاع، دون إذن خطي من الناشر.



تصدير

يُسعدنا أن نقدم للقارئ العربي ترجمة لواحد من أكثر الكتب انتشاراً في الجامعات العالمية في مجال أمن شبكات الحاسب ونظم المعلومات. وقد حرصنا على عدم حرفيَّة الترجمة مع الحفاظ على الدقة في النقل ووضوح المعنى. ويأتي هذا الجهد انطلاقاً من ثقتنا بأن اللغة العربية قادرة على مواكبة التقدم العلمي والتقني المستمر في شتى مجالات الحياة، وأنها لغة ثرية تمتلك الآليات التي تجعلها في مستوى غيرها من اللغات لتدريس المناهج العلمية والتقنية في الجامعات؛ كالإنجليزية وغيرها من اللغات الأوربية ولغات شرق أسيا. كما نأمل أن يكون هذا الكتاب إضافةً تُثري المكتبة العربية بواحد من الكتب المهمة في هذا المجال على المستوى الأكاديمي. وقد قمنا بتعريب المصطلحات اعتماداً على خبرتنا في هذا المجال وبالرجوع إلى مجامع اللغة العربية في العالم العربي وخاصة مجمع اللغة العربية بالقاهرة (معجم المصطلحات العلمية)، وكذلك إلى عدد من القواميس والمعاجم والموسوعات العلمية المتخصصة الأخرى (كقاموس المورد، ومعجم الكيلاني لمصطلحات الكمبيوتر والإنترنت، ودليل شعاع لمصطلحات الحاسب، ومعجم مصطلحات الكمبيوتر - الدار العربية للعلوم)، ومكتب تنسيق التعريب بالرباط (بنك المصطلحات الموحدة)، والبنك الآلي السعودي للمصطلحات العلمية (باسم)، والجمعية الدولية للمترجمين واللغويين العرب، وشبكة صوت العربية، والمركز العربي للتعريب والترجمة والنشر بدمشق (مجلة التعريب)، وغيرها من مواقع الإنترنت (انظر الملحق C). ومن الجدير بالذكر أنه رغم وجود كثير من الجدل حول تعريب العلوم وغرابة المصطلحات واختلافها من بيئة لأخرى فإن هذا لا يقلل من أهمية الجوانب الإيجابية للترجمة والتعريب.

والله نسأل أن يُيسر الإفادة من هذا الكتاب، وأن يكون حافزاً لكثير من المتخصصين للقيام بأدوارٍ مماثلة لتوطين العلوم الحديثة وتطبيقاتها في العالم العربي.

المترجمون

شكر وتقدير

من فضل الله علينا أن وفقنا لهذا العمل ويسره لنا، فالحمد الله الذي بنعمته تتم الصالحات، ونستغفره على أي تقصير كان منا.

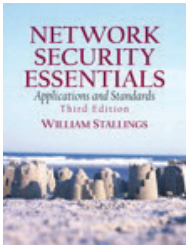
ونتوجه بخالص الشكر والعرفان لجامعة الملك فهد للبترول والمعادن بالمملكة العربية السعودية؛ لما أولته من دعم ملحوظ لهذا المشروع، ولما تؤديه من دور ريادي في تشجيع كل من يسهم في نهضة المجتمع وخدمته على الصعيد الإقليمي والعربي. كما نتوجه بالشكر لكل من تعاون معنا على إنجاز هذا العمل، ونخص بالذكر معالي مدير الجامعة، وعميد البحث العلمي، وعميد كلية علوم وهندسة الحاسب، ورئيس قسم علوم الحاسب والمعلومات، ورئيس قسم هندسة الحاسب، والزملاء من أعضاء هيئة التدريس بالكلية.

ونخص بالشكر الدكتور/ عصام عيد أبو غربية بقسم الدراسات الإسلامية والعربية بجامعة الملك فهد للبترول والمعادن لمراجعته الكتاب لغوياً، والشكر موصول للمحكمين الدوليين (سواءً لمقترح مشروع الترجمة أو لنسخة الكتاب النهائية) على تعليقاتهم البناءة وعلى إشادتهم بالدور الإيجابي للمترجمين في بساطة ووضوح العرض وتسلسل الأفكار.

كما نشكر دار النشر (John Wiley & Sons, Inc.) على موافقتها لنا على ترجمة الكتاب للغة العربية. ونشكر مكتبة العبيكان على نشرها هذا الكتاب وتوزيعه.

المترجمون

بيانات الكتاب المُترجم



- **عنوان الكتاب باللغة الإنجليزية:**

Network Security Essentials: Applications and Standards

- **المؤلف:** William Stallings
- **الطبعة:** الثالثة
- **تاريخ النشر:** 2007
- **الناشر:** Prentice Hall
- **الرقم الدولي:** ISBN-10: 0132380331، ISBN-13: 9780132380331

- **موقع الكتاب على شبكة الويب:**

http://williamstallings.com/NetSec/NetSec3e.html
http://www.pearsonhighered.com/stallings/details7.html

- **مجال التخصص:**

العام: هندسة وعلوم الحاسبات ونظم المعلومات.

الدقيق: أمن الشبكات ونظم المعلومات.

- **عنوان الكتاب باللغة العربية:**

أساسيات أمن الشبكات: تطبيقات ومعايير

نبذة عن المترجمين

لم يقتصر دور المترجمين على تحويل النص من اللغة الإنجليزية إلى اللغة العربية، ولكنهم اضطلعوا بدورٍ آخر – لكونهم متخصصين في مجال هندسة وعلوم الحاسب – وهو تنقيح المادة العلمية بما يتناسب مع السياق والبيئة العربية. وفيما يلي نبذة مختصرة عن مؤهلاتهم [وللمزيد من المعلومات يرجى الاطلاع على السيرة الذاتية لهم من خلال مواقعهم على شبكة الويب]:

الدكتور/ السيد محمد الألفي:

حصل على درجة الدكتوراه في هندسة الحاسبات (النظم الذكية وتطبيقاتها في شبكات الحاسب) من معهد ستيفنز للتكنولوجيا بالولايات المتحدة الأمريكية. وقام بالتدريس وبإعداد المذكرات لعدد من المناهج في مجال شبكات الحاسب وأمن الحاسبات والشبكات (باللغتين الإنجليزية والعربية). كما ألف عدداً من الأوراق البحثية، وشارك في العديد من اللجان الفنية لمؤتمرات علمية دولية في المجال ذاته؛ مثل: ICC، Globecom، WiMob، NSS، SECMCS، TrustCom، وغيرها. كما شارك بفعالية في تقييم وتحديث البرنامج الدراسي بقسم علوم الحاسب والمعلومات بجامعة الملك فهد للبترول والمعادن بالظهران وفي إعداد مناهج تفصيلية في مجال شبكات الحاسب في ضوء أحدث توجيهات للجنة الأمريكية (ABET/CSAB) لاعتماد البرامج الأكاديمية في مجال الحاسب الآلي. وشارك في إعداد البرنامج الأكاديمي لقسم علوم الحاسب بجامعة حائل بحائل، وفي مراجعة البرنامج الأكاديمي لكلية علوم وهندسة الحاسب بجامعة الأميرة نورة بنت عبد الرحمن بالرياض. وقد قام مع الباحث الثاني بترجمة كتاب " Computer Networking, 4/e, By J. Kurose and K. Ross" في مجلدين. كما حصل سيادته على كثير من الجوائز كان آخرها جائزة التميز في التدريس من جامعة الملك فهد عام 2011.

البريد الإلكتروني: alfy@kfupm.edu.sa

موقع الويب: http://faculty.kfupm.edu.sa/ics/alfy

الدكتور/ رضوان السعيد عبد العال:

حاصل على درجة الدكتوراه في الهندسة الكهربية من جامعة استراثكليد بالمملكة المتحدة. قام بتدريس هندسة اتصالات البيانات والحاسب بقسم هندسة الحاسب الآلي بجامعة الملك فهد للبترول والمعادن عدة مرات. وقبل ذلك – أثناء عمله بمعهد البحوث بالجامعة – كان مسؤولاً عن تحرير الجزء الخاص بإدارة مصادر الطاقة في التقرير السنوي لمعهد البحوث وترجمته إلى اللغة العربية لعدة سنوات. كما قام بترجمة عدد من الوثائق والنشرات الفنية ومستخلصات الأبحاث إلى اللغة العربية، وكذلك ترجمة عدة مقالات علمية؛ منها مقال عن الإنترنت لمجلة العلوم، وهي الترجمة العربية لمجلة ساينتفيك أميريكان الأمريكية وتصدرها مؤسسة الكويت للتقدم العلمي. وقد قام مع الباحث الأول بترجمة كتاب " Computer Networking, 4/e, By J. Kurose and K. Ross" في مجلدين.

البريد الإلكتروني: radwan@kfupm.edu.sa
موقع الويب: http://faculty.kfupm.edu.sa/coe/radwan

الدكتور/ علاء الدين أمين:

حاصل على درجة الدكتوراه في هندسة الحاسب من جامعة يوتاه بالولايات المتحدة الأمريكية ولديه خبرة أكاديمية تزيد عن العشرين عاماً بقسم هندسة الحاسب الآلي بجامعة الملك فهد للبترول والمعادن. وللدكتور أمين عدة أبحاث منشورة في مجلات علمية مُحكَّمة ومؤتمرات عالمية في مجالي تشفير المعلومات وأمن الحاسبات كما كان الباحث الرئيس لبحثٍ ممولٍ من مدينة الملك عبد العزيز للعلوم والتقنية – قد استكمل حديثاً – في نفس المجال. والدكتور أمين حائزٌ على أربع براءات اختراعات من مكاتب تسجيل الاختراعات بالولايات المتحدة الأمريكية وأوروبا وقد قدم حديثاً ثلاثة اختراعات بمجال التشفير وهي حالياً قيد الفحص والتدقيق بتلك المكاتب. وقد قام الدكتور أمين بترجمة بعض النشرات والكتيبات الخاصة بالتعريف بقسم هندسة الحاسب الآلي والمواد المُدَرَّسة به.

البريد الإلكتروني: amindin@kfupm.edu.sa
موقع الويب: http://faculty.kfupm.edu.sa/coe/amindin

جدول المحتويات

تمهيد XXI

الفصل الأول: مقدمة 1

1-1 اتجاهات الأمن 7

2-1 البنية الأمنية للربط بين الأنظمة المفتوحة 10

3-1 الهجمات الأمنية 11

4-1 خدمات الأمن 17

5-1 الآليات الأمنية 23

6-1 نموذج لأمن الشبكة 24

7-1 معايير وجمعية الإنترنت 27

8-1 الخطوط العريضة لهذا الكتاب 33

9-1 توصيات للمطالعة 34

10-1 مصادر للمعلومات على الويب 34

11-1 مصطلحات رئيسة 38

12-1 أسئلة للمراجعة ومسائل 39

الباب الأول: الكتابة المشفرة 41

الفصل الثاني: التشفير المتماثل وسريّة الرسائل 43

1-2 مبادئ التشفير المتماثل 46

2-2 خوارزميات التشفير المتماثل لكتل البيانات 55

3-2 تشفير تدفق البيانات وخوارزمية RC4 68

4-2 أنماط التشغيل بتشفير كتل البيانات 75

5-2 مواضع أجهزة التشفير 81

6-2 توزيع المفاتيح 84
7-2 توصيات للمطالعة 87
8-2 مصادر للمعلومات على الويب 87
9-2 مصطلحات رئيسة 88
10-2 أسئلة للمراجعة ومسائل 89

الفصل الثالث: الكتابة المشفّرة بالمفاتيح العامّة والتحقق من الرسالة 93
1-3 طرق توثيق الرسائل 95
2-3 دوال التحوير الآمنة وخوارزمية HMAC 103
3-3 مبادئ التشفير بالمفاتيح العامة 121
4-3 خوارزميات التشفير بالمفاتيح العامة 127
5-3 التوقيع الرقمي 140
6-3 إدارة المفاتيح 141
7-3 توصيات للمطالعة 145
8-3 مصادر للمعلومات على الويب 145
9-3 مصطلحات رئيسة 146
10-3 أسئلة للمراجعة ومسائل 147

الباب الثاني: تطبيقات أمن الشبكة 155

الفصل الرابع: تطبيقات التوثيق 157
1-4 نظام كيربيروس (Kerberos) 160
2-4 خدمة X.509 للتوثيق 189
3-4 البنية التحتية للمفاتيح العامة 206
4-4 توصيات للمطالعة 210
5-4 مصادر للمعلومات على الويب 211
6-4 مصطلحات رئيسة 212

7-4 أسئلة للمراجعة ومسائل 213
الملحق A-4 طرق التشفير في نظام كيربيروس 215

الفصل الخامس: أمن البريد الإلكتروني 219
1-5 بروتوكول الخصوصية الجيدة جداً (PGP) 222
2-5 امتدادات بريد الإنترنت متعددة الأغراض الآمنة (S/MIME) 257
3-5 مصادر للمعلومات عى الويب 286
4-5 مصطلحات رئيسة 287
5-5 أسئلة للمراجعة ومسائل 288
الملحق A-5 ضغط البيانات باستخدام خوارزمية ZIP 290
الملحق B-5 التحويل للأساس 64 (Radix-64) 295
الملحق C-5 توليد الأعداد العشوائية في PGP 299

الفصل السادس: أمن بروتوكول الإنترنت 305
1-6 لمحة عامة عن أمن بروتوكول الإنترنت 308
2-6 البنية الأمنية لبروتوكول الإنترنت (IP) 313
3-6 ترويسة التوثيق 323
4-6 التَّغليف الأمني للحمولة 331
5-6 الجمع بين الارتباطات الأمنية 340
6-6 إدارة المفاتيح 346
7-6 توصيات للمطالعة 363
8-6 مصادر للمعلومات على الويب 363
9-6 مصطلحات رئيسة 364
10-6 أسئلة للمراجعة ومسائل 365
الملحق A-6 التشبيك البيني وبروتوكول الإنترنت 367

الفصل السابع: أمن شبكة الويب 382

7-1 متطلبات أمن الويب 386

7-2 طبقة المقابس الآمنة وأمن طبقة النقل 391

7-3 المُعامَلات الإلكترونية الآمنة (SET) 422

7-4 توصيات للمطالعة 441

7-5 مصادر للمعلومات على الويب 441

7-6 مصطلحات رئيسة 442

7-7 أسئلة للمراجعة ومسائل 443

الفصل الثامن: أمن إدارة الشبكات 445

8-1 المفاهيم الأساسية لبروتوكول إدارة الشبكات البسيط (SNMP) 448

8-2 تسهيلات تجمعات SNMPv1 461

8-3 الإصدار الثالث لبروتوكول إدارة الشبكات البسيط 465

8-4 توصيات للمطالعة 506

8-5 مصادر للمعلومات على الويب 506

8-6 مصطلحات رئيسة 507

8-7 أسئلة للمراجعة ومسائل 508

الباب الثالث: أمن نظام الحاسب 513

الفصل التاسع: المتسللون 515

9-1 المتسللون 518

9-2 اكتشاف التسلل 525

9-3 إدارة كلمات السر 547

9-4 توصيات للمطالعة 564

9-5 مصادر للمعلومات على الويب 565

9-6 مصطلحات رئيسة 566

7-9 أسئلة للمراجعة ومسائل 567
الملحق A-9 مغالطة المعدل الأساسي 571

الفصل العاشر: البرمجيات الخبيثة 577
1-10 الفيروسات والتهديدات المتعلقة بها 579
2-10 الإجراءات المضادة للفيروسات 601
3-10 هجمات حجب الخدمة الموزَّعة 610
4-10 توصيات للمطالعة 620
5-10 مصادر للمعلومات على الويب 621
6-10 مصطلحات رئيسة 622
7-10 أسئلة للمراجعة ومسائل 623

الفصل الحادي عشر: الجدران النارية 625
1-11 مبادئ تصميم الجدران النارية 628
2-11 الأنظمة الموثوقة 649
3-11 المعايير المشتركة لتقييم أمن تقنية المعلومات 659
4-11 توصيات للمطالعة 667
5-11 مصادر للمعلومات على الويب 668
6-11 مصطلحات رئيسة 669
7-11 أسئلة للمراجعة ومسائل 670

الملاحق 673

الملحق A. بعض جوانب نظرية الأعداد 675
1-A الأعداد الأولية والأعداد الأولية نسبياً 677
2-A الحساب المقاسي 680

الملحق B. مشاريع لتعليم أمن الشبكة 683
B-1 مشاريع بحثية 385
B-2 مشاريع برمجية 686
B-3 تمارين مختبرية 687
B-4 واجبات كتابية 688
B-5 واجبات حول القراءة وكتابة التقارير 688

الملحق C. وقفة مع ترجمة وتعريب المصطلحات 689

ثبت المراجع 695

مسرد المصطلحات 707

اختصارات 715

تمهيد

برزت في الآونة الأخيرة شبكات الحاسب الآلي – وبخاصة الإنترنت – وأحدثت تغييرات جوهرية على مستوى الأفراد والمجتمعات، وزاد اعتماد الأفراد والشركات والجامعات والهيئات الحكومية وغير الحكومية على الخدمات التي تقدمها الشبكات كخدمة الاتصال عن طريق البريد الإلكتروني والاتصال الآني والمحادثة الفورية بين الأفراد، كما غيرت طريقة نشر المعلومات وتبادلها. ومع التطور المستمر استُحدِثت كثير من التطبيقات لزيادة الاستفادة من الشبكة؛ كالتعليم عن بعد، والحكومة الإلكترونية، والتسوق الإلكتروني، والمؤتمرات عبر الشبكة، وخدمات الترفية والألعاب من خلال الشبكة.

ومع هذا التطور السريع والنمو الهائل الملحوظ زادت المخاطر التي تتعرض لها المعلومات والأنظمة المتصلة بالشبكة، مما جعل قضية تأمين الشبكات وحماية تلك الأنظمة موضوعاً جوهرياً، وأصبحت الحاجة ماسةً إليه في كل وقت. ومن تلك المخاطر زيادة انتشار فيروسات الحاسب عبر الشبكة، وقرصنة البيانات الحساسة والبرامج، والتنصت الإلكتروني، والتزوير والاحتيال بالوسائل الإلكترونية، ورسائل الدعاية التي لا طائل منها للمُستقبِل. ونظراً لاعتماد تلك الأنظمة المرتبطة عبر الشبكة على بعضها البعض بشكل كبير، فإن إصابة أي جزء يؤدي إلى تقليل فاعلية النظام وقد يؤدي إلى شلل وتوقف تام للنظام بكامله. وقد نضجت مجالات أمن الشبكات والكتابة المشفّرة بشكل أدّى إلى تطوير تطبيقات عملية فرضت وجود معايير أمنية لحماية الشبكة.

لذا يأتي هذا الكتاب ليعطي مقدمة واضحة وشاملة لمبادئ أمن شبكات الحاسب والإنترنت وتطبيقاتها ومعاييرها بطريقة عملية ومعاصرة. كما يتضمن موجزاً عن الكتابة المشفّرة يشمل الخوارزميات والبروتوكولات التي تستند عليها تطبيقات أمن الشبكة وتشفير البيانات والتوقيعات الرقمية وتبادل مفاتيح الأمن.

ومؤلف الكتاب (الدكتور وليام ستولينج) ممارس ومعلّم معروف عالميّاً بإسهاماته الفعالة في مجال علوم الحاسب. فقد قام بتأليف أكثر من 17 كتاباً مختلفاً في مجالات البنية المعمارية للحاسب ونظم تشغيله واتصالات البيانات وشبكات الحاسب والإنترنت والكتابة المشفرة وأمن الحاسبات وأمن الشبكات، إضافةً إلى عدد من الأوراق البحثية. وهو حاصل على درجة الدكتوراه من معهد MIT في مجال علوم الحاسب وعلى درجة البكالوريوس من جامعة Notre Dame في مجال الهندسة الكهربية. وقد مكنته خبرته الطويلة لأكثر من 20 سنة في المجال الأكاديمي والصناعي كمدير واستشاري لعدد من شركات تطوير البرمجيات والمعاهد البحثية الحكومية من تقديم المادة العلمية بأسلوب شائق من خلال عدد من الأمثلة الواقعية مما يُساعد الطالب أو الممارس على فهم الصلة بين نظرية الأمن وتحديات الأمن اليومية في نظم الشبكات المختلفة. وقد فازت كتبٌ له عشر مرات بجائزة "أحسن كتاب أكاديمي"، ومن بينها هذا الكتاب.

وينقسم الكتاب إلى أحد عشر فصلاً بالإضافة إلى عدد من الملاحق ومسرد مصطلحات وثبت للمراجع، وتعدُّ الطبعة الثالثة آخر ما صدر من الكتاب وهى طبعة منقحة تعكس التطورات الحديثة في مجال أمن شبكات الحاسب. وللكتاب موقع على شبكة الويب يشتمل على عدد من المواد التعليمية، والأشكال التوضيحية، وعروض الـ PowerPoint للمحاضرات، وقائمة بالمصادر ومواقع الويب الإضافية، وكتيب مختبرات ومشروعات بحثية، ودليل حلول المسائل للمعلم.

ويُعد هذا الكتاب من الكتب الأكاديمية الأكثر انتشاراً في مجال أمن شبكات الحاسب والإنترنت، حيث يُستخدم في كثير من الكليات والجامعات العالمية كمرجع أساسي أو موصى به، من بينها:

- USA: Georgia Institute of Technology (Georgia Tech), School of Electrical and Computer Engineering, ECE 6612 Computer Network Security, Spring 2008:

  http://www.csc.gatech.edu/copeland/jac/6612/

- USA: INDIANA WESLEYAN UNIVERSITY, Computer and Information Sciences, CIS396 - Network Security, Spring 2007:

http://www.indwes.edu/faculty/zhu/profession/CLASSES/IWUCIS396-S07/SyllabusCIS396s07.pdf

- USA: UC Santa Cruz, School of Engineering, Computer Science, CMPS 122: Computer Security:

  http://www.cs.ucsc.edu/~elm/Classes/122/s03/syllabus.shtml

- USA: California State University, Department of Mathematics and Computer Science, CS 4525 - Principles of Network Security:

  http://www.sci.csuhayward.edu/mathcs/schedules/w09_cs.doc

- USA: University of Colorado at Colorado Springs, College of Engineering & Applied Science, CS 591 Fundamentals of Computer and Network Security:

  http://cs.uccs.edu/~cs591/

- USA: The University of Texas at Austin, Department of Computer Sciences, CS 378 - Network Security and Privacy:

  http://www.cs.utexas.edu/~shmat/courses/cs378_spring05/cs378_ref.html

- USA: George Mason University, ISA 562: Information Security Theory and Practice,

- USA: Florida State University, Computer Science, CIS 4362 - Network Security and Cryptography, http://www.cs.fsu.edu/ugcc/textbooks.html

- USA: The University of Alabama at Birmingham, Computer and Information Sciences, CS 336/536 Computer Network Security:

  http://www.cis.uab.edu/info/faculty/barnard/cs437/schedule.html

- USA: University of Houston, School of Science & Computer Engineering (SCE), CSCI 5931 Research Topic: Network Security:

  http://nas.cl.uh.edu/yang/teaching/csci5931netSecuritySpr05/index.html

- USA: University of Hawaii, Dept. of Electrical and Computer Engineering, EE491F Introduction to Computer and Network Security:

  http://www-ee.eng.hawaii.edu/~dong/EE491/Syllabus_EE491_Computer_Security.htm

- USA: Long Island University, Brooklyn Campus, Dept. of Computer Science, CS 678 – Data Security:

http://myweb.brooklyn.liu.edu/pchung/CS678_Syllabus_F06.html

- USA: Pace University, Seidenberg School of Computer Science and Information Systems. IT660 - Network Security:

  http://csis.pace.edu/~lixin/teaching/it660/

- USA: Polytechnic University, Department of Computer and Information Science, CS 393/682 – Network Security:

  http://cis.poly.edu/~ross/networksecurity/Overview.ppt

- USA: DAKOTA STATE UNIVERSITY, INFS 754 - Network Security and Intrusion Detection, Fall 2008:

  http://www.departments.dsu.edu/disted/courses/syllabus/fall2008/INFS754.htm

- USA: Sacramento State University, College of Engineering and Computer Science, CSC 254 - Network Security:

  http://www.ecs.csus.edu/csc/iac/csc296q/CSc296Q-num-for%20map.doc

- USA: Hofstra University, NY, Department of Computer Science, CSC290A - Network Security:

  http://www.cs.hofstra.edu/~cscvjc/Spring06/

- USA: Mills College, Oakland, California, Interdisciplinary Computer Science MCS 180(280) - Network Security: http://cs.mills.edu/180-security/

- USA: St. Joseph's College, Department of Mathematics/Computer Science, NJ, COM 260 COMPUTER AND NETWORK SECURITY:

  http://faculty.sjcny.edu/~fritz/COM260table.htm

- Canada: UNIVERSITY OF MANITOBA, Dept. of Computer Science, COMP 4580 - Computer Security:

  http://www.cs.umanitoba.ca/newsite/coursedescrip/comp4580.pdf

- Canada: University of Alberta, Dept. of Computer Science, CMPUT 299 Security in a Networked World:

  http://ugweb.cs.ualberta.ca/~c299/W08/B1/schedule/index.html

- Canada: University of British Columbia, Institute for Computing, Information, and Cognitive Systems, Master of Software Systems, CICS 518 - Computer and Information System Security:

  http://www.cs.ubc.ca/local/reading/books/reservearchive.pdf

- Canada: Carleton University, School of Computer Science, COMP 4108 - Computer and Systems Security, Fall 2008:

  http://www.scs.carleton.ca/~kranakis/

- Canada: Simon Fraser University, ITEC 413 Computer Security: Computer Security: Fundamentals, ITEC 414 Computer Security: Computer Security: Practice, ITEC 415 Computer Security: Distributed Systems Security

  http://www.siat.sfu.ca/resources/courses/ITEC413_414_415.pdf

- Canada: University of Windsor, Electrical and Computer Engineering, ECE88-590-18 - Network Security:

  http://athena.uwindsor.ca/users/e/erfani/main.nsf/9d019077a3c4f6768525698a00593654/db15005e011839a985256db900525365!OpenDocument

- UK: The University of Edinburgh, CS 0102- Computer Security:

  http://www.inf.ed.ac.uk/teaching/courses/cs/0102/lecoutline/sec-refs.pdf

- Ireland: Waterford Institute of Technology, Department of Computing, Mathematics & Physics, IT 4 - IT Security:

  http://emhain.wit.ie/~mcgibney/it4/

- India: West Bengal University of Technology, Computer Science and Engineering, CS 802D - Network Security:

  http://www.wbut.ac.in/syllabus/Computer_Science_&_Engineering_Syllabus.pdf

- India: SREE VIDYANIKETHAN Engineering College, Computer Science and Engineering, (CS 05317) INFORMATION SECURITY:

  http://www.vidyanikethan.edu/inside_2/syllabus/u-cse.pdf

- India: ANNA UNIVERSITY TIRUCHIRAPPALLI, CO5010 COMMUNICATION NETWORK SECURITY:

  http://www.tau.org.in/download/pg_syllabus/comp_commn.pdf

- Saudi Arabia: King Abdulaziz University, CS483 - Network Security and Applications:

  http://www.kau.edu.sa/obamasek/MySiteKAU/CS483.htm

- Saudi Arabia: Prince Mohammed Bin Fahd University, ITAP 3431: Network Security, http://www.pmu.edu.sa/enn/downloads%5CTIEC%5CPreparation%20Year%5CUndergraduate%20IT%20Programs%5CI%20Undergraduate%20IT%20Syllabi-FINAL%203%20of%205.pdf

- Saudi Arabia: King Saud University, ITC 430 Information Security: http://faculty.ksu.edu.sa/alwakeel/Document%20Library/IT%20Girls%20%20Department%20,%20Study%20plan%20,college%20of%20Comuter%20King%20Saud%20Univeersity.pdf

❖ أهمية الكتاب

تُعدُّ النسخة الإنجليزية لهذا الكتاب من أفضل الكتب الأكاديمية التي نشرت في مجال أمن الشبكات ومن أحدثها. ويُستخدَم الكتاب في كثير من جامعات العالم لتعليم طلاب البكالوريوس، كما يُستخدَم كمرجع أساسي لطلاب الدراسات العليا ولعدد من الباحثين والعاملين في حقل الشبكات خصوصاً المهتمين بمجال أمن الحاسبات والشبكات ونظم المعلومات.

وتكمن أهمية الكتاب المُترجَم في سد ثغرةٍ بالمكتبة العربية في مجال الكتب الدراسية للتعليم الجامعي. ويؤكد هذا المبادرة التي شرعت فيها وزارة التعليم العالي بالمملكة العربية السعودية مشكورةً لترجمة الكتب الأجنبية المستخدمة في تدريس العلوم الهندسية؛ مما يدل على الحاجة الماسة لمثل هذا الدور لإثراء المكتبة العربية بالمراجع في العلوم التقنية وإيجاد عدد من البدائل عند اختيار كتاب دراسي. وهناك أدوار مماثلة في بعض البلدان العربية حالياً كسوريا على سبيل المثال.

❖ مدى الإفادة من الكتاب

إن نشر مرجع أكاديمي مثل هذا الكتاب يمثل إضافة حقيقية للمكتبة العربية، ويؤكد أهمية الدور الريادي للجامعات في خدمة المجتمع؛ حيث يساعد ذلك على دعم العملية التعليمية في الجامعات العربية (التي تستخدم اللغة العربية في عملية التدريس)، وفي كليات المجتمع، وكليات المعلمين، وكليات البنات، والكليات التقنية، وكليات علوم الحاسب والمعلومات، وكليات الاتصالات والمعلومات ... إلخ. كما يساعد على دعم البحث العلمي، وتنشيط حركة الترجمة، ومواكبة التقدم العلمي في واحد من المجالات الحيوية ألا وهو مجال أمن الحاسبات والشبكات ونظم المعلومات.

❖ نوعية القراء

هذا الكتاب من الكتب الأكاديمية الشاملة في مجال أمن الحاسبات والشبكات ونظم المعلومات التي يمكن أن يستفيد منها فئات متعددة من القراء، وبخاصة في المجال الأكاديمي والبحثي؛ كطلاب البكالوريوس والدراسات العليا، والباحثين والمحترفين في مجال أمن الحاسبات والشبكات ونظم المعلومات. كما يمكن أن تستفيد منه فئات أخرى من القراء تضم: المهتمين بنقل وتوطين التقنيات الحديثة للعالم العربي، والمثقفين الراغبين في التعلم والتثقيف الذاتي في مجال تقنية وأمن المعلومات.

1

الفصل الأول

مقــدمة

محتويات الفصل:

1-1 اتجاهات الأمن

2-1 البنية الأمنية للربط بين الأنظمة المفتوحة

3-1 الهجمات الأمنية

4-1 خدمات الأمن

5-1 الآليات الأمنية

6-1 نموذج لأمن الشبكة

7-1 معايير وجمعية الإنترنت

8-1 الخطوط العريضة لهذا الكتاب

9-1 توصيات للمطالعة

10-1 مصادر للمعلومات على الويب

11-1 مصطلحات رئيسة

12-1 أسئلة للمراجعة ومسائل

" ..... إن عوامل الزمان والمكان والقوّة مجتمعة – تلك التي تعتبر العناصر الأساسية لنظرية الدفاع – تجعل عملية وضع خطة للدفاع مسألة معقدة إلى حدٍّ ما؛ مما يجعل من الصعب تحديد نقطة ثابتة للانطلاق." – من كتاب "عن الحرب" ، لكارل فون كلوز فيتز.

" ..... يعلمنا فَنُّ الحرب الاعتماد على جاهزيتنا الخاصة لملاقاة العدو وعلى مدى مناعة مواقعنا وليس على احتمال عدم قيامه بالهجوم علينا." – من كتاب "فن الحرب" ، لصن تزو.

شهدت العقود الأخيرة تغيّرين أساسيين في متطلبات تأمين المعلومات داخل منظمة ما. فقبل الانتشار الواسع لأدوات معالجة البيانات، كان توفير الأمن المطلوب للمعلومات يتم عن طريق استخدام أوعية حفظ آمنة (كخزانات الملفات المزودة بأقفال محكمة إحكاماً جيداً لحفظ الوثائق الحساسة) ومن خلال طرق إدارية كالتدقيق في اختيار الموظفين الجدد.

لكن مع الاستخدام المتزايد للحاسبات ظهرت الحاجة لطرق آلية لحماية الملفات والمعلومات الأخرى المخزنة على الحاسب؛ خاصةً في النظم المشتركة كنظم المشاركة في الوقت (time sharing). ازدادت الحاجة لتلك الطرق عندما أصبح النفاذ إلى تلك النظم ممكناً عبر شبكات الهاتف أو شبكات البيانات أو شبكة الإنترنت. ويُطلق مصطلح "أمن الحاسب" (computer security) بشكلٍ عام على مجموعة الأدوات المصمَّمة لحماية البيانات وإحباط عمل قراصنة الحاسب.

وتَمثَّل التغير الرئيس الثاني في ظهور الأنظمة الموزَّعة واستخدام الشبكات ووسائل الاتصالات لنقل البيانات بين مستخدم وحاسب أو بين حاسب وآخر. مما استدعى اتخاذ تدابير أمنية لحماية البيانات في أثناء انتقالها عبر الشبكات؛ وهو ما عُرف بـ "أمن الشبكات" (network security).

وفي الحقيقة يُعدُّ مصطلح "أمن الشبكات" غير دقيق إلى حدٍّ ما، فكل المؤسسات التجارية والحكومية والأكاديمية تقريباً تربط أجهزة معالجة البيانات

لديها بمجموعة من الشبكات المرتبطة معاً وتُشكِّل فيما بينها ما يُعرف بالشبكة البينية [1]. ولذا فالتعبير الأكثر دقة هنا هو "أمن الشبكة البينية" (internet security).

لا توجد في الواقع حدود فاصلة واضحة بين هذين الشكلين من أشكال الأمن: أمن الحاسب وأمن الشبكة البينية. فمثلاً: يُعدُّ فيروس الحاسب من أكثر أنواع الهجمات المنتشرة على أنظمة المعلومات، حيث يمكن إدخال الفيروس للنظام إما مادياً عن طريق قرص مرن أو قرص ليزر ملوث ومن ثم تحميله على الحاسب أو عن طريق الاتصال عبر الشبكة البينية. وفي كلتا الحالتين يتعين وجود أدوات أمن داخلية على الحاسب لاكتشاف الفيروس والتخلص منه بمجرد دخوله إلى نظام الحاسب.

يركِّز هذا الكتاب على أمن الشبكات البينية الذي يتألَّف من إجراءات لإعاقة ومنع وتجنب واكتشاف وتصحيح الانتهاكات الأمنية التي تؤثر على نقل المعلومات. ويُعدُّ هذا تعريفاً واسعاً يغطي مجموعة كبيرة من الاحتمالات الممكنة.

وحتى تتخيل الموضوعات التي سيتناولها هذا الكتاب سنذكر هنا أمثلة لبعض الانتهاكات الأمنية:

1. قد يرسل المستخدم A إلى المستخدم B ملفاً يحتوى على معلومات حساسة (كسجلات قائمة الرواتب) يجب المحافظة على سريتها، في حين يقوم مستخدم ثالث C – غير مخولٍ له قراءتها – بمراقبة عملية الإرسال والتقاط نسخة منها في أثناء انتقالها.
2. قد يرسل مدير شبكة D إلى الحاسب E الواقع تحت إدارته رسالة يطالبه فيها بتحديث ملف الترخيص (authorization file) لديه بإضافة المعلومات الخاصة بعدد من المستخدمين الجدد للسماح لهم بالدخول (الولوج) إلى ذلك الحاسب.

---

[1] لاحظ أننا استخدمنا هنا المصطلح internet الذي يبدأ بحرف صغير ليعني "الشبكة البينية" (أي مجموعة من الشبكات المرتبطة ببعضها)؛ وهو خلاف المصطلح Internet الذي يبدأ بحرف كبير (أي الشبكة الدولية والمعروفة بالإنترنت أو الشابكة التي يمكن أن تستخدمها منظمة ما كأحد الوسائل في ربط شبكاتها المختلفة).

ويقوم المستخدم F باعتراض الرسالة، ويعدّل محتوياتها بالإضافة أو الحذف، ثم يمرر الرسالة إلى E الذي يعتبرها بشكلٍ خاطئ قادمة من مدير الشبكة D ويقوم بتحديث ملف الترخيص وفقاً لذلك.

3. بدلاً من اعتراض الرسالة في الحالة السابقة، قد يقوم المستخدم F بإنشاء رسالته الخاصة بالمُدخلات المرغوبة لديه وإرسالها إلى الحاسب E كما لو كانت مرسلة من المدير D، وبناءً على ذلك يقوم E بتحديث ملف الترخيص.
4. قد يتم طرد موظف من عمله بدون سابق إنذار، ويقوم مدير شئون الموظفين بإرسال رسالة إلى أحد أنظمة الخادمات (servers) لإلغاء حساب الموظف (account) مع وضع ملاحظة على ملف الموظف كتأكيد للإلغاء. ويقوم الموظف المطرود باعتراض الرسالة وتأخيرها بدرجة كافية ليقوم بالولوج إلى الخادم للمرة الأخيرة لاسترجاع معلومات حساسة، وبعدها يقوم بتمرير الرسالة إلى الخادم لإتمام المهمة وتأكيد الإلغاء. يمكن أن يمر مثل هذا التصرف من الموظف المطرود دون اكتشاف لبعض الوقت.
5. قد يقوم أحد المستثمرين بإرسال رسالة إلى وكيله تتضمن تعليمات للقيام بعدة صفقات بسوق الأسهم وبعدها تنخفض قيمة تلك الأسهم ويُنكر المستثمر إرسال تلك الرسالة.

ورغم أن الأمثلة السابقة لا تغطي كافة الأنواع المحتملة لانتهاكات أمن المعلومات، فإنها توضح بعض مجالات الخطر التي تقع في دائرة اهتمام أمن الشبكات. إن توفير الأمن بين الشبكات أمرٌ معقدٌ وشائقٌ في نفس الوقت؛ ويرجع ذلك لعدة أسباب منها:

1. أن الأمن المتعلق بالاتصالات ليس بالبساطة التي قد تبدو للمبتدئين للوهلة الأولى. فرغم أن معظم متطلبات الأمن الرئيسة واضحة ويمكن تلخيصها في كلمات بسيطة تفسر نفسها بنفسها – كالسِّرية (confidentiality)، والتوثيق (authentication)، وعدم التنصل (non-repudiation) وسلامة البيانات (data integrity) – فإن الآليات المستخدمة لتلبية تلك المتطلبات يمكن أن تكون شديدة التعقيد، ويحتاج فهمها إلى تفكير دقيق ومنطقي.

2. عند تطوير آلية أو خوارزمية أمنية معينة ينبغي أن نأخذ في الاعتبار دائماً الهجمات المحتملة على العناصر الأمنية المذكورة أعلاه. وجدير بالذكر أن كثيراً من الهجمات الأمنية الناجحة صُمِّمت بالنظر إلى المسألة بطريقة مختلفة تماماً، حيث يتم استغلال نقاط ضعف غير متوقعة في آليات الحماية الأمنية نفسها.
3. نتيجة للنقطة السابقة فإن الإجراءات الواجب استخدامها لتوفير خدمات معينة غالباً ما تكون غير متوقعة بداهةً؛ ففي أغلب الأحيان لا تكون تفاصيل مثل تلك الإجراءات المحكمة المطلوبة واضحةً في البداية عند استعراض احتياجات أمنية معينة، ولكن تتضح تلك الإجراءات وتبدو معقولة فقط بعد أخذ مختلف الإجراءات المضادة بعين الاعتبار.
4. بعد تصميم آليات أمنية مختلفة، من الضروري تحديد مكان استخدامها، سواء من حيث موقعها المادي (مثلاً: في أية نقاط الشبكة تضاف تلك الآليات) أو من حيث موقعها المنطقي (مثلاً: في أية طبقة أو طبقات من بنية الشبكة؛ كبنية TCP/IP للإنترنت).
5. تتضمن آليات الأمن عادةً أكثر من مجرد بروتوكول أو خوارزمية محددة، فهي عادةً ما تتطلب أن يكون بحوزة المشتركين بعض المعلومات السرية (كمفاتيح التشفير)، الأمر الذي يثير تساؤلات حول عمليات تحديد تلك المعلومات السرية وتوزيعها وحمايتها. هناك أيضاً اعتماد على بروتوكولات الاتصالات والتي يمكن أن تعقّد مهمة تطوير الآلية الأمنية. فعلى سبيل المثال: لو كان التوظيف الصحيح للآلية الأمنية يتطلب وضع حدود على زمن انتقال رسالة من مُرسِل إلى مُستقبِل، فإن أي بروتوكول أو شبكة يؤدي إلى تأخيرات متغيرة غير متوقعة قد يُفقد تلك الحدود الزمنية أهميتها.

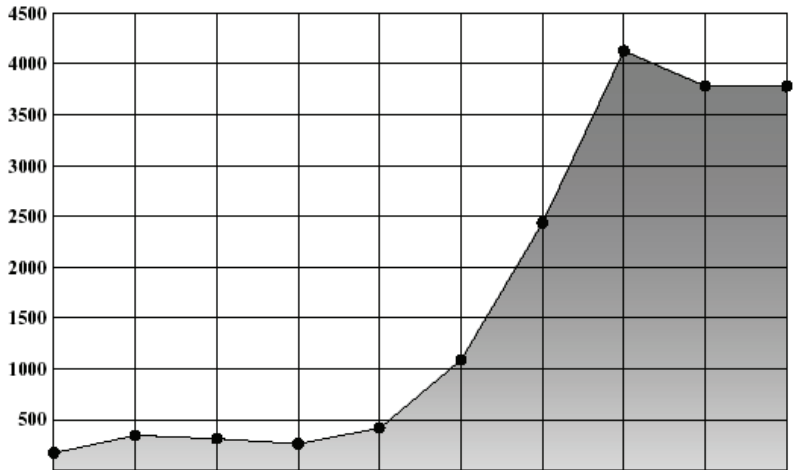
وهكذا فهناك كثيرٌ مما ينبغي أخذه في الاعتبار. في هذا الفصل سنلقي نظرة عامة على الموضوعات التي ستشكّل محتويات بقية الكتاب. سنبدأ باستعراض عام لخدمات أمن الشبكات وآلياتها، وباستعراض أنواع الهجمات التي صُمِّمت من

أجلها تلك الخدمات والآليات. ثم سنطوِّر نموذجاً عاماً يمكن من خلاله دراسة تلك الخدمات والآليات.

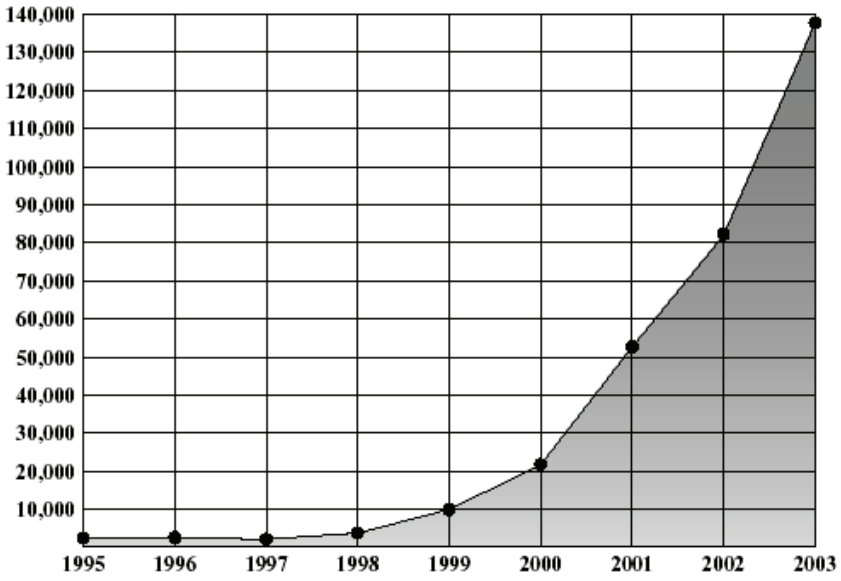
1-1 اتجاهات الأمن

أصدر مجلس البنية المعمارية للإنترنت ((Internet Architecture Board (IAB في عام 1994 تقريراً بعنوان "الأمن في بنية الإنترنت" (RFC 1636) يفيد بأن الإنترنت بحاجة إلى تأمين أكثر وأفضل. كما عرَّف التقرير المجالات الرئيسة لآليات الأمن، ومن بينها الحاجة إلى تأمين البنية التحتية للشبكة من العمليات غير المرخص لها كمراقبة حركة مرور البيانات والتحكم فيها، والحاجة إلى تأمين حركة مرور البيانات من مستخدم إلى مستخدم آخر باستخدام آليات التوثيق والتشفير.

هذه المخاوف لها ما يبررها تماماً. وتأكيداً على ذلك، خذ بعين الاعتبار الاتجاهات التي أفاد بها مركز التنسيق لفريق الاستجابة لطوارئ الحاسب (Computer Emergency Response Team/Coordination Center (CERT/CC)). يبين الشكل 1-1 (a) اتجاه مواطن الضعف المتعلقة بالإنترنت التي تم إبلاغها إلى CERT على مدى عشر سنوات. وهي تتضمن نقاط ضعف في أمان نظم تشغيل الحاسبات (مثل ويندوز ولينكس) ، وكذلك نقاط ضعف في موجّهات الإنترنت وغيرها من أجهزة الشبكة. كما يظهر في الشكل 1-1 (b) عدد الحوادث ذات الطابع الأمني المبلَّغة إلى CERT. وهي تتضمن هجمات حجب الخدمة (DoS)، وهجمات تزييف عناوين IP في رزم البيانات التي ينشئها المتسللون للتحايل على التطبيقات التي تستخدم توثيقاً مبنياً على فحص عناوين IP، بالإضافة إلى أشكال مختلفة من التنصت والتقاط الرزم من الشبكة (packet sniffing) التي تُمكِّن المهاجمين من قراءة البيانات المُرسَلة بما فيها معلومات الولوج (login) ومحتويات قاعدة البيانات.



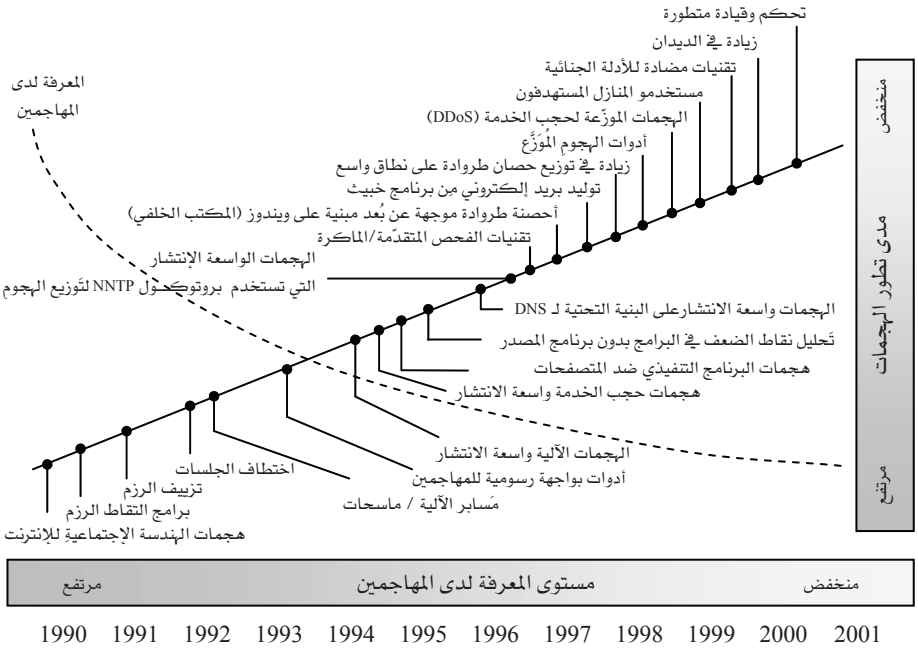
(a) الثغرات الأمنية المُبلَّغ عنها.



(b) الحوادث المُبلَّغ عنها.

الشكل 1-1: إحصاءات CERT.

ومع مرور الوقت نمت الهجمات على شبكة الإنترنت والأنظمة المتصلة بها وأصبحت أكثر تعقيداً، بينما انخفض مقدار المهارات والمعارف اللازمة لشن هجوم (الشكل 1-2). كما أصبح من الممكن أن تتم الهجمات بشكلٍ آلي مع التسبب في كمٍّ أكبر من الضرر. وتتزامن تلك الزيادة في الهجمات مع التزايد الملحوظ في استخدام الإنترنت وزيادة التعقيد في البروتوكولات والتطبيقات وفي الإنترنت نفسها. تعتمد البنى التحتية الحرجة في عملياتها بشكلٍ متزايد على الإنترنت، كما يعتمد المستخدمون الأفراد على أمن الإنترنت والبريد الإلكتروني والويب وتطبيقاته على نطاق واسع أكثر من أي وقت مضى، ولذا هناك حاجة ماسَّة إلى تشكيلة كبيرة من التقنيات والأدوات لمواجهة تلك التهديدات المتزايد. على المستوى الأساسي، تحظى خوارزميات التشفير للسرية (confidentiality) والتوثيق (authentication) بأهمية متزايدة. يحتاج المصممون أيضاً إلى التركيز على البروتوكولات المبنية على الإنترنت وعلى نقاط ضعف نظم التشغيل والتطبيقات الملحقة بها. يقدم هذا الكتاب مسحاً عاماً لكل تلك الموضوعات الفنية.



الشكل 1-2: اتجاهات تطور الهجمات والمعرفة لدى المهاجمين (المصدر: CERT)

2-1 البنية الأمنية للربط بين الأنظمة المفتوحة

يحتاج المدير المسؤول عن أمن مؤسسةٍ ما إلى طريقة منظمة لتحديد احتياجاتها الأمنية بشكلٍ فعَّال وتوصيف الحلول المقترحة لتلبيتها وتقييم المنتجات والسياسات الأمنية المختلفة واختيارها. وهذا أمر ليس بالسهل في بيئة مركزية لمعالجة البيانات؛ فما بالك عند استخدام الشبكات المحلية والشبكات واسعة النطاق (بالطبع ستتضاعف الصعوبة!).

للأسباب السابقة قام الاتحاد الدولي للاتصالات ITU باقتراح توصية رقم X.800 بعنوان "بنية أمنية للربط بين الأنظمة المفتوحة" (Security Architecture for OSI) لتحديد مثل تلك الطريقة المنظمة. ويُعدُّ النموذج المُقترح مفيداً للمديرين كطريقة لتنظيم مهمة توفير الأمن. وعلاوة على ذلك، ونظراً لأن هذه البنية تم تطويرها كمعيار دولي قياسي، فقد طوَّرت شركات الحاسب والاتصالات سمات أمنية لمنتجاتها وخدماتها لتتوافق مع ذلك التعريف المنظم للخدمات والآليات الأمنية.

ما يهمنا في هذا النموذج للبنية الأمنية للربط بين الأنظمة المفتوحة هو أنه يستعرض عدداً من المفاهيم التي يتناولها هذا الكتاب. فهو يركز على: الهجمات الأمنية، والآليات، والخدمات؛ التي يمكن تعريفها باختصار على النحو الآتي:

- الهجوم الأمني وهو أي فعل يُعرِّض أمن المعلومات الخاصة بمنظمةٍ ما للخطر.
- الآلية الأمنية وهي آلية مُصمَّمة لاكتشاف هجوم أمني أو منعه أو التعافي منه.
- الخدمة الأمنية وهي خدمة لتعزيز أمن أنظمة معالجة البيانات وانتقال المعلومات الخاصة بمنظمة ما. وتهدف مثل تلك الخدمات إلى مقاومة الهجمات الأمنية، وللقيام بذلك فإنها تستخدم واحدة أو أكثر من الآليات الأمنية.

يشيع في أدبيات الأمن استخدام مصطلحين بنفس المعنى تقريباً: التهديد والهجوم. ويبين الجدول 1-1 تعريفاً لكلٍّ منهما مقتبساً من مسرد مصطلحات أمن الإنترنت [RFC 2828].

الجدول 1-1: التهديدات والهجمات [RFC 2828].

التهديد	انتهاك محتمل للأمن، يحدث عند توافر ظرف ما أو قابلية معينة أو عند حدوث حدث يمكن أن يخرق الأمن ويسبب الأذى. إذن فالتهديد خطر محتمل يستغل نقاط ضعف معينة.
الهجوم	اعتداء على أمن النظام ينشأ من تهديد ذكي (بمعنى فعل مدروس ومتعمد لتجاوز الخدمات الأمنية وانتهاك السياسة الأمنية للنظام).

1-3 الهجمات الأمنية

من الطرق المفيدة لتصنيف الهجمات الأمنية – والمستخدمة في كلٍّ من X.800 وRFC 2882 – تقسيم تلك الهجمات إلى هجمات سلبية (passive attack) وهجمات نشطة أو فاعلة (active attacks). تحاول الهجمات السلبية (وهي هجمات مستترة يصعب اكتشافها) الحصول على معلومات من النظام ولكنها لا تؤثر على موارد النظام. أما الهجمات النشطة فتحاول الاعتداء لتعديل موارد النظام أو التأثير على عملياته.

1-3-1 الهجمات السلبية

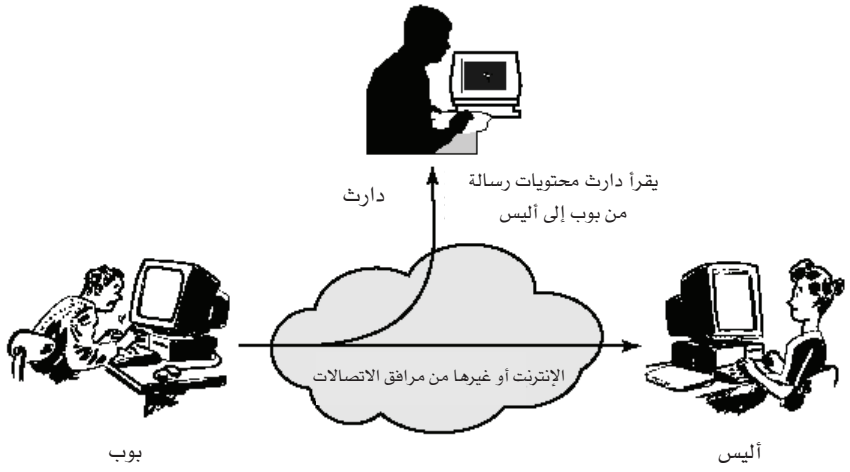
الهجمات السلبية هي في الحقيقة تنصت على إرسال البيانات أو مراقبتها بهدف الحصول على المعلومات التي يتم نقلها. من بين أنواع الهجمات السلبية: هجوم كشف محتوى رسالة وهجوم تحليل حركة مرور البيانات.

يمكن فهم هجوم "كشف محتوى رسالة" بسهولة من (الشكل 1-3 (a)). فأي مكالمة تليفونية، أو رسالة بريد إلكتروني، أو ملف مُرسَل يمكن أن

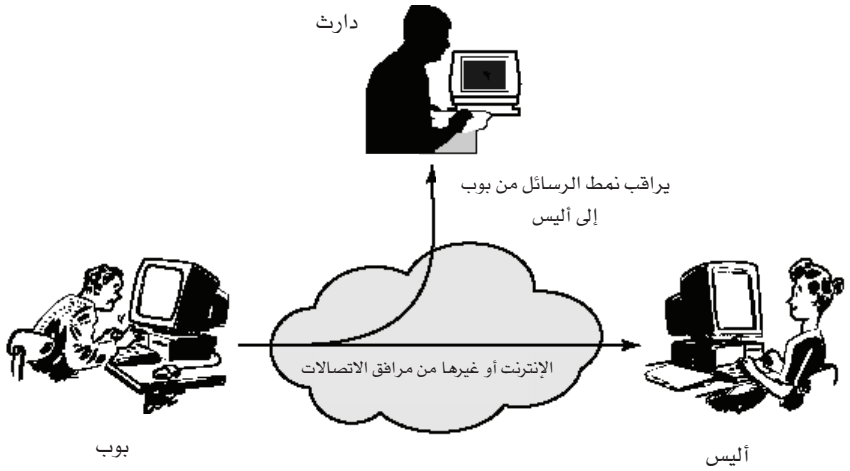
يحتوي على معلومات خاصة أو حساسة لا نريد أن يطَّلع عليها الآخرون أثناء انتقالها.

أما النوع الثاني من الهجمات السلبية فهو "هجوم تحليل حركة مرور البيانات"، وهو أكثر دهاءً وخفاءً (الشكل 1-3 (b)). افترض أن لدينا طريقة لحجب محتويات الرسائل أو حركة مرور المعلومات الأخرى، حتى لو التقط خَصْمٌ الرسالة أثناء انتقالها فإنه لن يمكنه استخلاص المعلومات منها. الأسلوب الشائع لحجب (masking) المحتويات هو التشفير. حتى بفرض أن لدينا حماية بالتشفير، فلا يزال بإمكان خَصْمٍ ما مراقبة نمط حركة تلك الرسائل. كما يمكنه تحديد موقع طرفي الاتصال ومعرفة هويتهما وملاحظة تردد الرسائل التي يجري تبادلها بينهما وكذلك أطوالها. يمكن بعد ذلك الاستفادة من هذه المعلومات في تخمين طبيعة الاتصالات التي تجري بينهما.

تعد الهجمات السلبية صعبة الاكتشاف؛ لأنها لا تقوم بأي تعديل للبيانات. وعادةً ما يتم إرسال الرسائل واستقبالها بشكلٍ عادي، ولا يكون المُرسِل ولا المُستقبِل على دراية بأن هناك طرفاً ثالثاً قد قرأ الرسائل أو تابع نمط حركة المرور. غير أنه من الممكن الحيلولة دون نجاح هذه الهجمات عادة عن طريق التشفير. وهكذا، يكون التركيز في التعامل مع الهجمات السلبية على كيفية الوقاية منها بدلاً من الكشف عنها.



(a) الإفصاح عن محتويات رسالة.



(b) تحليل حركة المرور.

الشكل 1-3: الهجمات السلبية.

1-3-2 الهجمات النشطة

تنطوي الهجمات النشطة على بعض التعديل في تدفق البيانات أو إنشاء تدفق خاطئ، ويمكن تقسيمها إلى أربع فئات: هجمات تنكر (masquerade)، وهجمات إعادة إرسال (replay attack)، وهجمات تعديل الرسائل، وهجمات حجب الخدمة (denial of service).

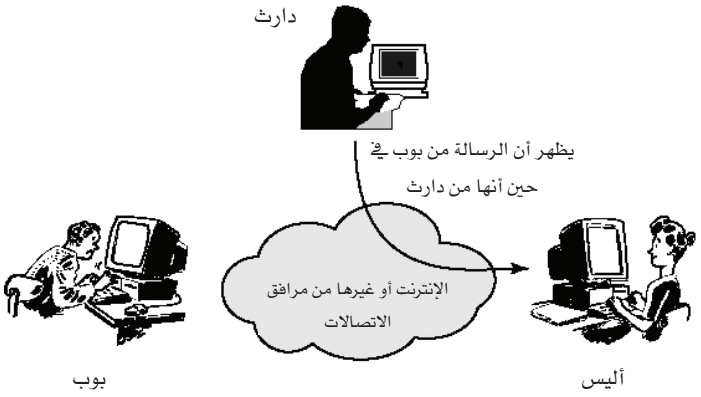
تحدث "هجمات التنكر" عندما يدَّعي كيان معين أنه كيان آخر (الشكل 1-4 (a)). عادة ما يشمل هجوم التنكر واحداً من أشكال الهجمات النشطة الأخرى. فعلى سبيل المثال، يمكن التقاط تسلسلات التوثيق التي اُستخدِمت أثناء عملية توثيق صحيحة وإعادة إرسالها لتمكين كيان ذي له امتيازات محدودة من الحصول على امتيازات إضافية.

أما هجوم "إعادة الإرسال" فيتضمن الحصول على وحدة بيانات بطريقة مستترة وإعادة إرسالها لإحداث تأثير غير مرخص به (الشكل 1-4 (b)).

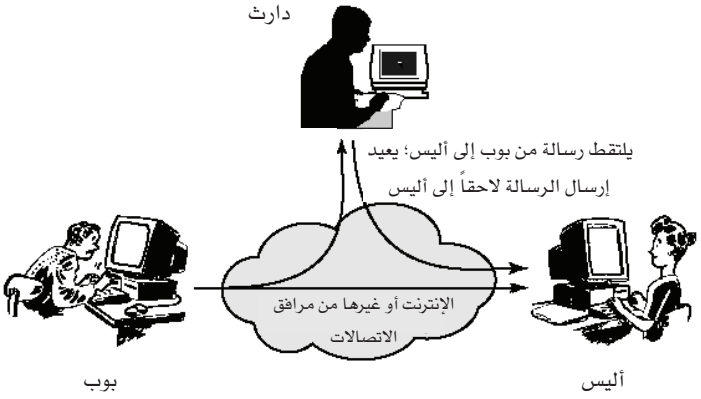
أما هجوم "تعديل الرسائل" فيعني ببساطة تغيير جزء من رسالةٍ شرعية تم اعتراضها، أو تأخير الرسائل، أو إعادة ترتيبها لإحداث تأثير غير مرخص به (الشكل 1-4 (c)). على سبيل المثال، الرسالة "اسمح لـ **جون سميث** بقراءة ملف الحسابات السرية" قد تُعدَّل لتصبح "اسمح لـ **فريد براون** بقراءة ملف الحسابات السرية".

وأخيراً هجوم "حجب الخدمة" يمنع أو يحول دون استخدام أو إدارة مرافق الاتصالات (الشكل 1-4 (d)). وقد يكون لهذا الهجوم هدفٌ محدَّدٌ؛ على سبيل المثال يمكن لكيان معين إيقاف جميع الرسائل الموجّهة لجهة محددة (ولتكن مثلاً، خدمة التدقيق الأمني). شكلٌ آخر من أشكال حجب الخدمة هو تعطيل الشبكة بكاملها، سواء بتعطيلها فعلياً أو عن طريق تحميلها بحمولة زائدة من الرسائل تفوق طاقتها بهدف إضعاف أدائها.

وللهجمات النشطة خصائص معاكسة لخصائص الهجمات السلبية. ففي حين أن الهجمات السلبية يصعب اكتشافها، إلا أن التدابير متاحة لتجنب نجاحها. في المقابل، من الصعوبة بمكان منع الهجمات النشطة تماماً؛ لأن من شأن ذلك أن يتطلب توفير حماية مادية لجميع مرافق الاتصالات والمسارات في كل الأوقات. وبدلاً من ذلك، فإن الهدف هو اكتشافها والتعافي من أي تعطيل أو خللٍ قد تسببه؛ لأن اكتشافها له تأثير رادع، كما يمكن أيضاً أن يسهم في تجنبها والوقاية منها.

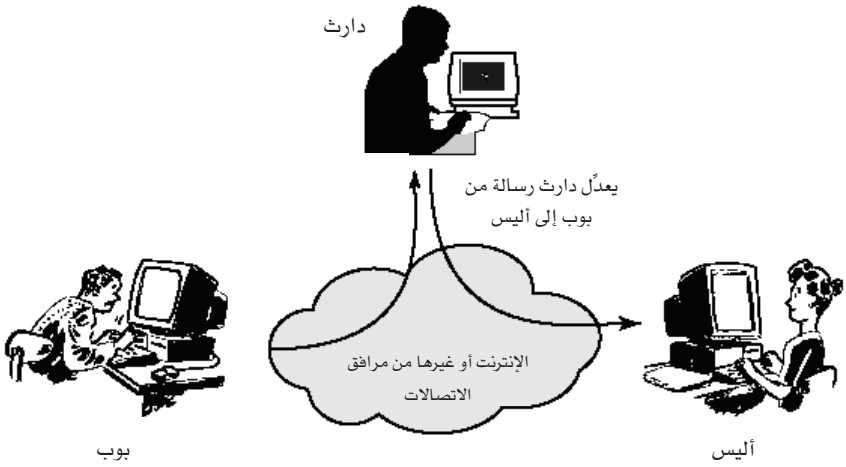


(a) هجوم التنكر.

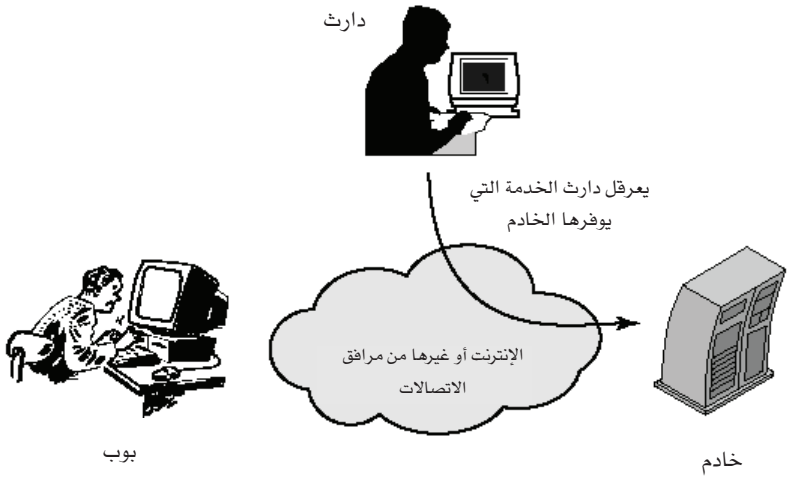


(b) إعادة الإرسال/التشغيل.

الشكل 1-4: الهجمات النشطة.



(c) تعديل الرسائل.



(d) الحرمان من الخدمة.

الشكل 1-4 (تابع): الهجمات النشطة.

4-1 خدمات الأمن

يُعرِّف X.800 الخدمة الأمنية على أنها خدمة توفرها طبقة معينة من طبقات البروتوكولات في أنظمة مفتوحة متصلة ببعضها لضمان أمن كافٍ لتلك الأنظمة أو للبيانات المنقولة بينها. ولعل تعريفاً أوضح يوجد في RFC 2828 كالآتي: عمليات أو خدمات الاتصالات التي يقدمها نظام معين لتوفير نوع محدد من الحماية لموارد النظام، وتقوم الخدمات الأمنية بتنفيذ السياسات الأمنية التي يتم تنفيذها بواسطة آليات الأمن.

يصنف X.800 هذه الخدمات إلى خمس فئات وأربع عشرة خدمة محددة (الجدول 1-2)؛ وسوف نتعرض لكل فئة منها في حينه [2].

1-4-1 التوثيق (Authentication)

تُعنَى خدمة التوثيق بالتأكد من أصالة الاتصال. ففي حالة رسالة وحيدة ـ كرسالة تحذير أو تنبيه مثلاً ـ فإن وظيفة خدمة التوثيق هي أن تؤكد للمُستلِم أن الرسالة قادمة فعلاً من المصدر المفترض أنها مُرسَلةٌ منه.

عند وجود تفاعل مستمر – كما في حالة الاتصالات بين نهاية طرفية (terminal) ونظام مضيف (host) – ينبغي مراعاة أمرين مهمين: أولهما: عند بدء الاتصال تؤكد هذه الخدمة أن كلا الكيانين موثَّقان، بمعنى أن كل كيان منهما هو فعلاً الكيان المفترض. ثانيهما: يجب أن تضمن الخدمة ألا يتم التدخل في الاتصال بين الطرفين الشرعيين من قِبَل طرف ثالث متنكر كأحد الطرفين الشرعيين بغرض إرسال أو استقبال غير مرخّص به.

---

[2] لا يوجد اتفاق عام على معاني بعض المصطلحات في أدبيات الأمن. فمثلاً يستخدم المصطلح (Integrity) أحياناً للإشارة إلى جميع جوانب أمن المعلومات، بينما يستخدم المصطلح (authentication) أحياناً للإشارة إلى التحقق من الهوية وإلى الوظائف المدرجة تحت (Integrity) في هذا الفصل. استخدامنا هنا يتوافق مع كلٍّ من X.800 وRFC2828.

الجدول 1-2: خدمات الأمن (X.800)

التحقق من الهوية: التأكد من أن الكيان المتصل هو فعلاً الكيان الذي يدعيه	
التحقق من هوية الكيان النظير	تستخدم في حالة وصلة منطقية لتوفير الثقة في هوية الكيانات المتصلة.
التحقق من هوية مصدر البيانات	في حالة نقل لاتوصيلي، تضمن هذه الخدمة أن مصدر البيانات المستلَمة هو فعلاً المصدر المُدَّعَى.
التحكم في النفاذ (الوصول): منع الاستخدام غير المرخّص للموارد (أي تحدد هذه الخدمة من يمكنهم النفاذ إلى مورد معين، وتحت أي ظروف يمكن أن يتحقق ذلك، وما هو المسموح به لهؤلاء المرخّص لهم باستخدام تلك الموارد)	
سرية البيانات: حماية البيانات من الإفصاح غير المرخّص به	
السرية التوصيلية	حماية كافة بيانات المستخدم عبر توصيلة.
السرية اللاتوصيلية	حماية كافة بيانات المستخدم بحماية كل كتلة من البيانات على حدة.
سرية حقول منتقاة	سرية حقول منتقاة ضمن بيانات المستخدم عبر توصيلة أو ضمن الكتلة الواحدة من البيانات.
سرية تدفق حركة المرور	حماية المعلومات التي قد تُستنتج من مراقبة حركة مرور تدفق البيانات.
سلامة البيانات: التأكد من أن البيانات قد استُلمت تماماً كما أُرسلت، بواسطة كيان مرخّص له (أي بدون تعديل أو إضافة أو حذف أو إعادة إرسال)	
السلامة التوصيلية مع إمكانية التعافي	توفر سلامة كافة بيانات المستخدم وتكاملها على اتصال كما تكتشف أي تعديل، أو إضافة، أو حذف، أو إعادة إرسال لأي بيانات ضمن سلسلة بيانات، مع محاولة التعافي.
السلامة التوصيلية بدون إمكانية التعافي	كما في البند السابق ولكنها توفر الكشف فقط بدون إمكانية التعافي.
السلامة التوصيلية للحقول المنتقاة	توفر سلامة الحقول المحددة ضمن بيانات المستخدم في كتل البيانات المنقولة عبر توصيلة وتتمثل في اكتشاف ما إذا كانت تلك الحقول قد تعرضت للتعديل أو الإضافة أو الحذف أو إعادة الإرسال.
السلامة اللاتوصيلية	توفر سلامة البيانات على مستوى الكتلة الواحدة من البيانات وقد تتمثل في اكتشاف أي تعديل للبيانات. وقد يصحب ذلك إمكانية محدودة لاكتشاف حالات إعادة الإرسال .
السلامة اللاتوصيلية للحقول المنتقاة	توفر سلامة الحقول المحددة على مستوى الكتلة الواحدة من البيانات؛ وتتمثل في اكتشاف ما إذا كان قد تم تعديل تلك الحقول.
عدم التنصل: توفر الحماية ضد إنكار أحد الكيانات التي أجرت اتصالاً ما أو شاركت في اتصالٍ ما أو في جزء منه.	
عدم تنصل المصدر	دليل على أن الرسالة قد أُرسلت بواسطة الطرف المحدد.
عدم تنصل الوجهة	دليل على أن الرسالة قد اُستلِمت بواسطة الطرف المحدد.

تم تعريف خدمتين من خدمات التوثيق في معيار X.800:

- **التحقق من هوية الكيان النظير (peer):** تقوم بالتأكد من هوية الكيان النظير في علاقة اتصال مباشر. ويُعدُّ الكيانان نظيرين إذا كانا ينفذان نفس البروتوكول في نظامين مختلفين؛ كوحدتين لبروتوكول TCP في نظامين متصلين. تستخدم هذه الخدمة في بداية مرحلة نقل البيانات عند اتصال النظيرين أو أثنائها، وتحاول الخدمة أيضاً توفير الثقة بأن كياناً ما لا يقوم بهجوم التنكر (masquerade) ولا بإعادة إرسال غير مرخّص به (replay) لاتصال سابق.

- **التحقق من هوية مصدر البيانات:** تقوم بالتثبت من مصدر وحدة بيانات. ولا توفر هذه الخدمة الحماية من نسخ وحدات البيانات أو تعديلها. هذا النوع من الخدمة يدعم تطبيقات لا يوجد فيها تفاعل مسبق بين الكيانات المتصلة كما في حالة البريد الإلكتروني.

2-4-1 التحكم في النفاذ (الوصول) (Access Control)

في سياق أمن الشبكة، يُعرَّف التحكم في النفاذ على أنه القدرة على الحد والتحكم في النفاذ للأنظمة المضيفة والتطبيقات من خلال وصلات الاتصالات. ولتحقيق ذلك، فإن كل كيان يحاول النفاذ لا بد أولاً من تحديد هويته، حتى يتسنى تفصيل حقوق النفاذ لتناسب كل مُستخدِم.

3-4-1 سرية البيانات (Data Confidentiality)

السرية هي حماية البيانات المُرسَلة من الهجمات السلبية. ويمكن تحديد عدة مستويات من الحماية بالنسبة لمحتوى البيانات المنقولة؛ ففي أشمل صورها تحمي هذه الخدمة جميع بيانات المُستخدِم المُرسَلة بين اثنين من المستخدمين على مدى فترة من الزمن. فعلى سبيل المثال، عند إنشاء اتصال TCP بين نظامين، فإن هذه الحماية الواسعة تمنع إفشاء أي بيانات للمستخدم مُرسَلة عبر تلك الوصلة. أمَّا في صورتها

الأضيق فإنَّ هذه الخدمة تقوم بحماية محدودة؛ قد تكون حماية رسالة واحدة، أو حتى حماية حقول بعينها في تلك الرسالة. وتُعدُّ هذه التعديلات الدقيقة أقل فائدةً من أسلوب الحماية الشاملة، وقد تكون أكثر تعقيداً وتكلفةً.

الجانب الآخر من السرية هو حماية حركة المرور من التحليل. وهذا يقتضي عدم تمكين المهاجم من مراقبة النظامين المتصلين (المُرسِل والمُستقبِل)، أو تردد أو طول الرسائل المتبادلة بينهما، أو غير ذلك من خصائص حركة المرور بينهما.

4-4-1 سلامة البيانات (Data Integrity)

كما هو الحال مع السرية، فإن خدمة السلامة يمكن أن تنطبق على سلسلة من الرسائل المتدفقة (stream of messages)، أو رسالة واحدة، أو مجموعة حقول مختارة ضمن رسالة واحدة. مرة أخرى، الصورة الأكثر فائدةً وبساطةً هي الحماية الشاملة لسلسلة الرسائل المتدفقة.

تضمن خدمة السلامة التوصيلية (connection-oriented integrity service) التي تتعامل مع سلسلة من الرسائل المتدفقة استلام الرسائل كما أُرسِلت (أي بدون نسخ، أو إضافة، أو تعديل، أو إعادة ترتيب، أو إعادة إرسال). وتغطي هذه الخدمة أيضاً تدمير البيانات. وهكذا فإن خدمة السلامة التوصيلية تتعامل مع كلٍّ من هجمات حجب الخدمة وهجمات التعديل لسلسلة الرسائل المتدفقة. في المقابل، توفر خدمة السلامة اللاتوصيلية (connectionless integrity service) – والتي تتعامل مع الرسائل الفردية بغض النظر عن أي سياق أوسع – بشكلٍ عام الحماية ضد تعديل الرسالة فقط.

نستطيع أن نميز بين الخدمة مع إمكانية التعافي من الهجوم (recovery) والخدمة بدون تلك الإمكانية. ولأن خدمة السلامة تختص بالهجمات النشطة، سنهتم هنا بعمليات الاكتشاف وليس الوقاية. إذا تم اكتشاف انتهاكٍ ما للسلامة، فإن الخدمة يمكنها ببساطة الإبلاغ عن هذا الانتهاك، حيث يتولى المُختصون أو بعض أجزاء البرمجيات الأخرى عمليات التعافي من هذا الانتهاك. وهناك آليات

أخرى متاحة للتعافي من فقدان سلامة البيانات، كما سنستعرض لاحقاً. يُعدُّ تضمين آليات ذاتية للتعافي بشكلٍ عام البديل الأكثر جاذبية.

5-4-1 عدم التنصل (Non-repudiation)

تحمي هذه الخدمة كلاً من المُرسِل والمُستقبِل من إنكار الطرف الآخر لرسالة قد أُرسِلت من قِبَل المُرسِل وتلقاها المُستقبِل بالفعل. ومن ثَمَّ، فعندما يتم إرسال رسالة، يكون بإمكان المُستقبِل أن يُثبِت أن المُرسِل المُفترَض هو في الواقع من قد أرسل تلك الرسالة. وبالمثل، عند استقبال رسالة، يمكن للمُرسِل أن يثبت أنَّ المُستقبِل المُفترَض هو من قد تلقى الرسالة بالفعل.

6-4-1 توافر الخدمة (Availability)

يُعرِّف كلٌّ من X.800 وRFC 2882 مدى توافر الخدمة بكون نظام ما أو أحد موارده متاحاً وقابلاً للاستخدام عند الطلب من قِبَل كيان مرخّص له ووفقاً لمواصفات الأداء للنظام (بمعنى أن أي نظام أو أحد موارده يُعدُّ متاحاً إذا كان يقوم بالخدمات المطلوبة منه وفقاً لتصميم النظام أو المورد كلما طُلِب منه ذلك). تؤدي بعض أنواع الهجمات إلى فقدان الخدمة أو تخفيضها، بعض هذه الهجمات يمكن مقاومته بالمضادات الآلية (automated countermeasures) كالتشفير والتوثيق، في حين يتطلب البعض الآخر نوعاً من الإجراءات المادية لمنع فقدان الخدمة التي توفرها بعض عناصر نظام موزَّع أو التعافي من فقدانها.

يعامل X.800 "توافر الخدمة" كخاصية متعلقة بمختلف الخدمات الأمنية. ومع ذلك، فإنه من المنطقي طلب هذه الخدمة تحديداً. إنَّ خدمة "توافر الخدمة" هي التي تحمي نظاماً ما لضمان توافره. وتعالج هذه الخدمة المخاوف الأمنية التي تثيرها هجمات حجب الخدمة. كما تعتمد على الإدارة والتحكم السليمين في موارد النظام، ولذا فهي تعتمد على خدمة التحكم في الوصول وغيرها من الخدمات الأمنية.

الجدول 1-3: الآليات الأمنية

آليات أمنية محددة: يمكن إدراجها ضمن طبقة البروتوكول المناسبة بغية توفير بعض الخدمات الأمنية للأنظمة المفتوحة OSI	
التشفير	استخدام خوارزميات رياضية لتحويل البيانات إلى نمط مشوّش مما يجعلها غير مفهومة. يعتمد تحويل البيانات واسترجاعها لاحقاً على خوارزمية، وقد يعتمد أيضاً على عددٍ من مفاتيح التشفير.
التوقيع الرقمي	إما بيانات مُلحقة أو تحويل مُشفّر لوحدة بيانات، مما يسمح لمُستلم وحدة البيانات بإثبات مصدرها وسلامتها، وكذلك حمايتها ضد التزوير (بواسطة المُستلم مثلاً).
التحكم في النفاذ (الوصول)	مجموعة متنوعة من الآليات تفرض حقوق النفاذ إلى الموارد.
سلامة البيانات	مجموعة متنوعة من الآليات المستخدمة لضمان سلامة وحدة البيانات أو تدفق وحدات البيانات.
تبادل التحقق من الهوية	آلية تستهدف التأكد من هوية كيان من خلال تبادل المعلومات.
حشو حركة المرور	إدراج بتات في ثغرات تدفق البيانات لاحباط محاولات تحليل حركة المرور.
التحكم في التوجيه	للتمكين من اختيار طرق محددة آمنة مادياً لبيانات معينة وتسمح بتغيير التوجيه، وخاصة عندما يُشتبه في وجود اختراق أمني.
إشهاد	استخدام طرف ثالث موثوق لضمان وجود خصائص معينة لتبادل البيانات.
آليات أمن شاملة: آليات أمنية ليست خاصة بخدمة معينة أو طبقة بروتوكول معينة للأنظمة المفتوحة OSI.	
الأداء الوظيفي الموثوق	هو الذي يُنظر إليه على أنه صحيح فيما يتعلق ببعض المعايير (كالمعايير التي تُحددها سياسة أمنية معينة).
العلامة الأمنية	العلامة المرتبطة بمورد (والذي قد يكون وحدة بيانات) وتسمي أو تحدد السمات الأمنية لذلك المورد.
اكتشاف الأحداث	اكتشاف الأحداث المتعلقة بالأمن.
سجل المراجعة الأمنية	بيانات تم تجميعها ويمكن استخدامها لتسهيل المراجعة الأمنية، والتي تشمل عمليات استعراض ومراجعة سجلات النظام وأنشطته بصورة مستقلة.
التعافي الأمني	تتعامل مع الطلبات المقدمة من آليات أمنية، كوظائف معالجة الأحداث والوظائف الإدارية، وتتخذ إجراءات التعافي الملائمة.

5-1 الآليات الأمنية

يعرض الجدول 1-3 الآليات الأمنية المُعرَّفة في X.800. وكما يتضح من الجدول، تنقسم الآليات إلى آليات تُنفَّذ في طبقة بروتوكول معينة، وأخرى ليست لها طبقة بروتوكول أو خدمة أمنية محددة. هذه الآليات سيتم تغطيتها في الأماكن الملائمة في الكتاب، ولذا لن نفصلها الآن فيما عدا تعليقنا على تعريف التشفير. يميز X.800 بين آليات تشفير منعكسة (reversible) (أي يمكن عكسها) وأخرى غير منعكسة (لا يمكن عكسها). فآلية التشفير المنعكسة هي ببساطة خوارزمية تشفير تسمح بتشفير البيانات ثم بإزالة هذا التشفير لاحقاً. أمَّا آليات التشفير غير المنعكسة فتشمل خوارزميات التحوير (hash algorithms) وأكواد توثيق الرسالة (message authentication codes)، والتي تستخدم في التوقيع الرقمي وتطبيقات توثيق الرسالة.

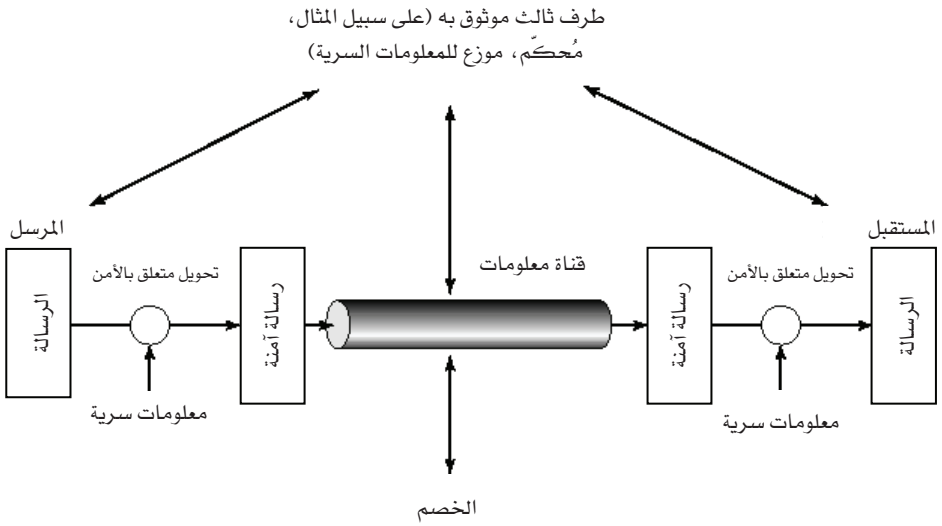
الجدول 1-4 مشتق من أحد جداول X.800؛ وهو يشير إلى العلاقة بين خدمات الأمن وآلياته.

الجدول 1-4: العلاقة بين خدمات الأمن وآلياته.

الخدمة	الآلية	التشفير	التوقيع الرقمي	التحكم في الوصول	سلامة البيانات	تبادل التوثيق	حشو حركة المرور	التحكم في التوجيه	الإشهاد
	التحقق من هوية النظير	نعم	نعم			نعم			
	التحقق من هوية مصدر البيانات	نعم	نعم						
	التحكم في النفاذ			نعم					
	السرية	نعم						نعم	
	سرية تدفق حركة المرور	نعم					نعم	نعم	
	سلامة البيانات	نعم	نعم		نعم				
	عدم التنصل (الإنكار)		نعم		نعم				نعم
	توافر الخدمة				نعم	نعم			

6-1 نموذج لأمن الشبكة

يمثل الشكل 1-5 نموذجاً عاماً لكثيرٍ مما سوف نناقشه بالكتاب. المطلوب نقل رسالة من طرف إلى آخر عبر شبكةٍ بينية، ويجب أن يتعاون هذان الطرفان حتى يتم التبادل. يتم إنشاء قناة معلومات منطقية عن طريق تحديد مسار الرسالة عبر الشبكة البينية من المصدر إلى الوجهة والتعاون بين الطرفين في استخدام بروتوكولات الاتصال (كبروتوكولات TCP/IP).



الشكل 1-5: نموذج لأمن الشبكات.

تظهر الجوانب الأمنية عندما يكون ذلك ضرورياً أو مرغوباً فيه لحماية نقل المعلومات من خَصْمٍ ما (opponent) يمكنه أن يُشكِّل تهديداً لسريتها (confidentiality) أو أصالتها (authenticity) أو ... إلخ. ويتضمن كل أسلوب من أساليب توفير الأمن عنصرين مهمين هما:

- تحويل (transformation) متعلق بالأمن يتم على المعلومات المراد إرسالها. من الأمثلة على ذلك تشفير المعلومات، والذي يُشوِّه الرسالة مما يجعلها غير

مفهومة من قِبَل الخَصْم، وكذلك إضافة شفرة تستند إلى محتويات الرسالة والتي يمكن استخدامها للتحقق من هوية المُرسِل.

- بعض المعلومات السرية المشتركة بين طرفي الاتصال والمجهولة للخَصْم ـ كما هو مأمول ـ مثل مفتاح التشفير الذي يستخدم مع دالة التحويل لتشويه الرسالة (تشفيرها) قبل نقلها وإزالة شفرتها عند استلامها [3].

وهناك حالات قد تحتاج لطرف ثالث موثوق لتحقيق الانتقال الآمن للمعلومات. فعلى سبيل المثال، قد يكون الطرف الثالث مسؤولاً عن توزيع المعلومات السرية المشتركة إلى طرفي الاتصال، مع الحفاظ عليها من أي خصم. أو قد تكون الحاجة لطرف ثالث للتحكيم في المنازعات التي قد تنشأ بين الطرفين الرئيسين فيما يتعلق بالتحقق من موثوقية انتقال الرسالة.

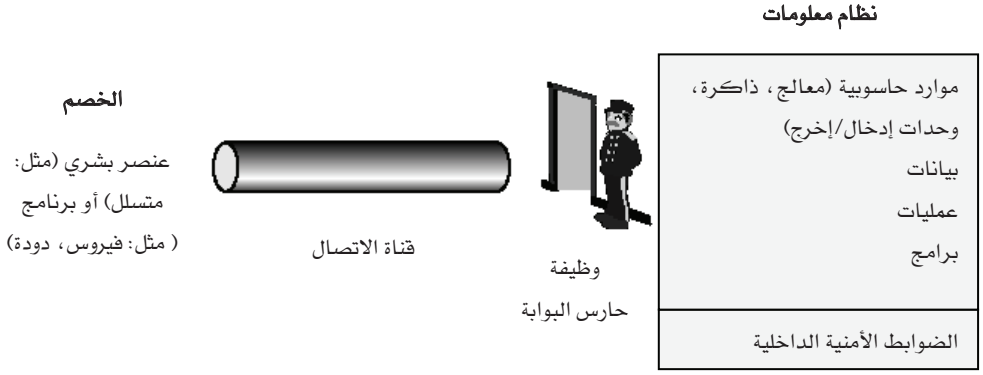
يدل هذا النموذج العام على أن هناك أربع مهام أساسية عند تصميم خدمة أمنية معينة:

1. تصميم خوارزمية لأداء التحويل المتعلق بالأمن. وينبغي أن تكون تلك الخوارزمية بحيث لا يتمكن الخَصْم من إحباط الغرض منها.
2. توليد المعلومات السرية لاستخدامها مع الخوارزمية.
3. تطوير وسائل لتوزيع ومشاركة المعلومات السرية.
4. تحديد بروتوكول لاستخدامه من قبل طرفي الاتصال اللذين يستخدمان الخوارزمية الأمنية والمعلومات السرية لتحقيق الخدمة الأمنية المطلوبة.

يركز الباب الثاني من هذا الكتاب على أنواع الآليات الأمنية والخدمات التي تتناسب مع النموذج المبين في الشكل 1-5. ومع ذلك، هناك حالات أخرى متعلقة بالأمن لا تتوافق تماماً مع هذا النموذج، ولكننا سنأخذها بعين الاعتبار في هذا الكتاب. وقد تم توضيح نموذج عام لهذه الحالات الأخرى في الشكل 1-6 الذي

[3] يناقش الفصل الثالث نوعاً من أنواع التشفير يُعرَف بتشفير المفتاح العام (Public-Key Encryption) حيث يحتاج طرف واحد فقط من طرفي الاتصال للاحتفاظ ببعض المعلومات السرية.

يعكس المخاوف بشأن حماية نظام معلومات من النفاذ غير المرغوب فيه. معظم القراء على دراية بالمخاوف الناجمة عن وجود القراصنة، الذين يحاولون اختراق النظم التي يمكن النفاذ إليها عبر الشبكة. يمكن أن يكون القرصان شخصاً ليس لديه أي نوايا خبيثة، غير أنه ببساطة يشعر بالرضا لمجرد نفاذه إلى نظام حاسب ما. وقد يكون المتسلل موظفاً ساخطاً يرغب في إحداث الضرر، أو قد يكون مجرماً يسعى إلى استغلال الحاسب لتحقيق مكاسب مالية (مثلاً الحصول على أرقام بطاقات الائتمان أو إجراء تحويلات مالية غير مشروعة).



الشكل 1-6: نموذج لأمن الوصول للشبكات.

وثمة نوع آخر من النفاذ غير المرغوب فيه إلى نظام الحاسب يتم فيه وضع المنطق (logic) الذي يستغل نقاط ضعف النظام، والذي يمكن أن يؤثر على برامج التطبيقات وكذلك برامج الخدمات، مثل برامج التحرير وبرامج الترجمة. يمكن أن تتعرض البرامج لنوعين من التهديدات:

- **تهديدات النفاذ إلى المعلومات** باعتراض البيانات أو تعديلها نيابةً عن مستخدمين لهم صلاحية الوصول لتلك البيانات.
- **تهديدات الخدمة** باستغلال عيوب الخدمة في أجهزة الحاسب لمنع استخدامها من قبل المستخدمين الشرعيين.

تعد الفيروسات والديدان مثالين على هجمات البرمجيات. يمكن إدخال مثل تلك الهجمات إلى نظام بواسطة قرص يحتوي على المنطق (logic) غير المرغوب فيه مخبأً في برنامج آخر مفيد. ويمكن أيضاً إدخاله إلى النظام عبر شبكة؛ وتستحوذ تلك الآلية الأخيرة على اهتمام أكبر في مجال أمن الشبكات.

تندرج الآليات الأمنية اللازمة للتعامل مع النفاذ غير المرغوب فيه ضمن فئتين أساسيتين (انظر الشكل 1-6). يمكن وصف الفئة الأولى بأنها وظيفة حارس البوابة. وتشمل إجراءات دخول مبنية على كلمة السر صُمِّمت بهدف منع النفاذ للجميع ماعدا المستخدمين المرخّص لهم ومنطق الفحص الذي تم تصميمه لاكتشاف ورفض الديدان، والفيروسات، وغيرها من الهجمات الأخرى المماثلة. وبمجرد حصول نفاذ إلى النظام من قِبَل مستخدم أو برنامج غير مرغوب فيه، يُلقى عبء اكتشاف المتسللين بالكامل على خط الدفاع الثاني والذي يتكون من مجموعة متنوعة من أدوات التحكم والمتابعة الداخلية التي ترصد النشاط وتحلل المعلومات المخزنة في محاولة للكشف عن وجود متسللين غير مرغوب فيهم. وسيتم تناول تلك القضايا في الباب الثالث.

7-1 معايير وجمعية الإنترنت

تم توحيد عدد كبير من بروتوكولات TCP/IP كمعايير قياسية أو هي في طريقها لذلك. من خلال اتفاق عالمي، وُكِّلت مسؤولية تطوير ونشر هذه المعايير إلى منظمة تعرف باسم جمعية الإنترنت (Internet Society)، وهي منظمة ذات عضوية مهنية تشرف على عدد من المجالس وفرق العمل المشتركة في تطوير شبكة الإنترنت والتوحيد القياسي. سيقدم هذا الجزء وصفاً موجزاً للطريقة التي تم بها تطوير معايير بروتوكولات TCP/IP.

1-7-1 منظمات الإنترنت ونشر طلبات التعليقات (RFC)

جمعية الإنترنت هي لجنة التنسيق لتصميم شبكة الإنترنت وإدارتها. وتشمل مجالات تشغيل شبكة الإنترنت نفسها، وتوحيد البروتوكولات التي تستخدمها

النظم الطرفية على الإنترنت من أجل التوافق (العمل البيني). تندرج تحت هذه الجمعية ثلاث منظمات هي المسؤولة عن العمل الفعلي لتطوير ونشر المعايير:

- مجلس البنية المعمارية للإنترنت ((Internet Architecture Board (IAB): مسؤوليته تعريف البناء الشامل للإنترنت، وتوفير الإرشاد والتوجيهات العامة لفريق مهام هندسة الإنترنت.
- فريق مهام هندسة الإنترنت ((Internet Engineering Task Force (IETF): هو ذراع الجمعية لهندسة وتطوير البروتوكولات لشبكة الإنترنت.
- فريق توجيه هندسة الإنترنت (Internet Engineering Steering Group ((IESG): مسؤول عن الإدارة الفنية لأنشطة فريق مهام هندسة الإنترنت IETF وعملية توحيد معايير الإنترنت.

تقوم مجموعات عمل مُجازة من قِبَل فريق مهام هندسة الإنترنت IETF بالتطوير الفعلي لمعايير وبروتوكولات جديدة للإنترنت. العضوية تطوعية في أي مجموعة عمل؛ ويمكن لأي طرف مهتم أن يشارك. أثناء تطوير مواصفات معينة، تقوم مجموعة العمل بإصدار مسودة من الوثيقة تكون متاحة على الإنترنت وتوضع في دليل IETF "مسودات الإنترنت" (Internet Drafts). قد تبقى الوثيقة على الإنترنت كمسودة إنترنت لمدة تصل إلى ستة أشهر، حيث يمكن للأطراف المعنية مراجعتها والتعليق عليها. وخلال ذلك الوقت، قد يوافق فريق توجيه هندسة الإنترنت IESG على نشر المسودة كطلب تعليقات (RFC). أمَّا إذا لم تُطَّور حالة المسودة لتصبح RFC خلال تلك الفترة، فإنها تسحب من الدليل. ويمكن لاحقاً للفريق العامل نشر نسخة منقحة منها.

إن فريق مهام هندسة الإنترنت (IETF) هو المسؤول عن نشر طلبات التعليقات (RFCs) بموافقة من فريق توجيه هندسة الإنترنت IESG. وتعد طلبات التعليقات (RFCs) بمثابة أوراق عمل للبحث والتطوير للإنترنت. أي وثيقة في هذه السلسلة قد تتناول أي موضوع يتعلق باتصالات الحاسبات أو أي موضوع آخر من محضر اجتماع ما لمواصفات معيارية. ينقسم عمل IETF إلى ثمانية مجالات، لكلٍّ منها مدير ويتكون كلٌّ منها من العديد من مجموعات العمل. يبين الجدول 1-5 مجالات عمل IETF واهتماماتها.

الجدول 1-5: مجالات فريق هندسة الإنترنت (IETF).

مجالات IETF	السمة	أمثلة لمجموعات العمل
عام	عمليات وإجراءات IETF	– الإطار العام للسياسة – عملية تنظيم معايير الإنترنت
التطبيقات	تطبيقات الإنترنت	– بروتوكولات ذات الصلة بالويب (HTTP) – سلامة التبادل الإلكتروني للبيانات عبر الإنترنت (EDI) – LDAP
الإنترنت	البنية التحتية للإنترنت	– بروتوكول الإنترنت الإصدار 6 (IPv6) – امتدادات PPP
العمليات والإدارة	المعايير والتعريفات لعمليات الشبكة	– بروتوكول إدارة الشبكات البسيط (SNMPv3) – مراقبة الشبكة البعيدة
التوجيه	البروتوكولات والإدارة لمعلومات التوجيه	– توجيه الإرسال المتعدد – OSPF – توجيه جودة الخدمة (QoS routing)
الأمن	بروتوكولات وتقنيات الأمن	– كيربيروس (Kerberos) – بروتوكول IPSec – X.509 – S/MIME – TLS
النقل	بروتوكولات طبقة النقل	– الخدمات المميَّزة – الاتصال عبر بروتوكول الإنترنت – NFS – بروتوكول RSVP
خدمات المستخدمين	طرق لتحسين نوعية المعلومات المتاحة لمستخدمي الإنترنت	– الاستخدام المسؤول للإنترنت – خدمات المستخدمين – وثائق FYI

1-7-2 عملية التوحيد القياسي (التقييس)

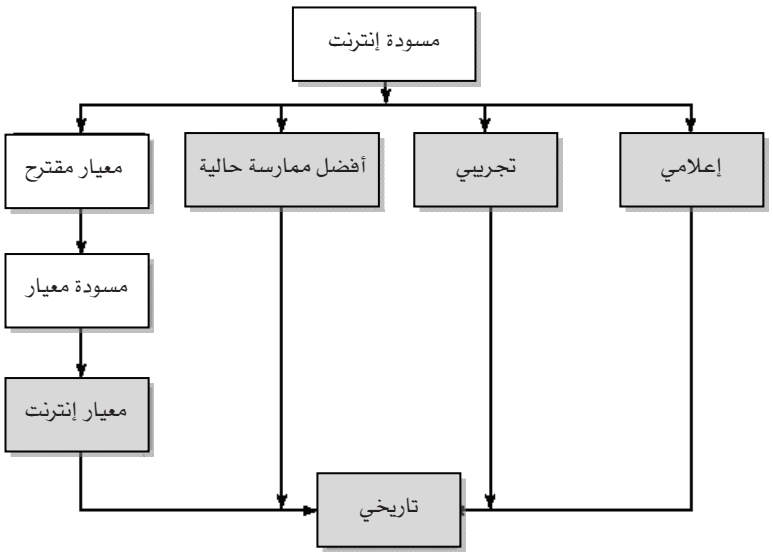
تقرير أيٍّ من طلبات التعليقات (RFCs) يصبح معياراً قياسياً للإنترنت تقوم به مجموعة توجيه هندسة الإنترنت IESG بناءً على توصيات IETF. ويراعى توافر المعايير الآتية لتصبح المواصفات معياراً قياسياً:

- أن تكون المواصفات مستقرة ومفهومة جيداً.
- أن تكون مؤهلة من الناحية الفنية.

- أن يكون قد ظهر لها عدة تنفيذات مستقلة ومتوافقة للعمل البيني مع توفر قدر كبير من الخبرة العملية في تشغيلها.
- أن تتمتع بدعم كبير من الجمهور.
- أن تكون ذات فائدة مُعترف بها لبعض أجزاء شبكة الإنترنت أو كلها.

الفارق الرئيس بين هذه المعايير وتلك المستخدمة من قِبَل الاتحاد الدولي للاتصالات ITU هو التأكيد في الأولى على التجربة العملية.

يُبيّن الجانب الأيسر من الشكل 1-7 سلسلة من الخطوات – تُعرف بمسار المعيار (standard track) – تمر عبرها المواصفات (specification) لتصبح معياراً قياسياً؛ وتم تعريف هذه العملية في RFC 2026. وتنطوي هذه الخطوات على زيادة في مقدار التدقيق والتحقق. وفي كل خطوة، يجب على فريق مهام هندسة الإنترنت (IETF) تقديم توصية للنهوض بالبروتوكول، ويجب أن تُصدِّق عليها مجموعة توجيه هندسة الإنترنت (IESG). وتبدأ العملية عندما تعتمد مجموعة توجيه هندسة الإنترنت نشر وثيقة مسودة إنترنت كطلب تعليقات (RFC) كمقترح لمعيار قياسي.



الشكل 1-7: عملية نشر طلب تعليقات (RFC) للإنترنت.

تمثل المربعات البيضاء في الشكل 1-7 الحالات المؤقتة التي ينبغي أن تدوم لأدنى فترة ممكنة عملياً. ومع ذلك ينبغي أن تبقي الوثيقة كمعيار مقترح لمدة لا تقل عن ستة أشهر، وكمسودة معيار لأربعة أشهر على الأقل، وذلك لإتاحة الوقت لمراجعتها والتعليق عليها. تمثل المستطيلات الرمادية في الشكل الحالات طويلة الأمد التي يمكن أن تمكث لسنوات.

لكي يتم تقديم مواصفة ما لتصبح مسودة معيار، يجب أن يكون قد تم تنفيذها على الأقل من قبل كيانين وأن يكون هذان التنفيذان مستقلين ومتوافقين للعمل البيني، ولهما تجربة تشغيلية مقبولة. وفي حال زيادة عدد تنفيذات تلك المواصفة زيادة كبيرة وتوفر خبرة تشغيلية كبيرة لها، يمكن رفع المواصفة لتصبح معياراً للإنترنت. وعندها يخصَص للمواصفة رقم معيار قياسي (STD)، وكذلك رقم RFC. أما عندما يتقادم بروتوكولٌ ما ويتوقف استخدامه، فيصنف كحالة تاريخية.

3-7-1 فئات معايير الإنترنت

تندرج جميع معايير الإنترنت ضمن إحدى فئتين:

- المواصفات الفنية (Technical specification-TS):
تحدد إما بروتوكولاً أو خدمةً أو إجراءً أو اصطلاحاً أو صيغةً (format). وتُعدُّ معظم معايير الإنترنت مواصفات فنية.

- بيان إمكانية التطبيق ((Applicability Statement (AS):
يحدد كيفية تطبيق واحدة أو أكثر من المواصفات الفنية (TSs) وتحت أي ظروف، وذلك لدعم إمكانية معينة للإنترنت. ويحدد بيان إمكانية التطبيق (AS) واحداً أو أكثر من المواصفات الفنية (TSs) ذات الصلة بتلك الإمكانية، ويمكن أن يحدد قيم متغيرات معينة أو نطاق القيم المقبول لتلك المتغيرات المرتبطة بمواصفة فنية أو مجموعات وظيفية من المواصفات الفنية الملائمة لتلك الإمكانية.

4-7-1 أنواع طلبات التعليقات الأخرى

هناك عدد من طلبات التعليقات التي ليست في طريقها لتصبح معايير إنترنت. وبعض طلبات التعليقات توحِّد نتائج مداولات مجتمع المختصين حول البيانات المبدئية أو الاستنتاجات بشأن أفضل وسيلة لأداء بعض العمليات أو وظائف المعالجة لفريق مهام هندسة الإنترنت IETF. يُطلق على طلبات التعليقات تلك "أفضل ممارسة حالية" (Best Current Practice (BCP)). تتبع عملية اعتماد BCPs نفس إجراءات اعتماد المعايير المقترحة تقريباً. فبخلاف مسار وثائق المعايير، لا يتطلب اعتماد BCPs ثلاث مراحل وإنما تنتقل الوثيقة من كونها مسودة إنترنت إلى وثيقة BCP معتمدة في خطوة واحدة.

البروتوكولات أو غيرها من المواصفات التي لا تُعدُّ جاهزة للتوحيد يمكن نشرها كتوصية RFC تجريبية. ويمكن إعادة تقديمها بعد مراجعتها وتنقيحها بدقة. يتم إدراج المواصفات كمعيار مقترح إذا أصبحت مستقرة عموماً، وتغلبت على أي مشاكل في التصميم، وأصبحت مفهومة جيداً، وتلقت قدراً كافياً من المراجعة من قِبَل مجتمع المهتمين بالإنترنت، وحظت بقدرٍ كافٍ من اهتمام المختصين كإسهام قيِّم. وأخيراً، تُنشر المواصفات لمجتمع الإنترنت للعِلم.

8-1 الخطوط العريضة لهذا الكتاب

يُعدّ هذا الفصل بمثابة مقدمة للكتاب كاملاً. وينتظم ما تبقى من الكتاب في ثلاثة أبواب:

- **الباب الأول:** يوفر دراسة موجزة لخوارزميات التشفير والبروتوكولات المتضمنة في تطبيقات أمن الشبكات، بما فيها التشفير، ودوال التحوير، والتوقيعات الرقمية، وتبادل المفاتيح.

- **الباب الثاني:** يبحث استخدام خوارزميات التشفير والبروتوكولات الأمنية لتوفير الأمن عبر الشبكات والإنترنت. وتشمل الموضوعات التي يغطيها هذا الجزء: التحقق من هوية المستخدم، والبريد الإلكتروني وأمن بروتوكول الإنترنت، وأمن الويب.

- **الباب الثالث:** يتعامل مع المرافق الأمنية المصممة لحماية نظام حاسب من التهديدات الأمنية، بما في ذلك المتسللين، والفيروسات، والديدان. كما يبحث هذا الجزء تقنيات برامج الحماية (الجدار الناري).

كثير من خوارزميات التشفير وبروتوكولات أمن الشبكات والتطبيقات المشروحة في هذا الكتاب قد تم توصيفها كمعايير. الأكثر أهمية هي معايير الإنترنت الوارد تعريفها في طلبات التعليقات (RFCs)، والمعايير الفيدرالية لمعالجة المعلومات (Federal Information Processing Standards (FIPS)، الصادرة عن المعهد الوطني للمعايير والتقنية (National Institute of Standards (NIST)).

9-1 توصيات للمطالعة

يتضمن [PFLE03] مقدمة جيدة لكلٍّ من الحاسب وأمن الشبكات. توجد دراستان أخريان من الدراسات الممتازة في [PIEP03] و[BISH05]. أما [BISH03] فيغطي إلى حدٍّ كبير نفس الأرضية التي يغطيها [BISH05]، ولكن بمزيد من الدقة والتفاصيل الرياضية. يمثل [SCHN00] قراءة قيّمة لكل المهتمين في مجال الحاسب أو أمن الشبكات، حيث يناقش حدود التقنية، والتشفير وخاصةً في توفير الأمن، والحاجة إلى أن يؤخذ في الاعتبار كلٌّ من الأجهزة، وتطبيقات البرامج، والشبكات، والأشخاص المهتمين بتوفير الأمن أو الهجوم عليه.

[BISH03] Bishop, M. *Computer Security: Art and Science*. Boston: Addison-Wesley, 2003.

[BISH05] Bishop, M. *Introduction to Computer Security*. Boston: Addison-Wesley, 2005.

[PFLE03] Pfleeger, C. *Security in Computing*. Upper Saddle River, NJ: Prentice Hall, 2003.

[PIEP03] Pieprzyk, J.; Hardjono, T.; and Seberry, J. *Fundamentals of Computer Security*. New York: Springer-Verlag, 2003.

[SCHN00] Schnier, B. *Secrets and Lies: Digital Security in Networked World*. New York: Wiley, 2000.

10-1 مصادر للمعلومات على الويب

هناك عدد من مصادر المعلومات المتاحة على الويب لدعم هذا الكتاب وللمساعدة في مواكبة التطورات في هذا الميدان.

1-10-1 موقع الكتاب على الويب

تم إنشاء موقع خاص بهذا الكتاب على الويب:

http://williamstallings.com/NetSec/NetSec3e.html

ويتضمن ذلك الموقع ما يأتي:

- مواقع مفيدة على الويب: هناك وصلات لمواقع ويب أخرى ذات صلة، مرتبة حسب فصول الكتاب، بما فيها المواقع المدرجة في هذه الفقرة وخلال هذا الكتاب.
- ورقة الأخطاء: تشمل قائمة بالأخطاء الواردة في هذا الكتاب، وسوف يتم تحديثها حسب الحاجة؛ لذا يرجى إرسال بريد إلكتروني بالأخطاء التي تكتشفها.
- الأشكال: جميع الأشكال الواردة في هذا الكتاب بصيغة PDF.
- الجداول: جميع الجداول الواردة في هذا الكتاب بصيغة PDF.
- شرائح عروض البور بوينت (PPT): مجموعة من شرائح العروض التقديمية مرتبة حسب فصول الكتاب.
- قائمة بريدية: يتضمن الموقع معلومات الاشتراك في القائمة البريدية للكتاب.
- مقررات دراسية عن أمن الشبكات: هناك روابط للصفحات الرئيسة للمقررات الدراسية التي تتبنى الكتاب بوصفه كتاباً أكاديمياً؛ قد تكون هذه الصفحات مفيدة في تقديم بعض الأفكار للأساتذة الآخرين حول كيفية تنظيم منهج مقرراتهم.

هناك أيضاً موقع لدعم طلبة علوم الحاسب:

http://williamstallings.com/StudentSupport.html

والغرض من هذا الموقع هو تقديم الوثائق والمعلومات والروابط الخاصة بعلوم الحاسب للطلاب والمهنيين. وتتدرج الوصلات والوثائق ضمن أربع فئات:

- الرياضيات: تشمل مراجعة سريعة للرياضيات الأساسية، ومقدمة في تحليل الاصطفاف (queuing analysis)، ومقدمة في نظام الأعداد، وروابط لمواقع عديدة في الرياضيات.
- كيف يمكن؟: نصائح وتوجيهات لحل الواجبات المنزلية، وكتابة التقارير الفنية، وإعداد العروض التقنية.

- موارد البحث: روابط لمجموعات مهمة من الأوراق البحثية والتقارير الفنية وقوائم المصادر.
- متفرقات: مجموعة متنوعة من الوثائق والوصلات المفيدة الأخرى.

2-10-1 مواقع أخرى على الويب

هناك عدد من مواقع الويب التي تقدم معلومات ذات صلة بموضوعات هذا الكتاب. في الفصول اللاحقة، وسنشير إلى مواقع محددة على الويب يمكن أن تجدها في الجزء "توصيات للمطالعة". ولأن عناوين المواقع على الويب كثيراً ما تتعرض للتغيير، فإننا لم نذكرها في الكتاب. لكن يمكن إيجاد عناوين تلك المواقع من موقع الكتاب على الويب. وستضاف عناوين أخرى لم يرد ذكرها في الكتاب لموقع الكتاب على الويب مع مرور الوقت.

ومواقع الويب الآتية هي للاهتمامات العامة المتعلقة بالتشفير وأمن الشبكات:

- موقع COAST: يتضمن مجموعة شاملة من الروابط ذات الصلة بالتشفير وأمن الشبكات.
- موضوعات الأمن بموقع فريق عمل هندسة الإنترنت (IETF): مواد تتعلق بجهود التوحيد القياسي لأمن الإنترنت.
- فهرس أمن الحاسبات والشبكات: دليل جيد عن الشركات والمنتجات التجارية، والأسئلة التي يتردد طرحها (FAQs)، ومحفوظات (أرشيف) المجموعات الإخبارية، وأوراق بحثية، وغيرها من المواقع على الويب.
- أسئلة يتردد طرحها عن التشفير (FAQ): أسئلة وأجوبة تفصيلية ومفيدة تغطي جميع جوانب التشفير.
- صفحة توم دانيجان عن الأمن: قائمة ممتازة من العناوين لمواقع على الويب عن التشفير وأمن الشبكات.
- اللجنة الفنية للأمن والخصوصية التابعة لـ (جمعية مهندسي الكهرباء والإلكترونيات (IEEE)): نسخ من نشراتها الإخبارية، ومعلومات عن الأنشطة المتصلة بـ IEEE.

- مركز مصادر أمن الحاسب: تحت رعاية المعهد الوطني للمعايير والتقنية (NIST)، ويتضمن مجموعة واسعة من المعلومات عن التهديدات الأمنية والتقنيات والمعايير.
- بؤرة الأمن: تشكيلة واسعة من المعلومات المتصلة بالأمن، مع التركيز على المنتجات المعروضة للبيع واهتمامات المستخدمين.
- معهد SANS: يشمل مجموعة واسعة من الأوراق البيضاء (white papers) على غرار موقع بؤرة الأمن.
- دليل موارد حماية البيانات: مجموعة متنوعة من الروابط.

3-10-1 مجموعات أخبار USENET

هناك عدد من مجموعات أخبار USENET مكرَّسة لبعض جوانب أمن الشبكات أو التشفير. وكما هو الحال مع جميع مجموعات USENET، قد يكون هناك الكثير من الموضوعات قليلة النفع، ولكن الأمر يستحق التجربة لمعرفة ما إذا كان هناك ما يلبي احتياجاتك. من المجموعات الأكثر ملاءمة:

- sci.crypt.research: أفضل مجموعة للالتحاق بها، وهي مجموعة لها مشرف وتتناول موضوعات بحثية؛ النشر بهذه المجموعة لابد أن يكون ذا صلة بالجوانب التقنية للتشفير.
- sci.crypt: مناقشة عامة حول التشفير والموضوعات ذات الصلة.
- sci.crypt.random-numbers: مناقشة حول عشوائية قوة التشفير.
- alt.security: مناقشة عامة للموضوعات الأمنية.
- comp.security.misc: مناقشة عامة لموضوعات أمن الحاسب.
- comp.security.firewalls: مناقشات حول منتجات جدار الحماية وتقنياته.
- comp.security.announce: أخبار وإعلانات من فريق الاستجابة لطوارئ الحاسب CERT.
- comp.risks: مناقشة المخاطر التي يتعرض لها الجمهور من أجهزة الحاسب والمستخدمين الآخرين.
- comp.virus: مجموعة مناقشة لها مشرف حول فيروسات الحاسب.

11-1 مصطلحات رئيسة

access control	التحكم في النفاذ
active threat	تهديد أمني نشط أو ظاهر
authentication	التحقق من الهوية
authenticity	الموثوقية
availability	توافر الخدمة، الإتاحية، الإتاحة
data confidentiality	سرية البيانات
data integrity	سلامة البيانات
Denial of Service (DoS)	الحرمان من الخدمة، حجب الخدمة
encryption	التشفير
integrity	السلامة
intruder	المهاجم
masquerade	هجوم التنكر
non-repudiation	عدم التنصل (عدم الإنكار)
OSI security architecture	البنية الأمنية للربط بين الأنظمة المفتوحة
passive attack	الهجوم السلبي، التهديد المستتر
replay	إعادة الإرسال، إعادة التشغيل
security attacks	الهجمات الأمنية
security mechanisms	الآليات الأمنية
security services	الخدمات الأمنية
traffic analysis	تحليل حركة المرور

12-1 أسئلة للمراجعة ومسائل

1-12-1 أسئلة للمراجعة

1-1 عرِّف الهيكل الأمني للتوصيل بين الأنظمة المفتوحة (OSI)

2-1 ما الفرق بين التهديدات الأمنية الظاهرة والمستترة؟

3-1 اسرد وعرف بإيجاز فئات الهجمات الأمنية الظاهرة والمستترة.

4-1 اسرد وعرف بإيجاز فئات الخدمات الأمنية.

5-1 اسرد وعرف بإيجاز فئات الآليات الأمنية.

2-12-1 مسائل

1-1 ارسم مصفوفة على غرار الجدول 1-4 توضح العلاقة بين الخدمات الأمنية والهجمات.

2-1 ارسم مصفوفة على غرار الجدول 1-4 توضح العلاقة بين الآليات الأمنية والهجمات.

الباب الأول

التشفير المتماثل

يُعدُّ التشفير إلى حدٍّ كبير أهم الأدوات المُستخدَمة لأتمتة تأمين الشبكات والاتصالات. ويشيع حالياً استخدام شكلين من أشكال التشفير: التشفير التقليدي أو المتماثل، والتشفير بالمفاتيح العامة أو غير المتماثل. ويتضمن الباب الأول من الكتاب دراسة مسحية للمبادئ الأساسية للتشفير المتماثل والتشفير بالمفاتيح العامة، كما يلقي الضوء على الخوارزميات شائعة الاستخدام فيهما، ويناقش أساسيات تطبيق هذين الأسلوبين للتشفير.

خريطة الطريق للباب الأول

الفصل الثاني: التشفير المتماثل وسرية الرسائل

يركز الفصل الثاني على التشفير المتماثل، وخاصةً الأساليب الأكثر استخداماً في هذا النوع من التشفير: معيار تشفير البيانات (DES) وتابعه معيار DES ثلاثي المفاتيح، وكذلك معيار التشفير المتقدم (AES). وبالإضافة إلى الأمور المتعلقة بالبناء الفعلي لخوارزميات التشفير المتماثل، هناك عدد من قضايا التصميم التي تتعلق باستخدام التشفير المتماثل لتوفير السرية. ويناقش هذا الفصل أيضاً تشفير الرسائل الطويلة، ويقارن بين تشفير الوصلة (link encryption) والتشفير من طرف إلى طرف (end-to-end encryption)، كما يستعرض أساليب توزيع المفاتيح.

الفصل الثالث: التشفير بالمفاتيح العامة وتوثيق الرسائل

كإجراء أمني، لا يقل التوثيق (authentication) أهميةً عن السرية (confidentiality). ويؤكد توثيق الرسالة – كحدٍّ أدنى – صحة ورودها من

مصدرها المُفترَض. وعلاوةً على ذلك، يمكن أن يشمل التوثيق الحماية من تغيير محتوي الرسالة، أو تأخيرها، أو إعادة بثها، أو إعادة ترتيب أجزائها. يبدأ الفصل الثالث بتحليل متطلبات التوثيق ثم يستعرض طرق التوثيق المُستخدَمة لتحقيق تلك المتطلبات. ومن العناصر الرئيسة في أنظمة التوثيق استخدام مُوثِّق (authenticator) ويأخذ عادةً شكل كود توثيق الرسالة (MAC) أو دالة تحوير (hash function). ويتناول الفصل أيضاً الاعتبارات التي ينبغي مراعاتها عند تصميم كلا النوعين من الخوارزميات، كما يتضمن تحليلاً لعدة أمثلة محددة.

يحتل التشفير بالمفاتيح العامة المرتبة الثانية ضمن أشكال التشفير الرئيسة بعد التشفير المتماثل، وقد أحدث ذلك الشكل من التشفير ثورةً في مجال تأمين الاتصالات. ويقدم الفصل الثالث أسلوب التشفير بالمفاتيح العامة، حيث يتم تناول خوارزمية RSA بالتفصيل، وكذلك استعراض موضوع إدارة المفاتيح. كما يغطي ذلك الفصل أيضاً أسلوب ديفي - هيلمان (Diffie-Hellman) الذي يُستخدَم على نطاقٍ واسع لتبادل المفاتيح. كما يُعرّف الفصل التوقيعات الرقمية ويستعرض تطبيقاتها المختلفة.

2

الفصل الثاني

التشفير المتماثل وسرية الرسائل

محتويات الفصل:

2-1 مبادئ التشفير المتماثل

2-2 خوارزميات التشفير المتماثل لكتل البيانات

2-3 تشفير تدفق البيانات وخوارزمية RC4

2-4 أنماط التشغيل بتشفير كتل البيانات

2-5 مواضع أجهزة التشفير

2-6 توزيع المفاتيح

2-7 توصيات للمطالعة

2-8 مصادر للمعلومات على الويب

2-9 مصطلحات رئيسة

2-10 أسئلة للمراجعة ومسائل

طوال فترة الظهيرة كان مونغو يعمل على شفرة شتيرن (Stern's code)، مستعيناً بدرجة رئيسة بآخر الرسائل التي قام بنسخها في موقع التنصت بميدان نيفين. وكان شتيرن واثقاً جداً. ولابد أنه كان على درايةٍ تامةٍ بأن محطة لندن المركزية على علم بوجود موقع التنصت هذا. وكان واضحاً - من فرط ثقتهم في عدم إمكانية اختراق شفرتهم- أنهم لم يكن يهمهم كم مرة يقوم مونغو بقراءة رسائلهم.

— حديث مع رجال غرباء، روث رينديل

بين قبائل وسط أستراليا يحمل كل رجل وامرأة وطفل اسماً سرياً مقدساً يخلعه عليه شيوخ القبيلة بعد ولادته بفترة وجيزة. ويكون الاسم معروفاً فقط لدى المؤتمنين من أفراد المجموعة ولا يُذكر هذا الاسم السري على الإطلاق إلا في أكثر المناسبات رسمية؛ ويُعد النطق به على مسمع من مجموعة أخرى بمثابة خرقٍ خطير لعرف القبيلة. وإذا حدث وذكر هذا الاسم، فلا يُذكر إلا همساً، وبعد أخذ كل الاحتياطات والتدابير لتجنب سماع الاسم من قِبَل أفراد من خارج المجموعة. يعتقد الفرد في تلك القبائل أن معرفة الغرباء لاسمه السري يمنحهم قدرات خاصة تمكّنهم من إيذائه بفعل السحر.

— الغصن الذهبي، السيد جيمس جورج فريزر

كان التشفير المتماثل، والذي يُعرف أيضاً بالتشفير التقليدي، أو التشفير بالمفتاح السري، أو التشفير بمفتاح واحد، النوع الوحيد من التشفير المستعمل قبل تطوير أنظمة التشفير بالمفاتيح العامة في أواخر السبعينيات.[1] ولا يزال هذا النوع بدرجةٍ كبيرةٍ أكثر نوعي التشفير استخداماً الآن.

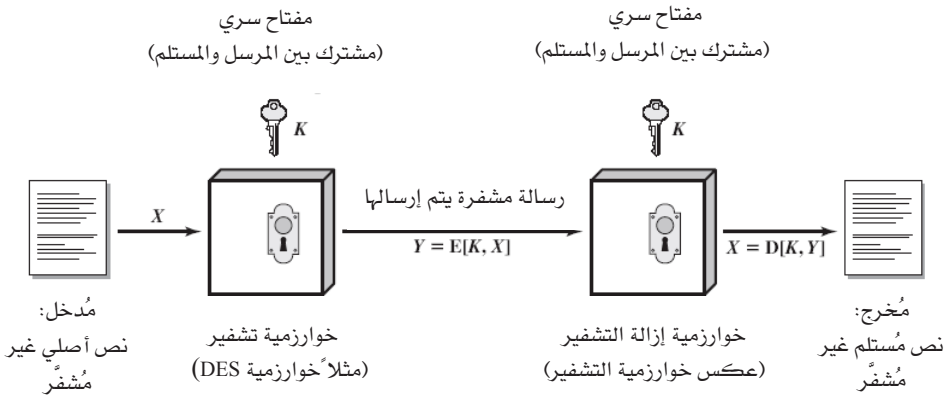
يبدأ هذا الفصل بإلقاء الضوء على نموذج عام لعملية التشفير المتماثل، مما سيمكننا من فهم السياق الذي تُستخدم فيه خوارزميات هذا النوع من التشفير. ونتناول بعد ذلك ثلاث خوارزميات مهمة لتشفير رزم (حزم) البيانات: خوارزمية DES، وخوارزمية DES الثلاثية، وخوارزمية AES. ويقدم الفصل بعد ذلك التشفير المتماثل لتدفق البيانات ويستعرض خوارزمية RC4 المستخدمة على نطاق واسع في هذا المجال. وبعدها نقوم بدراسة استخدام تلك الخوارزميات لتحقيق السرية المنشودة.

[1] ظهر وصف التشفير بالمفاتيح العامة لأول مرة في الأدبيات المفتوحة عام 1976، وتدّعي وكالة الأمن القومي الأمريكي (NSA) أنها اكتشفته قبل ذلك بعدة سنوات.

1-2 مبادئ التشفير المتماثل

يتألف نظام التشفير المتماثل من خمسة عناصر (الشكل 2-1):

- الرسالة الأصلية (غير المشفَّرة): وهي الرسالة أو البيانات الأصلية التي يتم إدخالها إلى خوارزمية التشفير كمُدخل.
- خوارزمية التشفير: تقوم خوارزمية التشفير بإدخال عدة تبديلات وتحويلات مختلفة على الرسالة الأصلية.
- المفتاح السري: المفتاح السري هو مُدخل آخر لخوارزمية التشفير.ويحدد هذا المفتاح التبديلات والتحويلات المختلفة التي تقوم الخوارزمية بإدخالها على الرسالة الأصلية.
- الرسالة المشفَّرة: وهي الرسالة المُحوّرة الناتجة كمُخرج من خوازمية التشفير. وتعتمد تلك الرسالة على الرسالة الأصلية (غير المشفَّرة) وعلى المفتاح السري. ويؤدي استخدام مفتاحين سريين مختلفين لتشفير نفس الرسالة الأصلية إلى إنتاج رسالتين مشفرتين مختلفتين.
- خوارزمية إزالة التشفير: وهي أساساً نفس خوارزمية التشفير ولكن مع تشغيلها بالعكس حيث يتم إدخال كلٍّ من الرسالة المشفَّرة والمفتاح السري إليها لتنتج الرسالة الأصلية.



الشكل 2-1: نموذج مبسط للتشفير المتماثل.

ولضمان تحقيق أمن البيانات باستخدام التشفير المتماثل هناك مطلبان أساسيان:

1. وجود خوارزمية تشفير قوية. كحد أدنى لا ينبغي للخَصْم الذي يعرف الخوارزمية وواحدة أو أكثر من الرسائل المشفَّرة بها أن يتمكن من فك تشفير الرسائل أو استنتاج المفتاح المستخدم في تشفيرها. وبعبارةٍ أخرى: مع معرفته بالخوارزمية المستخدَمة لا ينبغي أن يكون الخَصْم قادراً على فك الشفرة أو استنتاج المفتاح حتى لو كان بحوزته عدد من الرسائل المشفَّرة والرسائل الأصلية المقابلة لها.
2. ينبغي أن يكون كلٌّ من المرسل والمُستلِم قد تلقيا نسخةً من المفتاح السري بطريقةٍ مأمونة، وأن يحتفظا به بطريقةٍ آمنة. حيث إنه إذا استطاع شخصٌ ما معرفة المفتاح مع معرفته بالخوارزمية المستخدَمة، فسيمكنه قراءة أي اتصالات مشفَّرة تتم باستخدام ذلك المفتاح.

ومن المهم الإشارة إلى أن أمن التشفير المتماثل يعتمد على سرية المفتاح، وليس على سرية الخوارزمية. بمعنى أننا نفترض أنه سيكون من غير العملي فك شفرة رسالة بناءً على الرسالة المشفَّرة ومعرفة خوارزمية التشفير/إزالة التشفير. وبعبارةٍ أخرى، لا توجد حاجةٌ للحفاظ على سرية الخوارزمية؛ بل لابد فقط من الحفاظ على سرية المفتاح.

إن هذه الميزة للتشفير المتماثل تجعله صالحاً للاستخدام على نطاق واسع. فعدم الحاجة لسرية الخوارزمية مكّن المنتجين من تطوير شرائح تشفير إلكترونية (chips) منخفضة الكلفة. وهذه الشرائح متاحة حالياً على نطاق واسع ويتم استخدامها في عدد من المنتجات. ولذلك فإن الحفاظ على أمن المفتاح وسريته يمثل المشكلة الرئيسة عند استخدام التشفير المتماثل.

2-1-1 التشفير

عادةً ما تُصنَّف نظم التشفير بناءً على ثلاثة عوامل مستقلة هي:

1. نوع العمليات المُستخدَمة لتحويل الرسالة الأصلية إلى رسالةٍ مشفَّرة: تعتمد جميع خوارزميات التشفير على مبدأين أساسيين:
   - التعويض: حيث يُستبدل كل عنصر من عناصر الرسالة الأصلية (بت، أو حرف، أو مجموعة من البتات أو مجموعة من الحروف) بعنصر آخر.
   - النقل: حيث يُعاد ترتيب عناصر الرسالة الأصلية.

   وفي كلتا الحالتين يُشترط عدم فقدان أية معلومة متضمنة بالرسالة الأصلية (أي أن جميع العمليات يمكن عكسها). وتشتمل معظم نظم التشفير المستخدَمة، والتي تُعرف بالنظم الضربيّة (product systems) على عدة مراحل من التعويض والنقل.
2. عدد المفاتيح المستخدمة: إذا كان كلٌّ من المرسل والمستلِم يستخدم نفس المفتاح، يُعرف مثل هذا النظام بنظام التشفير المتماثل، أو أحادي المفتاح، أو ذي المفتاح السري، أو التقليدي. إذا كان المرسل والمستلم يستخدمان مفتاحين مختلفين، فإن النظام يُعرف بنظام التشفير غير المتماثل، أو ذي المفتاحين، أو نظام التشفير بالمفاتيح العامة.
3. الطريقة التي تتم بها معالجة الرسالة الأصلية: هناك نظم تشفير الكتلة (block cipher) حيث تُقسَّم الرسالة الأصلية إلى كتل ويتم تشفير كل كتلة منها على حِدَة لإنتاج الكتلة المناظرة من الرسالة المشفَّرة. أما في نظم تشفير التدفق (stream cipher) فتتم معالجة عناصر البيانات الداخلة في الرسالة الأصلية عنصراً عنصراً، بحيث تُنتج لكل عنصر من العناصر الداخلة نظيره من الرسالة المشفَّرة بشكل مستمر.

2-1-2 تحليل الشفرة

يُطلق على عملية محاولة اكتشاف الرسالة الأصلية أو مفتاح التشفير "تحليل الشفرة". وتعتمد الاستراتيجية التي يتبعها محلل الشفرة على طبيعة نظام التشفير ونوعية المعلومات المتوفرة لديه. ويلخص الجدول 2-1 الأنواع المختلفة من هجمات تحليل الشفرة على أساس كمية المعلومات المعروفة لدى محلل الشفرة. أصعب

مشكلة يمكن مواجهتها ستكون عند توفر الرسالة المُشفَّرة فقط. في بعض الأحيان حتى خوارزمية التشفير قد لاتكون معروفة، ولكن بشكلٍ عام يمكننا افتراض معرفة الخَصْم للخوارزمية المُستخدَمة للتشفير. أحد أنواع الهجوم المحتملة في ظل تلك الظروف هو الهجوم الاستقصائي (brute force) حيث يتم تجربة كل المفاتيح الممكنة. إذا كان عدد المفاتيح الممكنة كبيراً جداً، فإن هذه الطريقة تصبح غير عملية. وعليه يتعين على الخَصْم الاعتماد على تحليل الرسائل المشفَّرة نفسها، وعادةً بإخضاعها لعدد من الاختبارات الإحصائية. لاستخدام هذه الطريقة، ينبغي أن تتوافر للخَصْم فكرة عامة عن نوع الرسالة الأصلية، مثلاً هل هي نص بالإنجليزية أو الفرنسية، أو ملف تنفيذي (EXE file)، أو سرد لسطور برنامج مصدري بلغة جافا، أو ملف محاسبة، وهكذا. ويُعدُّ الهجوم المبني على معرفة الرسالة المشفَّرة فقط أسهل أنواع الهجوم من حيث الدفاع ضده، نظراً لأن الخَصْم يتوافر لديه أقل قدر من المعلومات يمكنه استغلالها. غير أنه في كثير من الحالات يتوافر لدى محللي الشفرات معلومات أكثر من ذلك. فقد يتمكن المحلل من الحصول على واحدة أو أكثر من الرسائل الأصلية غير المشفَّرة مع ما يقابلها من الرسائل المشفَّرة. أو قد يعرف المحلل أن أنماطاً بعينها من النصوص ستوجد بالرسالة الأصلية. على سبيل المثال، وتبدأ الملفات المكوّدة بصيغة Postscript دائماً بنفس النمط، كما أن رسائل التحويل الإلكتروني للأموال قد تتضمن في بدايتها شعاراً قياسياً موحداً، وهكذا. كل هذه أمثلة لنصوص غير مشفَّرة معروفة (known plaintext). وعن طريق هذه المعرفة قد يتمكن المحلل من استنتاج المفتاح بناءً على الطريقة التي تم بها تحويل ذلك النص الأصلي في الرسالة المشفَّرة.

الجدول 2-1: أنواع الهجوم على الرسائل المشفَّرة.

نوع الهجوم	المعروف لدى محلل الشفرة
النص المشفَّر	– خوارزمية التشفير – النص المُشفَّر المطلوب فك تشفيره
النص الأصلي المعروف	– خوارزمية التشفير – النص المشفَّر المطلوب فك تشفيره – زوج أو أكثر من النص الأصلي والنص المشفَّر المقابل والناتج عن استخدام المفتاح السري
النص الأصلي المختار	– خوارزمية التشفير – النص المشفَّر المطلوب فك تشفيره – رسالة أصلية يختارها محلل الشفرات، مع النص المشفَّر المقابل والناتج من تشفيرها باستخدام المفتاح السري
النص المشفَّر المختار	– خوارزمية التشفير – النص المشفَّر المطلوب فك تشفيره – رسالة يُفترض أنها مشفَّرة يختارها محلل الشفرات، مع النص العادي المقابل والناتج من إزالة تشفيرها باستخدام المفتاح السري
النص المختار	– خوارزمية التشفير – النص المشفَّر المطلوب فك تشفيره – رسالة أصلية يختارها محلل الشفرات، مع النص المشفَّر المقابل والناتج من تشفيرها باستخدام المفتاح السري – رسالة يُفترض أنها مشفَّرة يختارها محلل الشفرات، مع النص العادي المقابل الناتج من إزالة تشفيرها باستخدام المفتاح السري

من أنواع الهجوم وثيقة الصلة بهجوم النص الأصلي المعروف الهجوم الذي يمكن تسميته "هجوم الكلمة المحتملة" (probable-word attack). إذا كان الخَصْم يتعامل مع تشفير رسالة نثرية عامة فقد لا يعرف إلا القليل عن محتوى الرسالة. أما إذا كان يستهدف معلومات محددة للغاية، فقد تصبح أجزاء من الرسالة معروفة. فمثلاً عند إرسال ملف محاسبة كامل قد يستطيع الخَصْم تحديد مواضع بعض الكلمات الأساسية في عناوين الأعمدة. وكمثال آخر، قد يتضمن الكود المصدري

لبرنامج من إنتاج شركة معينة بياناً عن حقوق التأليف والنشر في موضع محدد بشكل قياسي.

وإذا استطاع محلل الشفرات بشكلٍ أو بآخر جعل نظام المصدر يُدخِل في النظام رسالة من اختيار المحلل فإننا نحصل على هجوم نص أصلي مختار (chosen-plaintext attack). وبصفةٍ عامة إذا كان المحلل قادراً بطريقة ما على اختيار الرسائل التي يقوم النظام بتشفيرها فقد يتعمد المحلل اختيار أنماط البيانات التي يمكن أن تفيده في الكشف عن بنية المفتاح.

تتضمن القائمة في الجدول 2-1 نوعين آخرين من الهجوم: هجوم نص مشفَّر مختار (chosen-ciphertext attack) وهجوم النص المختار (chosen-text attack). ورغم أن هذين النوعين أقل استخداماً كأسلوبين لتحليل الشفرة، فإنهما يمثّلان سبلاً ممكنة للهجوم.

فقط خوارزميات التشفير الضعيفة نسبياً هي التي لا تصمد أمام هجوم نص مشفَّر مختار. بصفةٍ عامة، تصمم خوارزميات التشفير بحيث تصمد أمام هجمات النصوص الأصلية المعروفة.

يُعدُّ نظام التشفير آمناً حسابياً (computationally secure) إذا تحقق في النص المشفَّر الذي ينتجه النظام أحد الشرطين الآتيين أو كلاهما:

- تكلفة فك الشفرة تتجاوز قيمة المعلومات المشفَّرة.
- الوقت اللازم لفك الشفرة يتجاوز العمر المفيد للمعلومات المشفَّرة.

وللأسف من الصعب جداً تقدير حجم الجهد اللازم للنجاح في تحليل شفرة نص مشفَّر. ولكن بافتراض عدم وجود نقاط ضعف أصيلة في رياضيات الخوارزمية، فيمكننا اللجوء إلى أسلوب الحل الاستقصائي (brute-force approach) حيث يمكن حساب تقديرات معقولة للتكلفة والوقت اللازمين.

ويتلخص أسلوب الحل الاستقصائي في تجربة كل مفتاح ممكن إلى أن يتم التوصل إلى ترجمة مفهومة للرسالة المشفَّرة والتي تمثل الرسالة الأصلية . وفي

المتوسط ينبغي تجربة نصف العدد الكلي للمفاتيح الممكنة لتحقيق النجاح المنشود. يبين الجدول 2-2 الوقت اللازم لتلك العملية لأطوال مختلفة للمفاتيح، ويلاحظ أن خوارزمية DES تستخدم مفتاحاً بطول 56 بتاً. وتفترض النتائج المدرجة بالجدول أن الزمن المطلوب لفك الشفرة باستخدام مفتاح معين هو 1 ميكروثانية، وهو تقدير معقول بالنسبة لقدرة الحاسبات العادية المتوفرة حالياً. أما في حال استخدام أنظمة تضم عدداً كبيراً من المعالجات الدقيقة تعمل على التوازي (massively parallel organizations of microprocessors)، فيمكن تحقيق معدلات معالجة أعلى من ذلك بكثير. ويبيّن العمود الأخير من الجدول 2-2 النتائج في حالة نظام يمكنه معالجة مليون مفتاح كل ميكروثانية. وتوضح هذه النتائج أنه بمثل هذا المستوى من الأداء فلن يكون بالإمكان اعتبار خوارزمية DES آمنةً حسابياً.

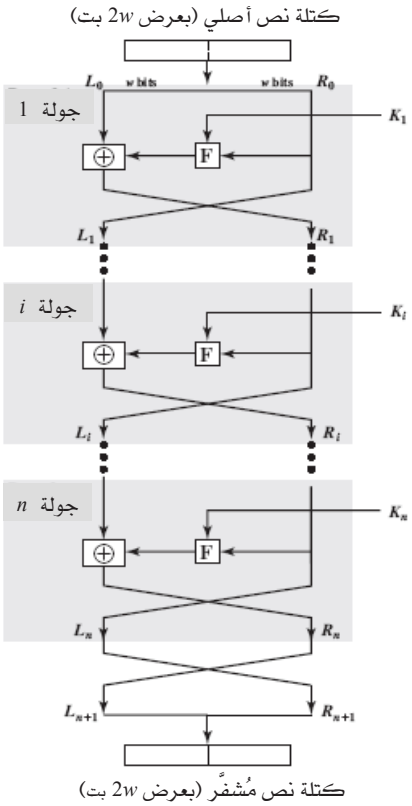
الجدول 2-2: متوسط الوقت اللازم للتوصل للمفتاح السري بأسلوب البحث الاستقصائي.

طول المفتاح	عدد المفاتيح الممكنة	الوقت اللازم بافتراض عملية تشفير كل 1 ميكروثانية	الوقت اللازم، بافتراض 10^6 عملية تشفير كل 1 ميكروثانية
32 بتاً	$2^{32} = 4.3 \times 10^9$	2^{31} ميكروثانية = 35.8 دقيقة	2.15 ميللي ثانية
56 بتاً	$2^{56} = 7.2 \times 10^{16}$	2^{55} ميكروثانية = 1142 عاماً	10.01 عاماً
128 بتاً	$2^{128} = 3.4 \times 10^{38}$	2^{127} ميكروثانية = 5.4×10^{24} عاماً	5.4×10^{18} عاماً
168 بتاً	$2^{168} = 3.7 \times 10^{50}$	2^{167} ميكروثانية = 5.9×10^{36} عاماً	5.9×10^{30} عاماً
26 حرفاً (تباديل)	$26! = 4 \times 10^{26}$	2×10^{26} ميكروثانية = 6.4×10^{12} عاماً	6.4×10^{6} عاماً

2-1-3 هيكل فيستل لخوارزميات التشفير

لكثير من خوارزميات التشفير المتماثل لكتل البيانات، بما في ذلك خوارزمية DES، هيكلٌ مشترك قام بوصفه لأول مرة هورست فيستل من شركة IBM في عام 1973 [FEIS73] ويبينه الشكل 2-2. تضم مدخلات خوارزمية التشفير كتلة بيانات من الرسالة الأصلية بطول $2w$ بت ومفتاح K، وتقسم كتلة الرسالة الأصلية

إلى نصفين: أيسر L_0 وأيمن R_0، ويمر هذان النصفان من البيانات عبر عددٍ n من جولات المعالجة، ثم يتم الجمع بينهما لتكوين الكتلة المشفَّرة. يدخل إلى كل جولة معالجة i مُدخلان أيسر L_{i-1} وأيمن R_{i-1} الناتجين من الجولة السابقة، وكذلك مفتاح فرعي K_i مستمد من المفتاح الكلي K. بشكلٍ عام، تختلف المفاتيح الفرعية K_i عن المفتاح الكلي وعن بعضها بعضاً، ويتم توليدها من المفتاح الكلي بواسطة خوارزمية فرعية لتوليد المفاتيح.



الشكل 2-2: شبكة فيستل التقليدية.

كل جولات المعالجة لها نفس الهيكل، حيث تجري عملية تعويض على النصف الأيسر من البيانات ويتم ذلك بإعمال الدالة F للتقريب (rounding) على النصف الأيمن من البيانات ثم إعمال وظيفة "أو-الحصرية" ((Exclusive-OR (XOR) على ناتج دالة التقريب مع النصف الأيسر من البيانات. ولدالة التقريب نفس الهيكل العام في كل الجولات، ولكن في كل جولة يتم تعديل متغيراتها حسب مفتاح التقريب الفرعي K_i. وبعد عملية التعويض تلك تُجرى عملية تبديل ما بين نصفي البيانات (بإحلال النصف الأيمن محل الأيسر والعكس بالعكس).

يُعدُّ هيكل فيستل أحد الأمثلة على هيكل عام تستخدمه جميع أنظمة التشفير المتمائل للكتل. وبشكلٍ عام، يتألف التشفير من سلسلة من الجولات تتم في كل جولة منها عمليات تعويض وتبديل تعتمد على المفتاح السري. وتعتمد تفاصيل خوارزمية معيَّنة للتشفير المتماثل للكتل على اختيار المتغيرات وسمات التصميم الآتية:

- مقاس الكتلة: توفِّر الكتلة الكبيرة مزيداً من الأمن (بافتراض تساوي العوامل الأخرى) ولكنها تقلل من سرعة التشفير وإزالته. تحقق الكتلة التي حجمها 128 بتاً توازناً معقولاً بين هذين العاملين وتُستخدم في معظم التصاميم الحديثة لتشفير الكتل تقريباً.
- طول المفتاح: توفِّر المفاتيح الطويلة المزيد من الأمن ولكن ذلك يمكن أن يقلل من سرعة التشفير وإزالته. والطول الأكثر شيوعاً في خوارزميات التشفير الحديثة هو 128 بتاً.
- عدد الجولات: من أساسيات التشفير المتماثل للكتل أن جولة واحدة لا تكفي لتوفير الأمن المطلوب، ولكن جولات متعددة توفِّر أمناً أكبر. والعدد المستخدم عادةً هو 16 جولة.
- الخوارزمية الفرعية لتوليد المفاتيح: تؤدي زيادة التعقيد في تلك الخوارزمية إلى زيادة صعوبة عملية تحليل الشفرة.
- دالة التقريب: مرة أخرى، تؤدي زيادة التعقيد هنا إلى المزيد من القدرة على مقاومة عمليات تحليل الشفرة.

بالإضافة لذلك، هناك الاعتباران الآتيان فيما يتعلق بتصميم أنظمة التشفير المتماثل للكتل:

- التشفير وإزالة التشفير السريع بالبرمجيات: في كثيرٍ من الأحيان يتم دمج التشفير ضمن التطبيقات أو وظائف الخدمة البرمجية مما لا يُحتاج معه إلى استخدام وحدات عتاد بنائية خاصة بذلك (hardware). وفي هذه الحالة لابد من أخذ سرعة تشغيل تلك البرامج كأحد الاعتبارات المهمة.
- سهولة التحليل: رغم أننا نود جعل خوارزميتنا أصعب مايمكن كي تستعصي على محاولات تحليل الشفرة، فإن هناك فائدة كبيرة في جعل الخوارزمية سهلة التحليل. فعندما يمكننا وصف الخوارزمية بإيجاز وشرحها بوضوح فإنه يسهل علينا تحليل الخوارزمية فيما يتعلق بنقاط ضعفها إزاء محاولات تحليل الشفرة، ومن ثم التوصل إلى مستوى أعلى من الضمان لقوتها. وجدير بالذكر أن خوارزمية DES، على سبيل المثال، لا تتصف بسهولة تحليل وظائفها.

عملية إزالة التشفير في التشفير المتماثل للكتل هي أساساً نفس عملية التشفير. والقاعدة هنا على النحو الآتي: استخدم الرسالة المشفَّرة كمُدخل للخوارزمية، ولكن مع استعمال المفاتيح الفرعية K_i بترتيب عكسي، بمعنى استعمل المفتاح K_n في الجولة الأولى وK_{n-1} في الجولة الثانية، وهكذا إلى أن تستعمل K_1 في الجولة الأخيرة. تلك ميزة لطيفة، حيث إننا لا نحتاج لتنفيذ خوارزميتين مختلفتين – واحدة للتشفير وأخرى لإزالته.

2-2 خوارزميات التشفير المتماثل للكتل

تُعدُّ خوارزميات تشفير الكتل أكثر أنواع خوارزميات التشفير المتماثل شيوعاً. تقوم عملية تشفير الكتل بمعالجة بيانات النص الأصلي على شكل كتل ذات مقاس ثابت وتنتج كتلة واحدة من النص المشفَّر (بنفس الحجم) لكل كتلة من كتل النص الأصلي. ويركّز هذا الجزء على أهم ثلاث خوارزميات للتشفير المتماثل

للكتل، وهي: معيار تشفير البيانات (DES)، ومعيار DES الثلاثي (3DES)، ومعيار التشفير المتقدم (AES).

2-2-1 معيار تشفير البيانات

تأسس نظام التشفير الأكثر استخداماً في العالم على معيار تشفير البيانات (DES) المعتمد عام 1977 من قِبل المكتب الوطني الأمريكي للمعايير (National Bureau of Standards) – وهو الآن المعهد الوطني الأمريكي للمعايير والتقنية (NIST) – كمعيار فيدرالي لمعالجة المعلومات رقم 46 (FIPS PUB 46). ويُطلق على الخوارزمية نفسها خوارزمية تشفير البيانات (DEA).[2]

❖ وصف الخوارزمية:

يبلغ طول الرسالة الأصلية 64 بتاً بينما يبلغ طول المفتاح 56 بتاً. أما الرسائل الأصلية الأطول من ذلك فيتم معالجتها على شكل كتل حجم كلٍّ منها 64 بتاً. يشبه هيكل خوارزمية DES هيكل شبكة فيستل المبين في الشكل 2-2 مع بعض الاختلافات البسيطة. وتستخدم خوارزمية DES عدد 16 جولة من جولات المعالجة. من المفتاح الأصلي (بطول 56 بتاً) يتم توليد 16 مفتاحاً فرعياً يُستخدم واحدٌ منها لكل جولة معالجة.

عملية إزالة التشفير بخوارزمية DES هي أساساً نفس عملية التشفير. القاعدة هنا على النحو الآتي: استخدم الرسالة المشفَّرة كمُدخل للخوارزمية، ولكن مع استعمال المفاتيح الفرعية K_i بترتيب عكسي، بمعنى استعمل المفتاح K_{16} في الجولة

---

[2] المصطلحات هنا مربكة بعض الشيء. فحتى وقتٍ قريبٍ كان من الممكن استخدام المصطلحين DES وDEA بنفس المعنى. غير أن أحدث طبعة من وثيقة DES تتضمن مواصفات خوارزمية DEA الواردة هنا بالإضافة إلى خوارزمية (3DES) التي سنتناولها فيما بعد. وتمثل كلٌّ من DEA و3DES جزءاً من معيار تشفير البيانات. وعلاوة على ذلك – إلى أن تم اعتماد الرسمي لمصطلح 3DES مؤخراً – كانت خوارزمية DEA الثلاثية يشار إليها عادةً بـ triple DES بينما تُكتب كـ 3DES. لتسهيل الأمر هنا سوف نستخدم 3DES.

الأولى وK_{15} في الجولة الثانية، وهكذا إلى أن تستعمل K_1 في الجولة الأخيرة (رقم 16).

❖ قوة خوارزمية DES:

تنقسم المخاوف بشأن قوة DES إلى فئتين: مخاوف حول الخوارزمية نفسها، ومخاوف حول طول المفتاح المستخدم (56 بتاً). تتركز المخاوف من الفئة الأولى حول إمكانية النجاح في تحليل الشفرة من خلال استغلال خصائص خوارزمية DES. على مر السنين، كانت هناك محاولات عديدة لاكتشاف واستغلال نقاط الضعف في الخوارزمية، مما جعل DES أكثر خوارزميات التشفير التي دُرست. ورغم قيام العديد من تلك المحاولات، فإنه لم ينجح أحد حتى الآن في اكتشاف نقطة ضعف قاتلة في DES.[3]

يُعدُّ طول المفتاح أكثر مدعاةٍ للقلق. بمفتاح طوله 56 بتاً يكون عدد المفاتيح الممكنة 2^{56}، أي حوالي 7.2×10^{16} مفتاحاً. وهكذا فظاهرياً يبدو أن الهجوم الاستقصائي غير عملي. فبافتراض أنه في المتوسط يحتاج الأمر إلى تجربة نصف عدد المفاتيح أثناء عملية البحث، فإن آلة تشفير DES واحدة تنفذ عملية تشفير كل 1 ميكروثانية ستستغرق أكثر من ألف سنة لفك الشفرة (انظر الجدول 2-2).

غير أن افتراض ميكروثانية واحدة كسرعة تشفير يُعد افتراضاً مفرطاً في التحفظ. أخيراً وفي يوليو 1998 ثبت بشكلٍ قاطع أن خوارزمية DES غير آمنة، عندما أعلنت مؤسسة Electronic Frontier Foundation (EFF) أنها تمكنت من فك شفرة DES باستخدام آلة صُمّمت خصيصاً لفك شفرات DES التي تم إنتاجها بكلفة لا تتجاوز 250 ألف دولار، حيث استغرق ذلك أقل من ثلاثة أيام. وقد نشرت مؤسسة EFF وصفاً تفصيلياً للآلة، مما مكّن الآخرين من بناء آلاتهم الخاصة لفك شفرات DES [EFF98]. وبطبيعة الحال ستستمر أسعار عتاد الحاسبات في الانخفاض بينما تتزايد سرعاتها، مما سيجعل خوارزمية DES عديمة القيمة تقريباً في نهاية المطاف.

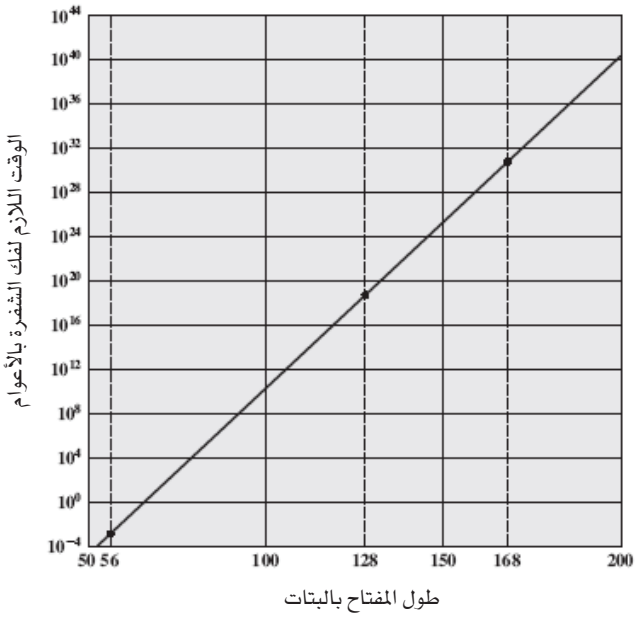
---

[3] على الأقل، لم يعلن أحد عن مثل هذا الاكتشاف بعد.

غير أنه من المهم الإشارة إلى أن عملية البحث عن المفتاح لا تقتصر على مجرد تجربة كل المفاتيح الممكنة الواحد تلو الآخر. فمحلل الشفرة ينبغي أن تكون لديه القدرة على التعرف على الرسالة الناتجة من إزالة التشفير كنص عادي، اللهم إلا إذا كانت الرسالة الأصلية معروفة فعلاً. إذا كانت الرسالة مجرد نص عادي باللغة الإنجليزية، فإنها ستظهر فجأة وبوضوح عند استخدام المفتاح الصحيح، ولكن ينبغي أتمتة عملية التعرف على النص الإنجليزي. إذا كانت الرسالة النصّية قد تم ضغطها قبل التشفير فإن التعرف عليها سيكون أكثر صعوبة. وإذا كانت الرسالة من نوع عام من البيانات كالبيانات العددية، وكانت مضغوطة، فإن أتمتة عملية التعرف ستصبح أكثر صعوبة. وعليه فإننا نحتاج إلى دعم الأسلوب الاستقصائي في البحث عن المفتاح بقدر من المعلومات عن النص العادي المتوقع استرجاعه وبعض وسائل التمييز التلقائي بين الرسالة الأصلية المتوقعة والرسائل المشوَّشة الناتجة عن استخدام مفتاح غير صحيح. إن الطريقة التي اتبعتها مؤسسة EFF لفك الشفرة تناولت هذه الأمور واستخدمت بعض أساليب الأتمتة التي يمكن أن تكون فعالة في كثير من المواقف.

وثمة نقطة أخيرة: إذا كان الشكل الوحيد من أشكال الهجوم التي يمكن استخدامها ضد خوارزمية التشفير هو الهجوم الاستقصائي، فإن السبيل لمواجهة مثل تلك الهجمات يكون واضحاً: استخدم مفاتيح أطول. لتحصيل تصور تقريبي عن طول المفتاح المطلوب، لنستخدم آلة EFF لفك الشفرة كأساس لعمل التقديرات اللازمة. جدير بالذكر أن تلك الآلة كانت نموذجاً أولياً ويمكننا افتراض أن الأجهزة التي تستخدم تقنيات اليوم ستكون أسرع وأكثر فاعلية من حيث التكلفة. لو افترضنا أن جهاز فك الشفرة يمكنه إزالة التشفير بمعدل مليون مرة كل ميكروثانية، وهو نفس المعدل المذكور في العمود الأخير من الجدول 2-2، فإن فك شفرة DES سيستغرق نحو 10 ساعات. ويمثل ذلك تسريعاً بحوالي 7 مرات مقارنةً بنتائج آلة EFF. باستخدام هذه النسبة، ويبيّن الشكل 2-3 الوقت اللازم لفك رسالة مشفَّرة بخوارزمية من طراز DES كدالة في طول المفتاح. فعلى سبيل المثال، باستخدام مفتاح طوله 128 بتاً، وهو طول شائع الاستخدام بين خوارزميات التشفير المعاصرة، يستغرق الأمر أكثر من 10^{18} عاماً لفك الشفرة باستخدام آلة EFF. حتى

لو أمكن تسريع إزالة التشفير الذي تقوم به الآلة بنسبة تريليون واحد (10^{12})، فسيستغرق فك الشفرة أكثر من مليون عاماً. وعليه، فإن استخدام مفتاح طوله 128 بتاً يضمن عملياً عدم اختراق (فك شفرات) الخوارزمية بواسطة الأساليب الاستقصائية لاكتشاف المفتاح.



الشكل 2-3: الوقت اللازم لفك الشفرة
(بافتراض إزالة التشفير بمعدل 10^6 مرة كل ميكروثانية)

2-2-2 خوارزمية DES الثلاثية

تم اعتماد خوارزمية DES الثلاثية (3DES) كمعيار للاستخدام في التطبيقات المالية في عام 1985 وذلك ضمن معيار المعهد الوطني الأمريكي للمعايير (ANSI) رقم X9.17. وفي عام 1999 تم تضمين 3DES كجزء من معيار تشفير البيانات بموجب النشرة FIPS PUB 46-3. تستخدم 3DES ثلاثة مفاتيح وتتضمن تنفيذ

خوارزمية DES ثلاث مرات. تتبع الخوارزمية تسلسلاً من عمليات التشفير (إزالة التشفير) التشفير ((encrypt-decrypt-encrypt (EDE) (انظر الشكل 2-4 (a)):

$$C = E(K_3, D(K_2, E(K_1, P)))$$

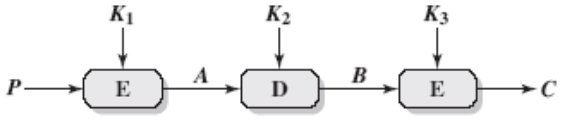
حيث:

C = الرسالة المشفَّرة

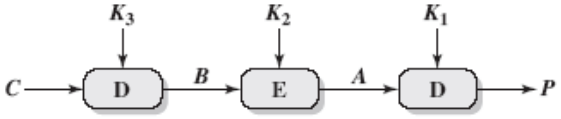
P = الرسالة الأصلية

$E(K,X)$ = تشفير الرسالة X باستخدام المفتاح K

$D(K,Y)$ = إزالة تشفير الرسالة Y باستخدام المفتاح K



(a) التشفير



(b) إزالة التشفير

الشكل 2-4: خوارزمية DES الثلاثية (3DES).

عملية إزالة التشفير ببساطة هي نفسها عملية التشفير مع عكس ترتيب استخدام المفاتيح (انظر الشكل 2-4 (b)):

$$P = D(K_1, E(K_2, E(K_3, C)))$$

ليست هناك أهمية تشفيرية معينة لعملية إزالة التشفير في المرحلة الثانية من خوارزمية DES3. الميزة الوحيدة لذلك هي تمكين مستخدمي 3DES من إزالة تشفير البيانات المشفَّرة بواسطة مستخدمي خوارزمية DES القديمة المفردة:

$$C = E(K_1, D(K_1, E(K_1, P))) = E[K,P]$$

باستخدام ثلاثة مفاتيح مختلفة، يكون لخوارزمية 3DES مفتاح بطول فعّال يبلغ 168 بتاً. أما معيار FIPS 46-3 فيسمح أيضاً باستخدام مفتاحين، وذلك بجعل $K_1 = K_2$، ومن ثم يكون الطول الفعّال للمفتاح 112 بتاً. يتضمن معيار FIPS 46-3 المبادئ التوجيهية الآتية فيما يتعلق باستخدام 3DES:

- 3DES هي خوارزمية التشفير المتماثل المعتمدة والمفضلة لدى FIPS.
- يسمح المعيار للأنظمة المتقادمة (legacy systems) باستخدام خوارزمية DES الأصلية، والتي تستخدم مفتاحاً واحداً طوله 56 بتاً، ولكن ينبغي أن تدعم كل الأنظمة الجديدة خوارزمية 3DES.
- تشجيع الهيئات الحكومية التي تستخدم أنظمة DES متقادمة على الانتقال إلى 3DES.
- من المتوقع أن تتعايش كلٌّ من 3DES ومعيار التشفير المتقدم (AES) كخوارزميات معتمدة من قِبَل FIPS، مما يسمح بالانتقال التدريجي إلى AES.

من الواضح صعوبة اختراق خوارزمية 3DES نظراً لأن الخوارزمية الأساسية وراءها هي خوارزمية DEA، ولهذا فإن 3DES يمكن أن تدّعي لنفسها نفس مقاومة تحليل الشفرة التي تتوفَّر لخوارزمية DEA. فضلاً عن ذلك، ونظراً لطول المفتاح الذي يبلغ 168 بتاً فإن الهجمات الاستقصائية لفك الشفرات تصبح مستحيلة من الناحية العملية.

في نهاية المطاف يُفترض أن تحل AES محل 3DES، ولكن هذه العملية سوف تستغرق عدة سنوات. يتوقع معهد NIST أن تبقى 3DES كخوارزمية معتمدة (لاستخدام الحكومة الأمريكية) في المستقبل المنظور.

3-2-2 خوارزمية معيار التشفير المتقدم (AES)

تمتاز 3DES بميزتين مهمتين تضمنان استخدامها على نطاق واسع خلال السنوات القليلة القادمة. أولاً، بمفتاح طوله 168 بتاً تتغلب الخوارزمية على نقاط الضعف التي عانت منها DEA في مواجهة الهجمات الاستقصائية. وثانياً، فإن خوارزمية التشفير الأساسية وراء 3DES هي نفس الخوارزمية المستخدمة في DEA والتي تعرضت للفحص والتمحيص أكثر من أي خوارزمية تشفير أخرى على مدى فترة طويلة من الزمن، ولم يثبت وجود أي هجوم تحليل شفرات فعال ضدها مبني على أساس الخوارزمية نفسها وليس على مبدأ الحل الاستقصائي. وبناءً على ذلك، فهناك درجة عالية من الثقة بأن 3DES تتمتع بمقاومة عاليةٍ جداً لتحليل الشفرة. وإذا كان الأمن هو العامل الوحيد الذي يُؤخذ بعين الاعتبار، فإن 3DES تصلح لاختيارها كخوارزمية التشفير المعيارية لعقود قادمة.

يكمن العيب الرئيس لخوارزمية 3DES في كونها بطيئةٍ نسبياً عند تنفيذها برمجياً، حيث كانت خوارزمية DEA الأصلية قد صُممت في منتصف السبعينيات بغرض تنفيذها في عتاد (hardware)، ولكن تنفيذها من خلال برمجيات لا يتمتع بكفاءة عالية. وعلى هذا فإن 3DES، والتي تستخدم ثلاثة أضعاف عدد الجولات التي تستخدمها DEA، ستكون من ثَمَّ أبطأ بنفس النسبة. ومن العيوب الثانوية أن كلاً من DEA و3DES تستخدم كتلاً بحجم 64 بتاً. لأسباب تتعلق بالكفاءة والأمن، يُعدُّ استخدام الكتل ذات الحجم الأكبر أمراً مرغوباً فيه.

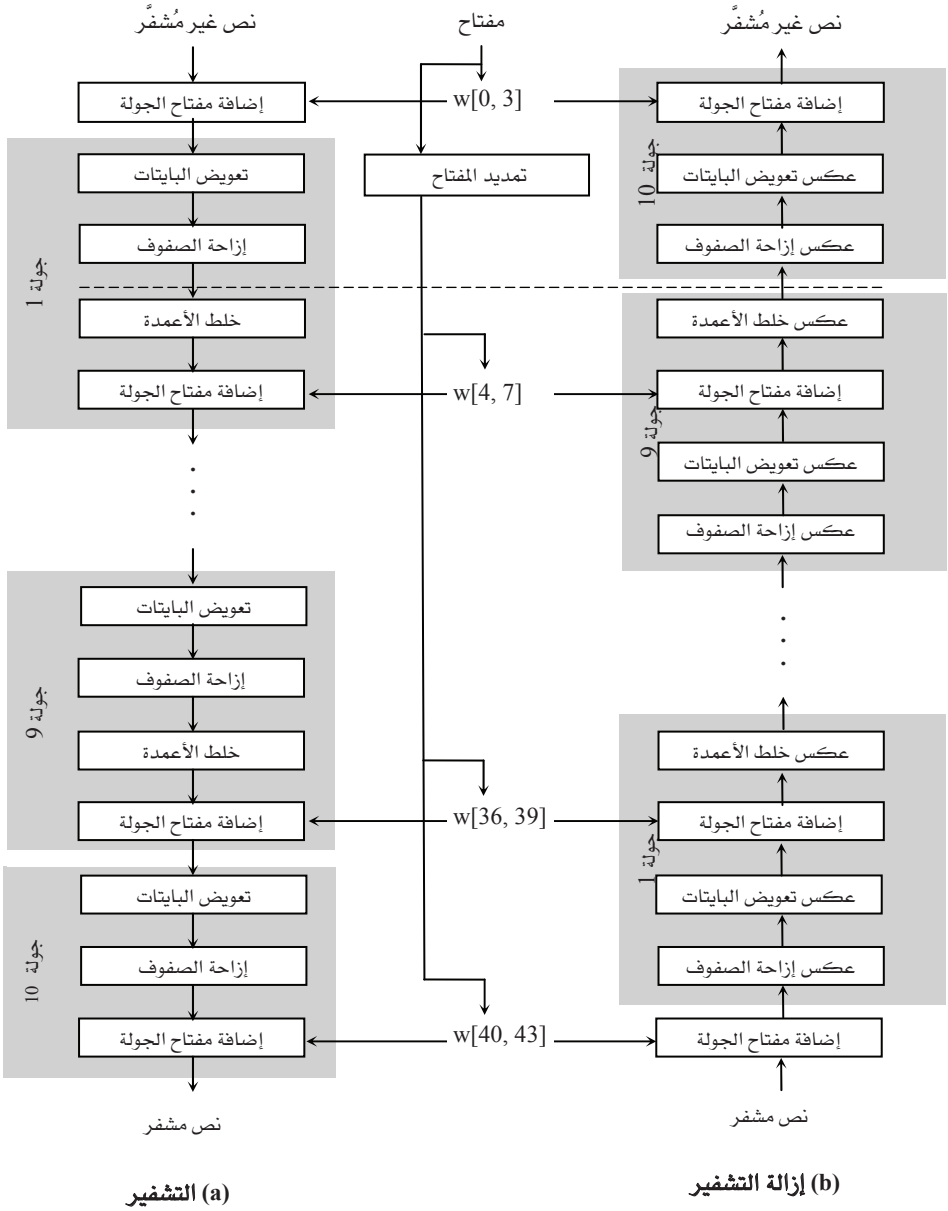
بسبب هذه العيوب، لا تُعدُّ 3DES مرشحاً معقولاً للاستخدام على المدى البعيد. وكبديل، أصدر معهد NIST في عام 1997 نداءً لتقديم عروض بمقترحات جديدة لمعيار التشفير المتقدم ((Advanced Encryption Standard (AES))، والذي ينبغي أن يوفر قوة أمنية تعادل أو تتفوق على قوة 3DES مع تحسين كفاءة الأداء بشكل كبير. وبالإضافة إلى تلك المتطلبات العامة، اشترط معهد NIST أن تكون AES خوارزمية تشفيرٍ متماثل، وأن تستخدم كتل بيانات بحجم 128 بتاً وتدعم مفاتيح بأطوال 128، و192، و256 بتاً. وتضمَّنت قائمة معايير التقييم لمقترحات الخوارزمية الجديدة: الأمن، والكفاءة الحسابية، واحتياجات الذاكرة، والملاءمة للتنفيذ في عتاد أو برمجيات، والمرونة.

في الجولة الأولى من عملية التقييم، تم قبول 15 خوارزمية مقترحة، وعبر الجولة الثانية أمكن تضييق مجال الاختيار إلى 5 خوارزميات. في نوفمبر 2001 انتهى معهد NIST من عملية التقييم ونشر وثيقة المعيار النهائي (FIPS PUB 197). وقد اختار معهد NIST خوارزمية Rijndael المقترحة لتكون خوارزمية AES الجديدة. والباحثان اللذان تقدما بهذا المقترح هما أخصائيا التشفير Joan Daemen وVincent Rijmen من بلجيكا.

❖ لمحة عن الخوارزمية:

تستخدم AES كتلة مقاسها 128 بتاً، ومفتاحاً طوله إمّا 128، أو 192، أو 256 بتاً. في وصفنا للخوارزمية هنا سنفترض مفتاحاً بطول 128 بتاً، وهو الطول المُتوقَّع أن يكون الأكثر شيوعاً.

يبيّن الشكل 2-5 هيكل خوارزمية AES. مُدخل الخوارزمية عبارة عن كتلة واحدة حجمها 128 بتاً. تُمثَّل تلك الكتلة على شكل مصفوفة مربّعة من البايتات في الوثيقة رقم FIPS PUB 197. ويتم نسخ الكتلة إلى صفيف الحالة (State array) حيث يجري تعديلها في كل مرحلة من مراحل التشفير أو إزالة التشفير. بعد المرحلة الأخيرة يتم نسخ صفيف الحالة إلى مصفوفة المُخرج (output matrix). بالمثل، يُمثّل المفتاح بطول 128 بتاً كمصفوفة مربعة من البايتات. ويتم بعد ذلك تمديد المفتاح إلى صفيف من كلمات جدولة المفتاح (key schedule words)؛ حيث تضم كل كلمة منها أربعة بايتات. يتألف الجدول الكلي للمفتاح بطول 128 بتاً من 44 كلمة. وتُرتّب البايتات ضمن المصفوفات حسب الأعمدة. لذا، وعلى سبيل المثال، تحتل البايتات الأربعة الأولى من كتلة الرسالة الأصلية بطول 128 بتاً العمود الأول من مصفوفة المُدخل (in matrix)، وتحتل البايتات الأربعة التالية العمود الثاني، وهكذا. وبالمثل، فإن البايتات الأربعة الأولى من المفتاح الممدد، والتي تشكل كلمة، تحتل العمود الأول من المصفوفة **w** (الشكل 2-5).



الشكل 2-5: التشفير وإزالة التشفير في خوارزمية AES.

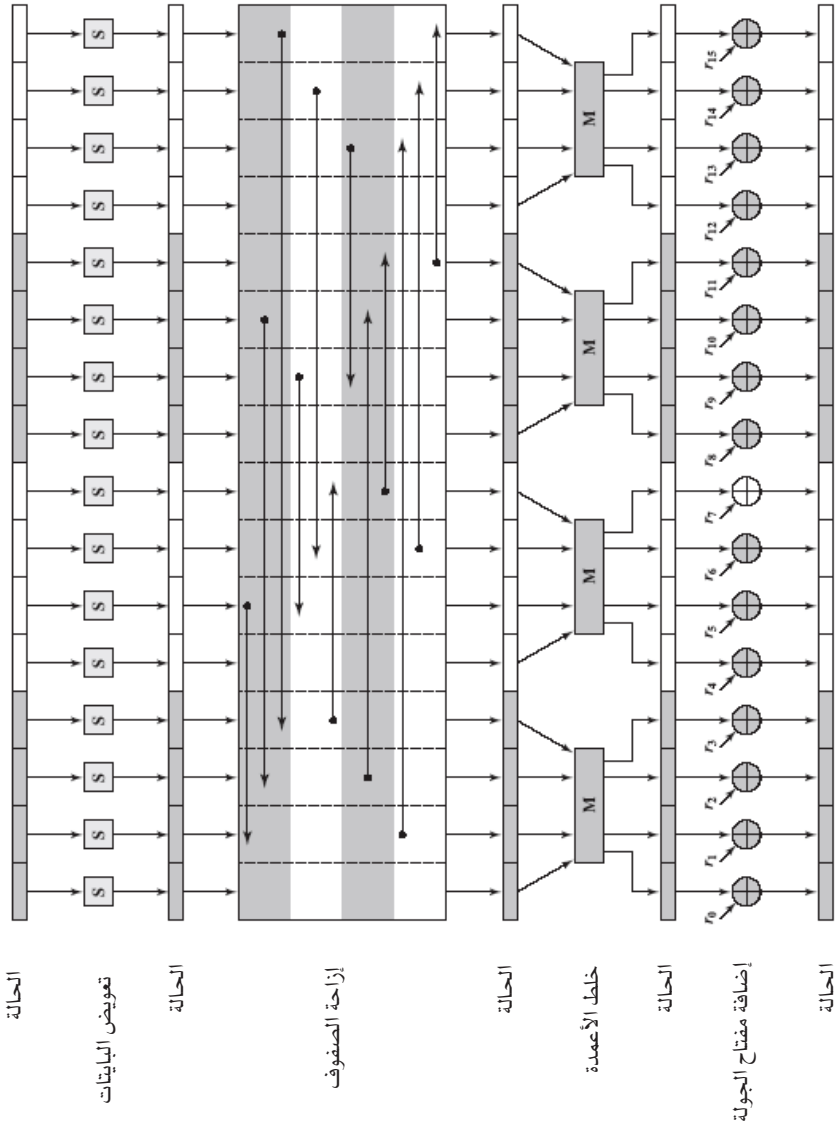
تُلقِي التعليقات الآتية بعض الضوء على خوارزمية AES:

1. من السمات الجديرة بالذكر لهذه الخوارزمية أنها لا تتبع هيكل فيستل. ويُذكر أنه في هيكل فيستل التقليدي تُستخدم نصف كتلة البيانات لتعديل النصف الآخر، ثم يتم بعد ذلك تبديل النصفين. فخوارزمية AES لا تستخدم هيكل فيستل، ولكنها تقوم بمعالجة كتلة البيانات بأكملها على التوازي أثناء كل جولة وذلك باستخدام عمليتي التعويض والتبديل.
2. يتم تمديد المفتاح الذي يقدم كمدخل للخوارزمية إلى صفيف يتألف من 44 كلمة كلٌّ منها (أي $\mathbf{w}[i]$) بطول 32 بتاً. تستخدم كل جولة من جولات المعالجة أربع كلمات مختلفة (128 بتاً) من هذا الصفيف كمفتاح جولة.
3. تستخدم الخوارزمية أربع مراحل مختلفة، واحدة للتبديل وثلاث للتعويض:
   - تعويض البايتات: حيث يُستخدَم جدول، يشار إليه بصندوق التعويض (S-box)[4] لتعويض بايتات كتلة البيانات بايتاً بايتاً.
   - إزاحة الصفوف: تبديل بسيط يتم على الصفوف صفاً صفاً.
   - خلط الأعمدة: تعويض يغيّر كل بايت في العمود كدالة في كل بايتات العمود.
   - إضافة مفتاح الجولة: عملية XOR بسيطة تُجرى على مستوى البتات بين البيانات الحالية للكتلة وجزءٍ من المفتاح الممدد.
4. هيكل بسيط للغاية. لكلٍّ من التشفير وإزالة التشفير، تبدأ الخوارزمية بمرحلة إضافة مفتاح الجولة، وتليها تسع جولات تتكون كل واحدة منها من الأربع مراحل المذكورة، وتعقبها جولة عاشرة تتألف من ثلاث مراحل. ويوضح الشكل 2-6 الهيكل الكامل لعملية التشفير.
5. يُستخدم المفتاح في مرحلة واحدة فقط من المراحل الأربع (مرحلة إضافة مفتاح الجولة). ولهذا السبب فإن الخوارزمية تبدأ وتنتهي بمرحلة إضافة مفتاح الجولة. أي مرحلة أخرى، سواءً طُبّقت في البداية أو في النهاية، قابلة للعكس دون الحاجة لمعرفة المفتاح ومن ثم فإنها لا تضيف أي أمن.

---

[4] يُستخدم المصطلح S-box (صندوق التعويض) بكثرة في وصف أنظمة التشفير المتماثل للإشارة إلى جدول يُستخدم مع آلية للتعويض مبنية على الرجوع إلى جدول لتحديد التعويض المطلوب.

6. يلاحظ أن مرحلة إضافة مفتاح الجولة في حد ذاتها لن تكون صعبة الاختراق. كما أن المراحل الثلاث الأخرى مجتمعة تقوم بخلط البيانات ولكنها بحد ذاتها لا توفّر الأمن لأنها لا تستخدم المفتاح. يمكن النظر للخوارزمية كعمليات متعاقبة من تشفير الكتلة بدالة XOR (إضافة مفتاح الجولة)، يليها خلط بايتات الكتلة (بالمراحل الثلاث الأخرى)، يليها تشفير بدالة XOR وهكذا دواليك. ويتسم هذا النظام بأنه فعّال وآمن للغاية.
7. يمكن عكس كل مرحلة من المراحل الأربع بسهولة. لعكس كلٍّ من مراحل تعويض البايتات، وإزاحة الصف، وخلط الأعمدة، تستخدم دالة عكسية في خوارزمية إزالة التشفير. أما بالنسبة لمرحلة إضافة مفتاح الجولة، فتتم الوظيفة العكسية بعملية XOR لمفتاح الجولة نفسه مع بيانات الكتلة، تطبيقاً للعلاقة: $A \oplus A \oplus B = B$.
8. كما هو الحال مع معظم أنظمة تشفير الكتلة، تستخدم خوارزمية إزالة التشفير المفتاح الممدد بترتيب عكسي. ومع ذلك، فخوارزمية إزالة التشفير ليست مطابقة لخوارزمية التشفير، وذلك نتيجة للطبيعة الخاصة لهيكلة AES.
9. بمجرد التأكد من أن المراحل الأربع جميعاً قابلة للعكس، يسهل التحقق من أن إزالة التشفير تستعيد بالفعل الرسالة الأصلية . يبيّن الشكل 2-5 أن خطوات التشفير وإزالة التشفير تسيران في اتجاهين متضادين رأسياً. عند كل نقطة أفقية (على سبيل المثال، على الخط الأفقي المنقط في الشكل) تكون لعمليتي التشفير وإزالة التشفير نفس الحالة (state).
10. تتألف الجولة الأخيرة من كلٍّ من عمليتي التشفير وإزالة التشفير من ثلاث مراحل فقط. مرة أخرى، يرجع ذلك إلى الطبيعة الخاصة لهيكلة AES، كما أنه مطلوب لجعل عملية التشفير قابلة للعكس.



الشكل 2-6: جولة تشفير ضمن خوارزمية AES.

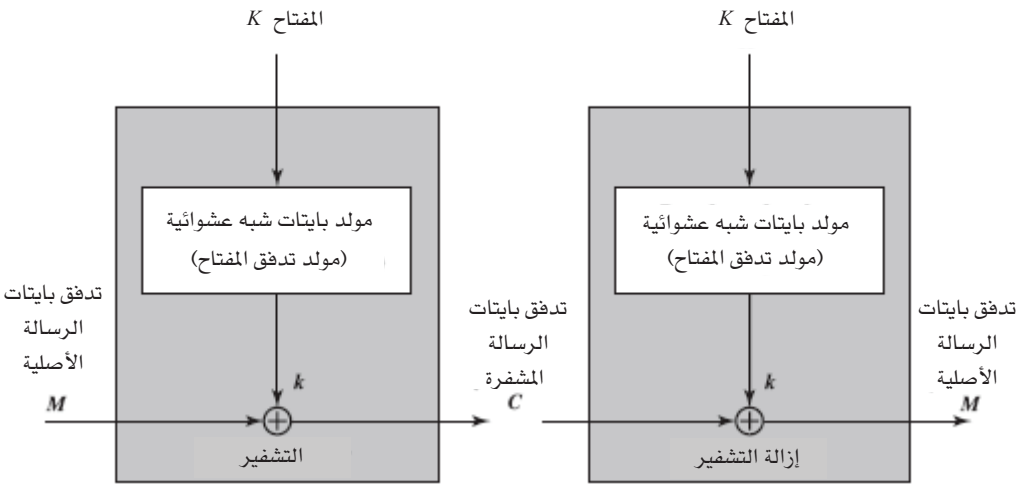
3-2 تشفير التدفق وخوارزمية RC4

يقوم نظام تشفير الكتلة بتشفير عناصر بيانات الكتلة المُدخلة في المرة الواحدة منتِجاً كتلة مُخرج مشفَّرة لكل كتلة مُدخل. أما نظام تشفير التدفق فيقوم بمعالجة عناصر البيانات الداخلة باستمرار عنصراً عنصراً في كل مرة، منتجاً عنصر مُخرج مُشَفَّراً لكل عنصر مُدخل بطريقةٍ مستمرة. ورغم أن تشفير الكتلة يُعدُّ أكثر شيوعاً بكثير، فإن هناك تطبيقات معينة يُعدُّ تشفير التدفق أكثر ملاءمةً لها، وسنورد بعض الأمثلة على ذلك لاحقاً في هذا الكتاب. في هذا الجزء سنلقي نظرةً على نظام RC4 والذي ربما يُعدُّ أشهر أنظمة تشفير التدفق المتماثلة. نبدأ بلمحة عامة عن هيكل تشفير التدفق ثم نتناول خوارزمية RC4 بشيءٍ من التفصيل.

2-3-1 هيكل تشفير التدفق

يقوم نظام تشفير التدفق في العادة بتشفير بايت واحد من الرسالة الأصلية في كل مرة، ومع ذلك فمن الممكن تصميم النظام بحيث يعمل على بت واحد في كل مرة أو على وحدات أكبر من بايت واحد في كل مرة. ويمثل الشكل 2-7 هيكل تشفير التدفق. وفي هذا الهيكل يتم إدخال مفتاح إلى مولد بتات شبه عشوائي (pseudorandom) يُنتِج سلسلة من الأرقام بطول 8 بتات لكلٍّ والتي تبدو عشوائية. التدفق شبه العشوائي هو الذي لا يمكن التنبؤ به بدون معرفة المفتاح الداخل والذي يبدو كما لو كان ذا طابع عشوائي. ويتم دمج مخرج المولّد، والذي يُعرف بتدفق المفتاح keystream، مع تدفق الرسالة الأصلية بمعدل بايت في كل مرة، باستخدام عملية "أو-الحصرية" (XOR) على البتات. على سبيل المثال، إذا كان البايت الخارج من المولد هو 01101100 والبايت المقابل من الرسالة الأصلية هو 11001100، فإن البايت المشفَّر المُخرج في هذه الحالة يكون:

بايت الرسالة الأصلية	11001100
بايت تدفق المفتاح	⊕ 01101100
البايت المشفَّر المُخرج	10100000



الشكل 2-7: هيكل تشفير التدفق.

تتطلب إزالة التشفير استخدام نفس تسلسل الأرقام شبه العشوائي المستخدم أثناء التشفير:

بايت الرسالة المشفَّرة	10100000
بايت تدفق المفتاح	⊕ 01101100
بايت الرسالة الأصلية المُخرج	11001100

يشبه نظام تشفير التدفق نظام حشوة المرة الواحدة (one-time pad). ويكمن الفرق في أن حشوة المرة الواحدة تستخدم سلسلة أرقاماً عشوائيةً بحق، بينما يَستخدم نظام تشفير التدفق أرقاماً شبه عشوائية. ويورد [KUMA97] اعتبارات التصميم الآتية فيما يتعلق بتشفير التدفق:

1. ينبغي أن تكون فترة التكرار لسلسلة التشفير طويلة، حيث يَستخدم مولد الأرقام شبه العشوائية دالةً تنتج سلسلة أرقام محددة تتكرر دورياً. وكلما ازداد طول فترة التكرار، كان من الصعب القيام بتحليل الشفرة.

2. ينبغي أن يقارب تدفق المفاتيح في خصائصه مولِّد أرقام عشوائية حقيقياً بأقرب قدرٍ ممكن. فعلى سبيل المثال، ينبغي أن يكون هناك عددٌ متساوٍ تقريباً من القيم 0 و1. وإذا نظرنا إلى تدفق المفاتيح كسلسلة من البايتات، فينبغي أن يتكرر ظهور كل قيم البايتات الـ 256 نفس عدد المرات تقريباً. وكلما كان تدفق المفاتيح أكثر عشوائيةً، كلما كانت الرسالة المشفَّرة أكثر عشوائية، مما يزيد من صعوبة عملية تحليل الشفرة.
3. نلاحظ من الشكل 2-7 أن الناتج من مولد الأرقام العشوائية يعتمد على قيمة المفتاح المستخدم. للحماية ضد الهجمات الاستقصائية، ينبغي أن يكون المفتاح طويلاً بدرجة كافية. وتنطبق هنا أيضاً نفس الاعتبارات التي تنطبق على أنظمة تشفير الكتلة. وعليه، فباستخدام التقنية المتوفِّرة حالياً يُنصح باستخدام مفتاح لا يقل طوله عن 128 بتاً.

باستخدام مولد أرقام عشوائية مصمم جيداً، يمكن الحصول على نظام تشفير تدفق بنفس درجة أمان نظام تشفير كتلة له نفس طول المفتاح تقريباً. وتكمن الميزة الرئيسة لأنظمة تشفير التدفق على أنظمة تشفير الكتلة في كونها أسرع وأنها تستخدم كوداً أصغر بكثير. فخوارزمية RC4 مثلاً التي نتناولها في هذا الجزء يمكن تنفيذها في بضعة أسطر من الكود. يَستخدم الجدول 2-3 بيانات من [RESC01] للمقارنة بين سرعة تنفيذ RC4 وثلاث خوارزميات أخرى مشهورة لتشفير الكتلة. غير أن تشفير الكتلة يمتاز بإمكانية إعادة استخدام المفاتيح، أما في حالة تشفير التدفق فإن تشفير رسالتين عاديتين باستخدام نفس المفتاح يؤدي غالباً إلى جعل عملية تحليل الشفرة أمراً بسيطاً للغاية [DAWS96]. عند إجراء عملية XOR على تدفقي نصّين مشفَّرين تكون النتيجة هي نفس ناتج إجراء XOR على تدفقي النصّين الأصليين قبل التشفير. فإذا كان النص غير المشفَّر سلسلة من الحروف، أو أرقام بطاقات الائتمان، أو غيرها من سلسلة البايتات التي تحكمها قواعد معروفة، فقد تنجح عمليات التحليل في فك الشفرة.

الجدول 2-3: مقارنة سرعات تنفيذ التشفير المتماثل على معالج Pentium II.

نظام التشفير	طول المفتاح	السرعة (ميجابت/ثانية)
DES	56	9
3DES	168	3
RC2	متغير	0.9
RC4	متغير	45

للتطبيقات التي تتطلب تشفير/إزالة تشفير بيانات متدفقة، كتلك التي تمر عبر قناة اتصال أو وصلة متصفح/إنترنت، قد يكون تشفير التدفق هو البديل الأفضل. أما في التطبيقات التي تتعامل مع كتل من البيانات؛ كنقل الملفات والبريد الإلكتروني وقواعد البيانات، فقد يكون من الأنسب استخدام تشفير الكتلة. ومع ذلك، فمن الممكن استخدام كلا الأسلوبين للتشفير في أي تطبيق تقريباً.

2-3-2 خوارزمية RC4

خوارزمية RC4 هي خوارزمية تشفير تدفق صمّمها رون ريفيست من شركة RSA Security في عام 1987. تستخدم الخوارزمية مفتاحاً بطول متغير، وتجري عملياتها على البايتات، وهي مبنية على أساس التبديل العشوائي. تبيّن التحاليل أن فترة التكرار للخوازمية طويلة للغاية، حيث تزيد على 10^{100} [ROBS95a]. تتضمَّن الخوارزمية إجراء ما بين 8 و16 عملية آلة (machine operations) لإنتاج كل بايت في المخرج، ويُتوقع أن تعمل RC4 بسرعات عالية عند تنفيذها برمجياً. تُستخدم RC4 في معايير SSL/TLS (Secure Sockets Layer/Transport Layer Security) (طبقة المقابس الآمنة/ أمن طبقة النقل) والتي تم تحديدها للاتصال بين المتصفحات والخوادم على شبكة الإنترنت. كما أنها تستخدم في بروتوكول WEP (الخصوصية المكافئة للشبكات السلكية Wired Equivalent Privacy) والبروتوكول الأحدث WPA (الوصول اللاسلكي المحمي Wireless Protected Access) واللذان يشكّلان جزءاً من معيار IEEE 802.11 للشبكات اللاسلكية المحلية. احتفظت شركة RSA

Security بخوارزمية RC4 كسرٌ تجاري خاص بها. غير أنه في سبتمبر 1994 قام مجهولون بوضع الخوارزمية على قائمة Cypherpunks الخاصة بممرّري البريد المجهولين على شبكة الإنترنت.

تمتاز خوارزمية RC4 بالبساطة الشديدة وبسهولة شرحها. ويُستخدم مفتاح بطول متغير من 1 إلى 256 بايتاً (8 إلى 2048 بتاً) لتهيئة متجه الحالة S (state vector) بطول 256 بايتاً، والذي يتألف من العناصر S[0], S[1], . . ., S[255]. في جميع الأوقات، ويحتوي المتجه S على تبديلة من كل الأرقام بطول 8 بتات (من 0 إلى 255). للقيام بعملية التشفير وإزالة التشفير يتم توليد بايت k (انظر الشكل 2-7) من المتجه S باختيار أحد العناصر الـ 256 بشكل نظامي. وفي كل مرة يتم توليد قيمة جديدة للبايت k يتم أيضاً تبديل عناصر المتجه S مرة أخرى.

❖ تهيئة متجه الحالة S:

لبدء تشغيل الخوارزمية يتم تحديد قيم عناصر متجه الحالة S لتأخذ القيم من 0 إلى 255 بترتيب تصاعدي؛ أي S[0] = 0, S[1] = 1, … , S[255] = 255. كما يتم توليد متجه مؤقت جديد T. إذا كان طول المفتاح K هو 256 بايتاً يتم نقل K بأكمله إلى T، وإلا فإنه لمفتاح طوله keylen بايتاً يتم نسخ أول keylen عنصراً من K إلى T ثم يكرر K عدة مرات حسب الحاجة لملء T. ويمكن تلخيص هذه العمليات الأولية على النحو الآتي:

```
/* التهيئة */
for i = 0 to 255 do
    S[i] = i;
    T[i] = K[i mod keylen];
```

بعد ذلك نستخدم T لإنتاج التبديلة الأولية لـ S. ويتم ذلك بالبدء من S[0] والاتجاه إلى S[255]، ولكل قيمة S[i] يتم تبديل S[i] ببايت أخرى من S تبعاً لنظام تحدده T[i]:

```
/* S التبديلة الأولية لـ */
j = 0;
for i = 0 to 255 do
      j = (j + S[i] + T[i]) mod 256;
      Swap (S[i], S[j]);
```

ونظراً إلى أن العملية الوحيدة المستخدمة هي عملية تبادل، فإن الأثر الوحيد هو تبديل S. في نهاية العملية سيبقى S يحتوي على جميع الأرقام من 0 إلى 255.

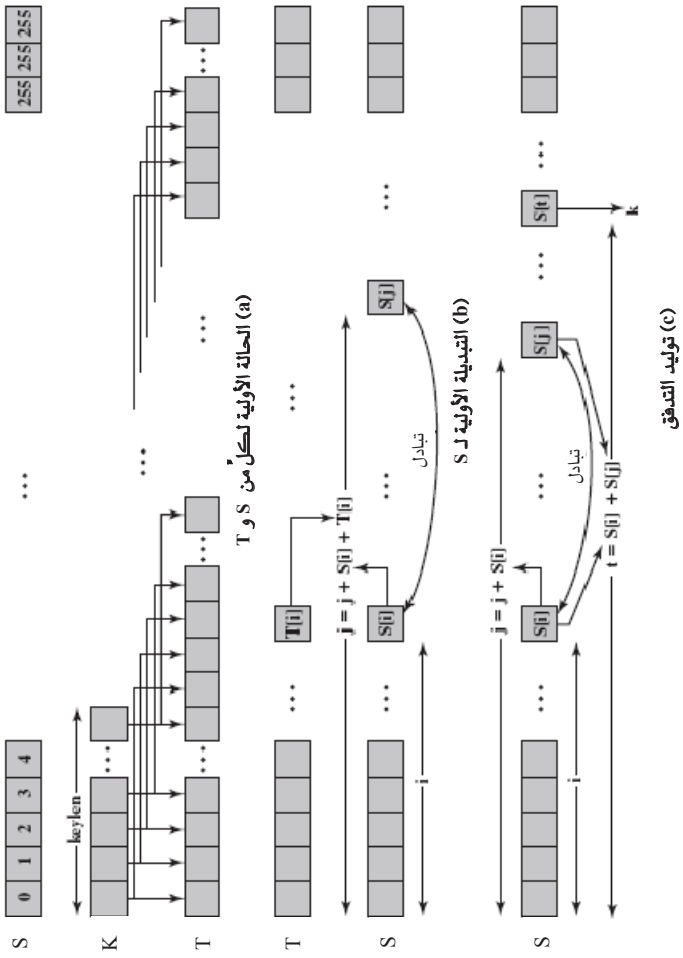
❖ توليد التدفق:

بعد الانتهاء من تهيئة المتجه S لا تكون هناك حاجة لاستخدام المفتاح المُدخل بعد ذلك. ويتضمَّن توليد التدفق التدوير عبر كل العناصر S[i]، ولكل عنصر S[i] يتم تبديل S[i] ببايت آخر من S تبعاً لنظام تحدده الترتيبة الحالية لـ S. بعد الوصول إلى S[255] تستمر العملية بدءاً بـ S[0] من جديد:

```
/* توليد التدفق */
i, j = 0;
while (true)
   i = (i + 1) mod 256;
   j = (j + S[i]) mod 256;
   Swap (S[i], S[j]);
   t = (S[i] + S[j]) mod 256;
   k = S[t];
```

في عملية التشفير، تُجرى عملية XOR بين قيمة k والبايت التالي من الرسالة الأصلية. لإزالة التشفير تُجرى عملية XOR بين قيمة k والبايت التالي من الرسالة المشفَّرة.

ويوضح الشكل 2-8 منطق خوارزمية RC4.



الشكل 2-8: منطق خوارزمية RC4.

❖ **قوة خوارزمية RC4:**

نُشرت عدة أبحاث حول تحليل أساليب مهاجمة RC4، منها على سبيل المثال [KNUD98]، [MIST98]، [FLUH00]، [MANT01]، [PUDO02]، [PAUL03]، [PAUL04]. لم تثبت فعالية أيٍّ من أساليب الهجوم تلك ضد خوارزمية RC4 بمفتاح طوله 128 بتاً. يتناول [FLUH01] مشكلة أكثر خطورة، حيث يوضح المؤلفون أن بروتوكول WEP، والمفترض أنه يوفر السرية على شبكات 802.11 اللاسلكية المحلية، يكون عرضة لهجوم معين. وحقيقة الأمر أن المشكلة ليست في خوارزمية RC4 في حد ذاتها ولكن في الطريقة التي يتم بها توليد المفاتيح المستخدمة كمُدخل إلى RC4. ولا يبدو أن هذه المشكلة خطيرة فيما يتعلق بالتطبيقات الأخرى التي تستخدم RC4، كما أنه يمكن علاجها في بروتوكول WEP بتغيير الطريقة المستخدَمة لتوليد المفاتيح. وتوضح هذه المشكلة الصعوبات التي تكتنف تصميم نظام آمن يتضمَّن كلاً من وظائف التشفير والبروتوكولات التي تستخدمها.

4-2 أنماط التشغيل في تشفير الكتل

في تشفير الكتل تتم معالجة كتلة بياناتٍ واحدةٍ في كل مرة، ويبلغ طول الكتلة 64 بتاً في كلٍّ من DES و3DES. للتعامل مع رسائل أصلية أطول من ذلك، يتعين تقسيم الرسالة الأصلية إلى كتل يضم كلٌّ منها 64 بتاً (وحشو الكتلة الأخيرة عند اللزوم لتصبح 64 بتاً). وأبسط طريقة لعمل ذلك هي استخدام ما يعرف بنمط دفتر الشفرة الإلكتروني (electronic codebook (ECB))، حيث تعالَج كتلة من الرسالة الأصلية طولها 64 بتاً في كل مرة ويتم تشفير كل كتلة باستخدام نفس المفتاح. يُستخدم مصطلح codebook (دفتر الشفرة) هنا لأنه – لنفس المفتاح المُستخدَم – تُنتِج كل كتلة (64 بتاً) من الرسالة الأصلية رسالةً مشفَّرةً فريدة. وعليه، يمكن للمرء تخيل دفتر شفرة هائل يضم عنصر رسالة مشفَّرة (64 بتاً) لكل نمط ممكن من قيم الـ 64 بتاً التي يمكن أن تمثل كتلة من كتل الرسالة الأصلية.

في نمط ECB للتشغيل، إذا تكرر ظهور نفس الكتلة التي تضم 64 بتاً أكثر من مرة في الرسالة الأصلية، فإنها تنتج دائماً نفس الرسالة المشفَّرة. ولهذا السبب، فإن نمط ECB قد لا يوفر وسيلة تشفير آمنة للرسائل الطويلة. إذا كانت الرسالة منتظمة بدرجة كبيرة، فقد يتمكن محلل الشفرة من استغلال هذا الانتظام. على سبيل المثال، إذا كان معروفاً أن الرسالة تبدأ دائماً ببعض الحقول المحدّدة مسبقا، فقد يُتاح لمحلل الشفرة عددٌ من الأزواج المعروفة من الرسالة الأصلية والرسالة المشفَّرة المناظرة للعمل عليها. إذا تضمَّنت الرسالة عناصر متكررة بفترة تكرار تبلغ مضاعفات 64 بتاً، فسيكون بوسع محلل الشفرات تحديد تلك العناصر، مما قد يساعد في تحليل الشفرة أو يوفر فرصة لاستبدال أو إعادة ترتيب كتل البيانات في الرسالة. للتغلب على أوجه القصور الأمني لنمط ECB، يُفضل استخدام أسلوب يضمن أنه إذا ما تكررت نفس الكتلة من الرسالة الأصلية، فإنها تنتج كتلاً مختلفة من الرسالة المشفَّرة. وسنتناول في هذا الجزء أسلوبين يحققان هذا الشرط تم تعريفهما في الوثيقة FIPS PUB 81.

1-4-2 نمط سَلسَلة كتل الشفرة

في نمط سَلسَلة كتل الشفرة (cipher block chaining (CBC)) يدخل إلى خوارزمية التشفير ناتج إجراء عملية XOR على الكتلة الحالية من الرسالة الأصلية والكتلة السابقة من الرسالة المشفَّرة، ويُستخدم المفتاح نفسه لكل كتلة (انظر الشكل 2-9). ويؤدي ذلك في واقع الأمر إلى ربط معالجة الكتل المتعاقبة للرسالة الأصلية. فكتلة البيانات الداخلة لخوارزمية التشفير ليس لها علاقة ثابتة بالكتلة المناظرة من الرسالة الأصلية، ومن ثم فلن يكشف التشفير بهذا النمط عن تكرار أنماط الـ 64 بتاً في الرسالة الأصلية.

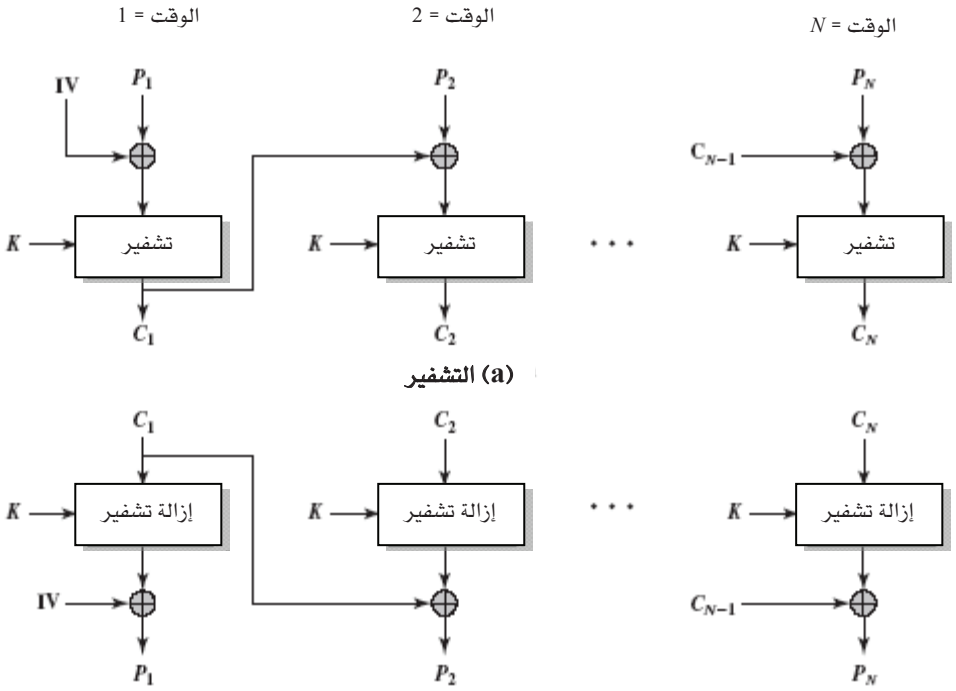
في عملية إزالة التشفير تُمرَّر كل كتل الشفرة عبر خوارزمية إزالة التشفير. تُجرى عملية XOR على النتيجة مع الكتلة السابقة من الرسالة المشفَّرة لإنتاج كتلة الرسالة الأصلية. للتحقق من أن ذلك يعطي النتيجة الصحيحة، يمكننا كتابة:

$$C_j = E(K, [C_{j-1} \oplus P_j])$$

حيث E[K,X] هو تشفير الرسالة الأصلية X باستخدام مفتاح K، والرمز ⊕ يمثل عملية "أو-الحصرية")، ومن ثم:

$$D(K, C_j) = D(K, E(K, [C_{j-1} \oplus P_j]))$$
$$D(K, C_j) = C_{j-1} \oplus P_j$$
$$C_{j-1} \oplus D(K, C_j) = C_{j-1} \oplus C_{j-1} \oplus P_j = P_j$$

مما يثبت صحة الشكل 2-9.



الشكل 2-9: نمط سلسلة كتل الشفرة (CBC).

لإنتاج أول كتلة من الرسالة المشفَّرة تُجرى عملية XOR بين متجه أوّلي (Initialization Vector (IV)) والكتلة الأولى من الرسالة الأصلية. في حالة إزالة التشفير، تُجرى عملية XOR بين المتجه الأوّلي IV وناتج خوارزمية إزالة التشفير للحصول على أول كتلة من الرسالة المشفَّرة.

يجب أن يكون المتجه IV معروفاً لكلٍّ من المرسل والمستلم. وينبغي حماية كلٍّ من IV والمفتاح لتحقيق أقصى قدر من الأمن. ويمكن أن يتم ذلك بإرسال IV باستخدام نمط ECB للتشفير. من أسباب ضرورة حماية IV الآتي: إذا استطاع الخَصْم خداع المستلم بحيث يَستخدم قيمة مختلفة لـ IV، فسيكون بوسع الخَصْم قلب (عكس) بتات مختارة في أول كتلة من الرسالة الأصلية التي يتلقاها المستلم، لإثبات ذلك، خذ بعين الاعتبار ما يأتي:

$$C_1 = \mathrm{E}(K, [\mathrm{IV} \oplus P_1])$$
$$P_1 = \mathrm{IV} \oplus \mathrm{D}(K, C_1)$$

باستخدام $X[j]$ للدلالة على البت j ضمن الـ 64 بتاً التي تضمها الكمية X يمكننا كتابة

$$P_1[i] = \mathrm{IV}[i] + \mathrm{D}(K, C_1)[i]$$

وباستخدام خصائص العملية XOR، يمكننا القول بأن:

$$P_1[i]' = \mathrm{IV}[i]' + \mathrm{D}(K, C_1)[i]$$

حيث تشير "′" لعملية قلب البت (bit complementation). يعني ذلك أنه إذا تمكن الخَصْم من قلب بتات بعينها في المتجه الأوّلي IV فسيتمكن بذلك من قلب (أي تغيير) البتات المناظرة في كتلة الرسالة الأصلية التي يسترجعها المستلم.

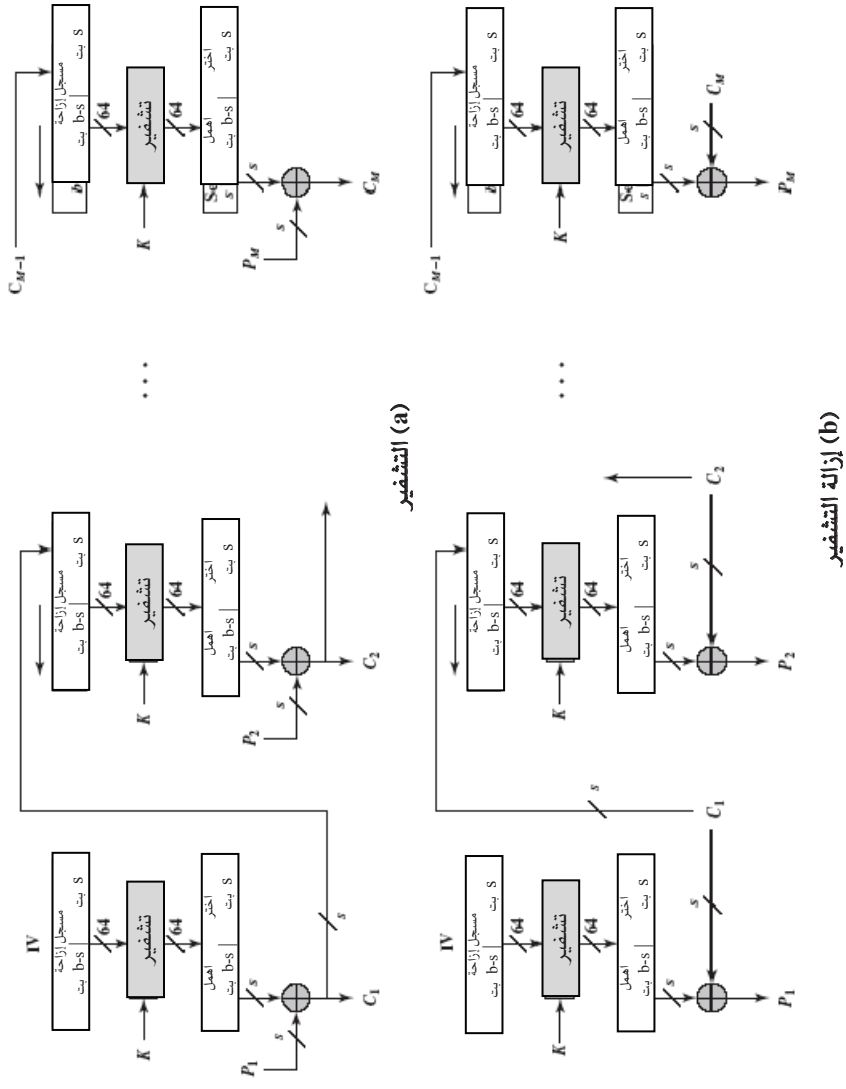
يُستخدم نمط CBC على نطاق واسع في تطبيقات الأمن، كما سنرى في الباب الثاني من الكتاب.

2-4-2 نمط التشفير بالتغذية المرتجعة

من الممكن تحويل أي خوارزمية تشفير كتلة إلى خوارزمية تشفير تدفق باستخدام نمط التشفير بالتغذية المرتجعة (Cipher Feedback (CFB) mode). باستخدام تشفير التدفق لا نحتاج لحشو الرسالة الأصلية لتصبح عدداً كاملاً من الكتل، كما أنه يمكن استخدامها للعمل في الوقت الحقيقي. وعليه، فعند إرسال سلسلة من الحروف، سيكون بالإمكان تشفير كل حرف وإرساله في الحال باستخدام نظام تشفير تدفق يعمل على الحروف.

من المزايا المرغوبة لنظام تشفير التدفق أن تكون الرسالة المشفَّرة بنفس طول الرسالة الأصلية. وعليه، فعند إرسال حروف يتألف كلٌّ منها من 8 بتات ينبغي تشفير كل حرف باستخدام 8 بتات. إذا استخدم النظام أكثر من 8 بتات فإن ذلك يمثل إهداراً لسعة الإرسال.

يمثل الشكل 2-10 نظام التشفير بنمط CFB. يُفترض في الشكل أن وحدة الإرسال هي s بت، ومن القيم الشائعة $s = 8$. كما في نمط CBC، تتم سَلسَلة وحدات بيانات الرسالة الأصلية معاً بحيث تعتمد الرسالة المشفَّرة لأي وحدة من الرسالة الأصلية على كل الوحدات السابقة للرسالة الأصلية.



الشكل 2-10: نمط التغذية المرتجعة للشفرة (CBF) بـ *s* بت.

خذ بعين الاعتبار أولاً عملية التشفير. يتألف المُدخل إلى خوارزمية التشفير من سِجِل إزاحة (shift register) طوله 64 بتاً يُملأ في البداية بمتجه أوّلي (IV). تُجرى عملية XOR بين الـ s بت في أقصى اليسار (الأعلى وزناً) من مُخرج دالة التشفير والوحدة الأولى من الرسالة الأصلية P_1 لإنتاج الوحدة الأولى من الرسالة المشفَّرة C_1 والتي يتم إرسالها. بالإضافة إلى ذلك، تتم إزاحة محتويات سِجِل الإزاحة s خانة إلى اليسار وتوضع C_1 في الـ s بت بأقصى اليمين (الأدنى وزناً) من سِجِل الإزاحة. وتستمر هذه العملية حتى يتم تشفير جميع وحدات الرسالة الأصلية .

للقيام بإزالة التشفير، تُتبع نفس الطريقة، غير أنه يتم إجراء عملية XOR بين الوحدة التي يتم استلامها من الرسالة المشفَّرة ومُخرج دالة التشفير لإنتاج وحدة من الرسالة الأصلية. لاحظ أن المُستخدَم هنا هو دالة التشفير وليس دالة إزالة التشفير. ويمكن تفسير ذلك بسهولة. افترض أن $S_s(X)$ ترمز للـ s بت الأعلى وزناً في X، ومن ثم فإن

$$C_1 = P_1 \oplus S_s[E(K, IV)]$$

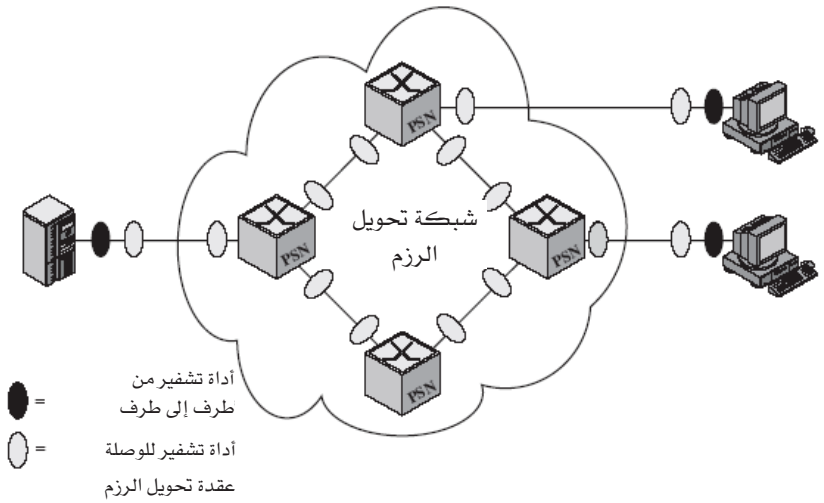
ولذلك فإن

$$P_1 = C_1 \oplus S_s[E(K, IV)]$$

وينطبق نفس المنطق على الخطوات التي تلي ذلك في تلك العملية.

5-2 مواضع أجهزة التشفير

يُعدُّ التشفير أقوى الأساليب وأكثرها شيوعاً لمواجهة الأخطار التي تهدد أمن الشبكات. وعند استخدام التشفير علينا أن نقرر ما الذي نشفره؟ وأين سنضع أجهزة التشفير؟ هناك بديلان أساسيان: تشفير الوصلة، والتشفير من طرف إلى طرف. يوضح الشكل 2-11 استخدام هذين الأسلوبين عبر شبكة تحويل الرزم.



الشكل 2-11: التشفير عبر شبكة تحويل الرزم.

في تشفير الوصلة تزوّد كل الوصلات المعرّضة للهجوم بأداة تشفير على كلٍّ من طرفي الوصلة، وبذلك يتم تأمين كل حركة مرور للبيانات على جميع وصلات الاتصال. ورغم أنه في شبكة كبيرة يتطلب ذلك الكثير من أجهزة التشفير، فإنه يوفر مستوى عالٍ من الأمن. ولكن من عيوب هذا الأسلوب أنه يتعين إزالة التشفير في كل مرة تدخل فيها الرسالة إلى محول رزم، وذلك كي يتسنى للمحول قراءة العنوان (رقم الدائرة الافتراضية) الموجود في ترويسة الرزمة حتى يتمكن من توجيهها باتجاه وجهتها النهائية. وهكذا تكون الرسالة عرضةً للهجوم عند كل محول رزم. واذا كانت الشبكة شبكة عامة لتحويل الرزم فلن يكون للمستخدم أي سيطرة على الأمن في عقد الشبكة.

في التشفير من طرف إلى طرف تتم عملية التشفير وإزالة التشفير في الأنظمة الطرفية. ويقوم المصدر أو الطرف المرسِل (مضيف أو محطة) بتشفير البيانات. ويتم إرسال البيانات بصيغتها المشفَّرة، ومن ثم تنتقل عبر الشبكة دون تغيير إلى وجهتها النهائية (محطة أو مضيف). يشترك طرف الوجهة النهائية مع طرف المصدر في مفتاح سري، ومن ثم يتسنى له إزالة تشفير البيانات. وقد يبدو أن هذا الأسلوب يؤمّن

انتقال البيانات ضد هجمات وصلات الشبكة ومحولاتها. ومع ذلك، فلا تزال هناك نقطة ضعف.

خذ الوضع التالي بعين الاعتبار. يرتبط مضيف بشبكة X.25 لتحويل الرزم، وينشئ دائرة إفتراضية إلى مضيف آخر، ويتأهب لنقل البيانات إلى ذلك المضيف باستخدام التشفير من طرف إلى طرف. وتُرسَل البيانات عبر تلك الشبكة على شكل رزم يتألف كلٌّ منها من ترويسة في المقدمة تتبعها بعضٌ من بيانات المستخدم. أي جزء من كل رزمة سيقوم طرف المصدر بتشفيره؟ وإذا افترضنا أن المصدر يقوم بتشفير الرزمة بكاملها، بما في ذلك الترويسة فلن تفلح هذه الطريقة، لأنه كما ذكرنا فإن طرف الوجهة فقط هو الذي يمكنه إزالة التشفير. لذا فستتلقى عقدة تحويل الرزم رزمة مشفَّرة وسيتعذر عليها قراءة الترويسة، ومن ثَمَّ لن تكون قادرة على توجيه الرزمة. ويترتب على ذلك أن طرف المصدر يمكنه تشفير الجزء من الرزمة الخاص ببيانات المستخدِم فقط، ويجب أن يترك الترويسة غير مشفَّرة بحيث يمكن قراءتها بواسطة الشبكة.

وهكذا، فباستخدام التشفير من طرف إلى طرف يتم تأمين بيانات المُستخدِم، أما نمط حركة البيانات فيبقى غير مؤمّن، حيث تنقل ترويسات الرزم واضحة بدون تشفير. لتحقيق قدر أكبر من الأمن، نحتاج إلى كلٍّ من تشفير الوصلات والتشفير من طرف إلى طرف، كما هو مبين في الشكل 2-11.

باختصار، عند استخدام كلٍّ من أسلوبي التشفير معاً، يقوم المضيف بتشفير جزء الرزمة الخاص ببيانات المستخدِم باستخدام مفتاح للتشفير من طرف إلى طرف. ويتم بعد ذلك تشفير الرزمة بكاملها باستخدام مفتاح لتشفير الوصلة. بينما تعبر الرزمة الشبكة انطلاقاً من المصدر إلى الوجهة يقوم كل محول رزم بإزالة تشفير الرزمة باستخدام مفتاح تشفير الوصلة لقراءة الترويسة ثم يعيد تشفير الرزمة بكاملها لإرسالها من جديد على الوصلة التالية. وبهذه الطريقة تُعدُّ الرزمة بكاملها مؤمّنة باستثناء الوقت الذي تكون فيه الرزمة بالفعل في ذاكرة محول الرزم، حيث تكون ترويسة الرزمة عندئذٍ غير مشفَّرة.

6-2 توزيع المفاتيح

لكي يُستخدم التشفير المتماثل بنجاح، ينبغي أن يشترك طرفا عملية تبادل البيانات في نفس المفتاح، وأن يتم توفير الحماية لذلك المفتاح من وصول الآخرين إليه. علاوة على ذلك، عادةً ما يُنصح بتغيير المفتاح من حين لآخر للحد من كمية البيانات التي قد تتعرض للانكشاف إذا استطاع المهاجمون معرفة المفتاح. لذا، فإن قوة أي نظام للتشفير ترتكز على أسلوب توزيع المفاتيح، وهو المصطلح الذي يشير إلى وسيلة توصيل المفتاح للطرفين اللذين يرغبان في تبادل البيانات، بدون السماح لغيرهما بالاطلاع على المفتاح. ويمكن تحقيق توزيع المفاتيح للطرفين A وB بعدد من الطرق منها:

1. يمكن أن يختار الطرف A مفتاحاً ويتم توصيل ذلك المفتاح مادياً إلى الطرف B.
2. يمكن أن يقوم طرف ثالث باختيار المفتاح وتوصيله مادياُ إلى كلٍّ من A وB.
3. إذا كان قد سبق لأيٍّ من A وB استخدام مفتاح جديد مؤخراَ، يمكن لأيٍّ من الطرفين إرسال المفتاح الجديد للطرف الآخر، مشفَّراً باستخدام المفتاح القديم.
4. إذا توافر لكلٍّ من A وB توصيلة مشفَّرة مع طرف ثالث C، يمكن للطرف C توصيل مفتاح على الوصلتين المشفَّرتين إلى كلٍّ من A وB.

يتطلب الخياران 1 و2 تسليم المفتاح يدوياً، الأمر الذي يُعدُّ مطلباً معقولاً في حالة تشفير الوصلة حيث ستقوم أداة التشفير على الوصلة فقط بتبادل البيانات مع الشريك على الطرف الآخر من الوصلة. أما فيما يتعلق بالتشفير من طرف إلى طرف، فإن التسليم اليدوي للمفاتيح يُعدُّ مزعجاً. في نظام موزّع، قد يحتاج أي مضيف أو طرف لتبادل بيانات مع العديد من الأنظمة المضيفة والأطراف الأخري. وعليه، فإن كل جهاز سيحتاج إلى عدد من المفاتيح يتم توفيرها بشكل ديناميكي وتكون هذه المشكلة أصعب بشكلٍ خاص في حالة نظام موزَّع يغطى منطقة واسعة.

الخيار 3 ممكن مع كلٍّ من تشفير الوصلة والتشفير من طرف إلى طرف، ولكن إذا حدث ونجح مهاجم في الوصول إلى أحد المفاتيح فسوف تنكشف كل المفاتيح التي تليه. حتى لو تم تغيير مفاتيح تشفير الوصلة بصورة متكررة، فينبغي أن يتم ذلك يدوياً. لتوفير المفاتيح للتشفير من طرف إلى طرف، يفضل استخدام الخيار 4.

يوضح الشكل 2-12 طريقة لتحقيق متطلبات الخيار 4 للتشفير من طرف إلى طرف. في هذا الشكل، تم إهمال تشفير الوصلة، والذي يمكن إضافته أو عدم إضافته حسب الحاجة. في هذه الترتيبة يمكن تحديد نوعين من المفاتيح:

- مفتاح الجلسة: عندما يرغب نظامان طرفيان (مثلاً: من المضيفات أو المحطات الطرفية) في الاتصال فيما بينهما، فإنهما يقيمان توصيلة منطقية (مثلاً: دائرة افتراضية). يتم تشفير كل بيانات المستخدِم بمفتاح جلسة يُعطَى مرة واحدة طوال فترة وجود تلك التوصيلة المنطقية، وفي ختام الجلسة أو التوصيلة يتم التخلص من المفتاح.
- المفتاح الدائم: وهو مفتاح للاستخدام بين الكيانات بغرض توزيع مفاتيح الجلسات.

تتألف الترتيبة من العناصر الآتية:

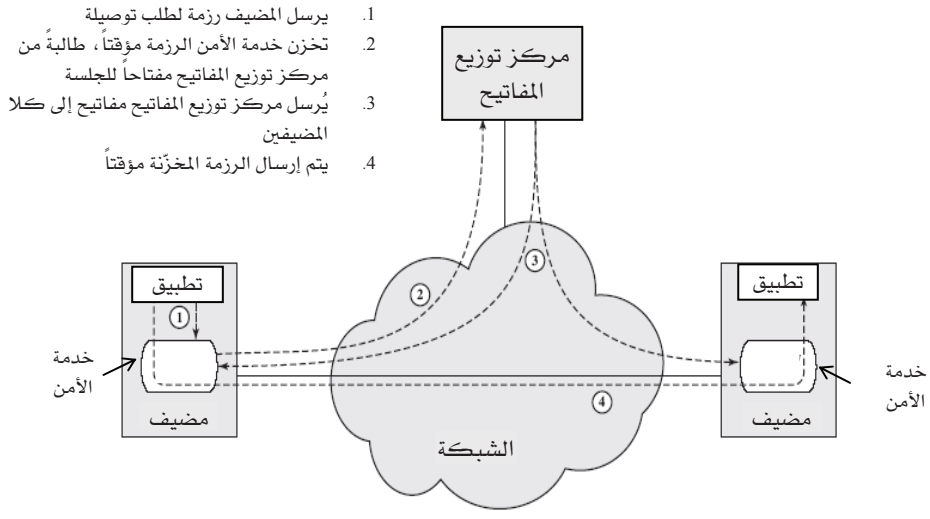
- مركز توزيع المفاتيح: يقوم مركز توزيع المفاتيح (Key Distribution Center (KDC)) بتحديد أي النظم يُسمح لها بالاتصال مع بعضها بعضاً. عند التصريح لنظامين بإنشاء توصيلة، يوفر مركز توزيع المفاتيح مفتاح جلسة لتلك التوصيلة للاستخدام لمرة واحدة فقط.
- وحدة خدمة الأمن (Security Service Module (SSM)): تقوم هذه الوحدة، والتي قد تتكون من وظائف في بروتوكول طبقة واحدة، بعملية التشفير من طرف إلى طرف وتحصل على مفاتيح الجلسات نيابة عن المستخدِمين.

يوضح الشكل 2-12 الخطوات اللازمة لإنشاء توصيلة. عندما يرغب مضيف في إنشاء توصيلة مع مضيف آخر، فإنه يقوم بإرسال رزمة طلب توصيلة (الخطوة 1).

تقوم وحدة SSM بتخزين تلك الرزمة وتتقدم إلى مركز KDC بطلب إنشاء توصيلة (الخطوة 2). يتم تشفير الاتصال بين SSM وKDC باستخدام مفتاح رئيس يشترك فيه فقط وحدة SSM تلك ومركز KDC. إذا وافق مركز KDC على طلب التوصيلة، فإنه يولد مفتاح جلسة ويوصّله إلى كلٍّ من وحدتي SSM المعنيتين، وذلك باستخدام مفتاح دائم مخصص لكل وحدة (الخطوة 3). تقوم الآن وحدة SSM التي طلبت التوصيلة بسحب رزمة طلب التوصيلة، ومن ثم يتم إنشاء توصيلة بين النظامين الطرفيين (الخطوة 4). يتم تشفير كل البيانات المتبادلة بين النظامين الطرفيين بواسطة وحدة SSM الخاصة بكلٍّ منهما وذلك باستخدام مفتاح الجلسة لمرة واحدة.

توفّر الآلية الأتوماتيكية لتوزيع المفاتيح مزايا المرونة والديناميكية اللازمة للسماح لعدد من المُستخدِمين من محطات طرفية بالوصول إلى عدد من المضيفين، وكذلك للمضيفين بتبادل البيانات بين بعضهم بعضاً.

هناك طريقة أخرى لتوزيع المفاتيح تستخدم التشفير بالمفاتيح العامة، وسوف نرجئ مناقشتها إلى الفصل الثالث.



الشكل 2-12: التوزيع الأتوماتيكي للمفاتيح لبروتوكول مبنيٍّ على استخدام توصيلة.

7-2 توصيات للمطالعة

يوفّر [STAL06a] تغطيةً أكثر عمقاً للمواضيع المدرجة في هذا الفصل. يُعدّ [SCHN96] مرجعاً أساسياً لتغطية خوارزميات التشفير، حيث يتضمَّن وصفاً لكل خوارزمية أو بروتوكول تشفير نُشرت حتى وقت كتابة ذلك الكتاب تقريباً. يوجد مسح تفصيلي آخر جدير بالاهتمام في [MENE97]. يوفر [STIN06] معالجة أكثر عمقاً مع مناقشة رياضية محكمة.

[FEIS73] Feistel, H. "Cryptography and Computer Privacy." *Scientific American*, May 1973.

[MENE97] Menezes, A.; van Oorschot, P.; and Vanstone, S. *Handbook of Applied Cryptography*. Boca Raton, FL: CRC Press, 1997.

[SCHN96] Schneier, B. *Applied Cryptography*. New York: Wiley, 1996.

[STAL06a] Stallings, W. *Cryptography and Network Security: Principles and Practice*, Fourth Edition. Upper Saddle River, NJ: Prentice Hall, 2006.

[STIN06] Stinson, D. *Cryptography: Theory and Practice*. Boca Raton, FL: CRC Press, 2006.

8-2 مصادر للمعلومات على الويب

- صفحة AES على الويب: تتضمَّن صفحة AES على موقع NIST وثيقة المعيار بالإضافة إلى عدد من الوثائق الأخرى ذات الصلة.
- صالون AES على الويب: يتضمَّن قوائم شاملة للمصادر تحتوي على الوثائق والأبحاث عن AES مع إمكانية للحصول على نسخ إلكترونية منها.
- أنماط تشغيل أنظمة تشفير الكتلة: صفحة على موقع NIST تتضمَّن معلومات وافية عن أنماط التشغيل المعتمدة من قِبَل NIST.

9-2 مصطلحات رئيسة

Advanced Encryption Standard (AES)	معيار التشفير المتقدم (AES)
block cipher	تشفير الكتلة
brute-force attack	الهجوم الاستقصائي
Cipher Block Chaining (CBC) mode	نمط سلسلة كتل الشفرة (CBC)
Cipher FeedBack (CFB) mode	نمط التغذية المرتجعة للشفرة (CFB)
ciphertext	رسالة مشفَّرة
cryptanalysis	تحليل الشفرة
Data Encryption Standard (DES)	معيار تشفير البيانات (DES)
decryption	إزالة التشفير
Electronic CodeBook (ECB) mode	نمط دفتر الشفرة الإلكتروني (ECB)
encryption	التشفير
end-to-end encryption	التشفير من طرف إلى طرف
Feistel cipher	هيكل فيستل للتشفير
key distribution	توزيع المفاتيح
link encryption	تشفير الوصلة
plaintext	الرسالة الأصلية
session key	مفتاح الجلسة
stream cipher	تشفير التدفق
subkey	المفتاح الفرعي
symmetric encryption	التشفير المتماثل
Triple DES (3DES)	خوارزمية DES الثلاثية (3DES)

10-2 أسئلة للمراجعة ومسائل

1-10-2 أسئلة للمراجعة

2-1 ما المكونات الأساسية لنظام التشفير المتماثل؟

2-2 ما الوظيفتان الأساسيتان المستخدَمتان في خوارزميات التشفير؟

2-3 كم عدد المفاتيح اللازمة لكي يتمكن شخصان من الاتصال عبر نظام للتشفير المتماثل؟

2-4 ما الفرق بين تشفير الكتلة وتشفير التدفق؟

2-5 ما الأسلوبان العامان للهجوم على شفرة بهدف فكها؟

2-6 لماذا تستخدم بعض أنماط تشفير الكتل التشفير فقط، بينما يَستخدم البعض الآخر كلاً من التشفير وإزالة التشفير؟

2-7 عرِّف المقصود بالتشفير الثلاثي.

2-8 لماذا يشتمل الجزء الأوسط من خوارزمية 3DES على إزالة التشفير وليس إعادة التشفير؟

2-9 ما الفرق بين تشفير الوصلة والتشفير من طرف إلى طرف؟

2-10 اذكر الطرق التي يمكن استخدامها لتوزيع المفاتيح السرية على طرفين للاتصال المشفَّر باستخدامها.

2-11 ما الفرق بين مفتاح الجلسة والمفتاح الرئيس؟

2-12 عرِّف المقصود بمركز توزيع المفاتيح.

2-10-2 مسائل

2-1 وضّح أن عملية إزالة تشفير فيستل هي عكس عملية تشفير فيستل.

2-2 ما قيمة مفتاح RC4 الذي يترك قيمة S بدون تغيير خلال عملية التهيئة الأولية؟ بمعنى أنه بعد التبديلة الأولية لـ S تكون عناصر S تساوي القيم من 0 إلى 255 بترتيب تصاعدي.

2-3 لخوارزمية RC4 حالة داخلية سرية هي تبديلة من كل القيم المحتملة للمتجه S والمؤشرين i وj.

a. كم عدد البتات اللازمة لتخزين تلك الحالة الداخلية باستخدام طريقة تخزين مباشرة؟

b. لنفترض أننا سننظر إلى الأمر من وجهة نظر كمية المعلومات التي تمثلها الحالة. عندئذٍ سنحتاج إلى تحديد عدد الحالات المختلفة الموجودة، ثم نحسب لوغاريثم ذلك العدد للأساس 2 لتعيين عدد بتات المعلومات المطلوبة لتمثيل ذلك. باستخدام هذه الطريقة، كم بتاً سنحتاجها لتمثيل الحالة؟

2-4 في نمط دفتر الشفرة الإلكتروني (ECB)، إذا حدث خطأ في كتلة من كتل الرسالة المشفَّرة أثناء إرسالها، فستتضرر نتيجةً لذلك الكتلة المناظرة في الرسالة الأصلية فقط. أما في نمط سَلسَلة كتل الشفرة (CBC)، فسينتقل ذلك الخطأ. فمثلاً وجود خطأ في كتلة C_1 المرسَلة (في الشكل 2-9) سيُفسد أيضاً كلاً من الكتلتين P_2 وP_1.

a. هل تتضرر أي كتل أبعد من P_2؟

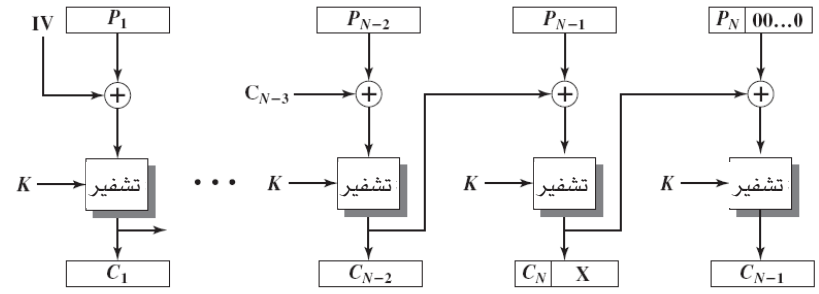
b. لنفترض وجود خطأ في أحد البتات بالنسخة المصدرية للكتلة P_1. كم عدد كتل الرسالة المشفَّرة التي سينتقل إليها ذلك الخطأ؟ ما تأثير ذلك عند المستلِم؟

2-5 حشو CBC هو نمط لتشغيل نظام تشفير الكتلة يُستخدم بالفعل في خوارزمية RC5، ولكن يمكن استخدامه أيضاً في أي نظام لتشفير الكتلة. ويمكن لحشو CBC التعامل مع رسالة أصلية بأي طول. تكون الرسالة المشفَّرة أطول من الرسالة الأصلية بكتلة واحدة كحد أقصى. يُستخدم الحشو للتأكد من أن طول الرسالة الأصلية من مضاعفات مقاس الكتلة. يُفترض أن طول الرسالة الأصلية غير المشفَّرة هو عدد صحيح من البايتات. يتم حشو تلك الرسالة الأصلية في نهايتها بعدد من البايتات يتراوح من 1 إلى bb بايت، حيث bb تساوي حجم الكتلة بالبايتات. تأخذ كل البايتات المحشوة نفس القيمة، وهي عدد البايتات التي تم حشوها. على سبيل المثال، إذا تم حشو 8 بايتات فستتضمَّن كل بايت حشو نمط البتات 00001000. لماذا لا يُسمح باستخدام حشوة صفرية؟ أي إذا كان طول الرسالة الأصلية من مضاعفات حجم الكتلة، فلماذا لا نمتنع عن الحشو كليةً في هذه الحالة؟

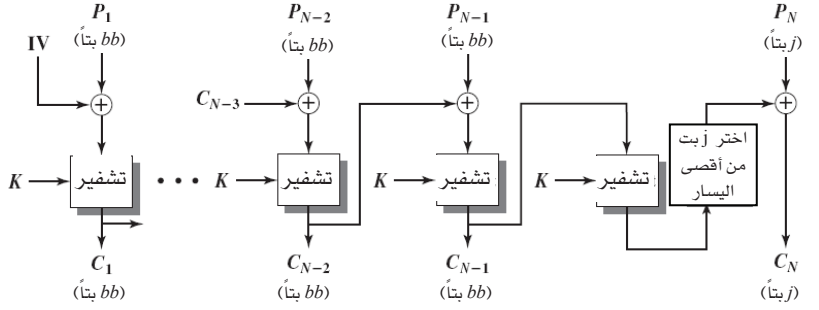
2-6 قد لا يكون أسلوب الحشو مناسباً في كل الحالات. على سبيل المثال، قد نرغب في تخزين الرسالة المشفَّرة في نفس الذاكرة المؤقتة التي كانت تحوي الرسالة الأصلية. وفي هذه الحالة، لا بد من أن تكون الرسالة المشفَّرة بنفس طول الرسالة الأصلية. هناك نمط تشغيل لتحقيق ذلك الغرض باختزال الرسالة المشفَّرة يُعرَف بـ ciphertext stealing (CTS). ويوضح الشكل 2-13 (a) طريقة لتنفيذ ذلك الأسلوب.

a. اشرح كيفية عمل تلك الطريقة.

b. صف كيف يمكن إزالة تشفير كلٍّ من الكتلتين C_n و C_{n-1}.



(a) نمط اختزال الرسالة المشفرة



(b) طريقة بديلة

الشكل 2-13 أنماط استخدام تشفير الكتلة مع رسالة أصلية بطول ليس من مضاعفات حجم الكتلة.

7-2 يبيّن الشكل 2-13(b) طريقة بديلة لأسلوب CTS لإنتاج رسالة مشفَّرة مساوية في الطول للرسالة الأصلية عندما لا يكون طول الأخيرة من مضاعفات حجم الكتلة.

a. اشرح الخوارزمية.

b. اشرح لماذا تفضّل طريقة CTS على هذه الطريقة المبينة في الشكل 2-13(b).

8-2 إذا حدث خطأ في أحد البتات أثناء نقل حرف من رسالة مشفَّرة باستخدام نمط التشغيل CFB بـ 8 بتات. إلى أي مدى سينتقل الخطأ؟

9-2 تعاني نظم توزيع المفاتيح التي تستخدم مركزاً للتحكم في الوصول و/أو مركزاً لتوزيع المفاتيح من وجود نقاط مركزية تكون عرضة للهجوم. ناقش الآثار الأمنية المترتبة على هذه المركزية في تلك النظم.

10-2 لنفترض أن شخصاً ما يقترح الطريقة التالية للتأكد من أن لدى كلٍّ واحد منكما نفس المفتاح السري. تقوم أنت بتوليد سلسلة بتات عشوائية بطول المفتاح، ثم تُجري

عملية XOR بينها وبين المفتاح، وترسل النتيجة عبر قناة الاتصال. يقوم شريكك بإجراء عملية XOR بين الكتلة التي يستلمها والمفتاح (والذي يجب أن يكون هو نفس مفتاحك) ويرسل الناتج إليك. تقوم أنت بالتحقق، فإذا وجدت أنك استلمت منه نفس سلسلة البتات العشوائية التي استخدمتها في البداية فستكون قد أثبتّ أن شريكك لديه نفس المفتاح السري، مع أنه لم يقم أيٌّ منكما بإرسال المفتاح نفسه على القناة. هل هناك عيب في هذه الخطة؟

الفصل الثالث

3 التشفير بالمفاتيح العامة وتوثيق الرسائل

محتويات الفصل:

3-1 طرق توثيق الرسائل

3-2 دوال التحوير الآمنة وخوارزمية HMAC

3-3 مبادئ التشفير بالمفاتيح العامة

3-4 خوارزميات التشفير بالمفاتيح العامة

3-5 التوقيع الرقمي

3-6 إدارة المفاتيح

3-7 توصيات للمطالعة

3-8 مصادر للمعلومات على الويب

3-9 مصطلحات رئيسة

3-10 أسئلة للمراجعة ومسائل

لكل مصري اسمان: اسم حقيقي واسم شهرة، أو اسم خاص واسم عام على التوالي. وعادة ما يُعلَن اسم الشهرة أو الاسم العام للملأ، في حين يُفصَح عن الاسم الحقيقي أو الاسم الخاص بحذر.

ـ من كتاب "الغصن الذهبي" للسيد جيمس جورج فريزر.

تقتضي الحكمة الحذر عند التعامل مع الغرباء للوقاية من التأثير الوخيم الذي قد ينجم عن ممارساتهم. ولذا قبل أن يُسمَح لهم بالدخول إلى حي، أو على الأقل قبل أن يُسمَح لهم بالاختلاط بحرية مع سكان الحي، يقوم السكان الأصليون في كثير من الأحيان بطقوس معينة لنزع القوى السحرية من الغرباء أو ـ إذا جاز التعبير ـ لتنقية الجو الملوث الذي يُفترَض أن يكونوا محاطين به.

ـ من كتاب "الغصن الذهبي" للسيد جيمس جورج فريزر.

بالإضافة إلى سرية الرسائل، يُعدُّ توثيق الرسائل (التحقق منها) من الوظائف المهمة لأمن الشبكة. ويبحث هذا الفصل ثلاثة جوانب لتوثيق الرسائل. أولاً: سننظر في استخدام أكواد ودوال التحوير لتوثيق الرسائل، بعدها سنلقي نظرة على المبادئ الأساسية للتشفير بالمفاتيح العامة وعلى خوارزميتين محددتين للتشفير بالمفاتيح العامة. وتُستخدَم تلك الخوارزميات في تبادل مفاتيح التشفير التقليدية. ثم سننظر في استخدام التشفير بالمفاتيح العامة لتوليد توقيعات رقمية للرسائل توفر طريقة محسَّنة لتوثيقها. وأخيراً سنعاود النظر في مسألة إدارة مفاتيح التشفير.

1-3 طرق توثيق الرسائل

يحمي التشفير ضد الهجوم السلبي (التنصت). لكن هناك مطلب آخر وهو الحماية من الهجوم النشط (تزوير البيانات والعمليات). وتُعرَف الحماية من مثل هذه الهجمات بتوثيق الرسائل أو التحقق منها. ويقال أن رسالةً أو ملفاً أو وثيقةً من الوثائق أو غير ذلك من تجمعات البيانات موثَّقة عندما تكون حقيقية ومرسلة من المصدر الذي يدَّعي إرسالها. ويُعرَّف توثيق الرسالة بأنه إجراء يتيح

للأطراف المتصلة توثيق الرسائل المَستلَمة، ويشمل ذلك جانبين مهمين هما التحقق من أن مضمون الرسالة لم يتغير ومن أن مصدر الرسالة موثَّق. وقد نرغب أيضاً في التحقق من وقت الرسالة (للتأكد من أنها لم تُؤخَّر ثم أعيد إرسالها بشكلٍ متعمد)، وكذلك التأكد من التسلسل الصحيح للرسائل المتدفقة بين الطرفين المتصلين.

3-1-1 التوثيق باستخدام التشفير التقليدي

يُمكن القيام بالتوثيق ببساطة باستخدام طرق التشفير التقليدية. فلو افترضنا أن المُرسِل والمُستلِم فقط اتفقا على مفتاح مشترك (كلمة سرية)، عندئذ يكون بوسع المُرسِل الحقيقي فقط تشفير الرسالة بنجاح للطرف الآخر. وعلاوة على ذلك، إذا كانت الرسالة تتضمن كوداً للكشف عن الأخطاء ورقم تسلسل، فسيمكن للمُستلِم التأكد من عدم إجراء أي تعديلات على الرسالة وأن ترتيبها في سلسلة من الرسائل صحيح. كذلك إذا كانت الرسالة تتضمن خاتماً بالوقت، فسيمكن للمُستلِم التأكد من أن الرسالة لم تتأخر عما هو متوقع في عبورها الشبكة ولم يتم إعادة إرسالها.

3-1-2 التوثيق بدون تشفير الرسالة

في هذا الجزء سنفحص عدة طرق لتوثيق الرسالة لا تعتمد على التشفير. في جميع هذه الطرق، يتم توليد وسم (tag) للتوثيق تذيّل به كل الرسائل المُرسَلة. لاحظ أن الرسالة نفسها ليست مُشفَّرة ويمكن قراءتها عند المُستلِم بشكلٍ مستقل عن وظيفة التوثيق. ولأن الطرق التي نناقشها في هذا الجزء لا تشفر الرسالة، فهي لا توفر السرية للرسائل. لمّا كانت طرق التشفير التقليدية توفر التوثيق، وتُستخدَم على نطاق واسع مع المنتجات المتوفرة، فلماذا لا نستخدم ببساطة تلك الطرق التي توفر الاثنين معاً: السرية والتوثيق؟ يقترح [DAVI89] ثلاث حالات يُفضَّل فيها توثيق الرسالة بدون توفير السرية:

1. يوجد عدد من التطبيقات تُبث فيها الرسالة نفسها إلى عدد من المُستلِمين. مثال ذلك إرسال إشعار لمستخدمي الشبكة بأنها غير متاحة الآن، أو إرسال إشارة إنذار من مركز للتحكم. في مثل هذه الحالات يكون قيام وجهة واحدة فقط بتوثيق الرسالة أقل كلفةً وأكثر اعتمادية؛ ولذا يجب أن تُبث الرسالة غير مُشفَّرة ولكنها تتضمن وسمة التوثيق. يقوم النظام المسؤول بالتحقق منها، فإذا اكتشف وجود مخالفة، فإنه يقوم بتنبيه الوجهات الأخرى للرسالة.
2. في سيناريو آخر محتمل، يتم تبادل الرسائل بين طرفين يكون أحدهما محملاً بعبء ثقيل وليس لديه الوقت الكافي لإزالة تشفير كل الرسائل التي يتلقاها. في هذه الحالة تتم عملية التوثيق على أساس انتقائي تُختار فيه الرسائل بشكلٍ عشوائي.
3. تبدو فكرة التحقق من سلامة برنامج حاسب باستخدام نص غير مُشفَّر جذابة، حيث سيتسنى تشغيل البرنامج دون الحاجة إلى إزالة التشفير في كل مرة ومن ثم التوفير في استخدام موارد المعالج. بإلحاق وسمة التوثيق بالبرنامج سيكون بالإمكان فحصها عند الحاجة للتأكد من سلامة البرنامج.

ومن ثَمَّ فهناك مكانٌ لكلٍّ من التوثيق والتشفير لتلبية المطالب الأمنية.

❖ كود توثيق الرسالة:

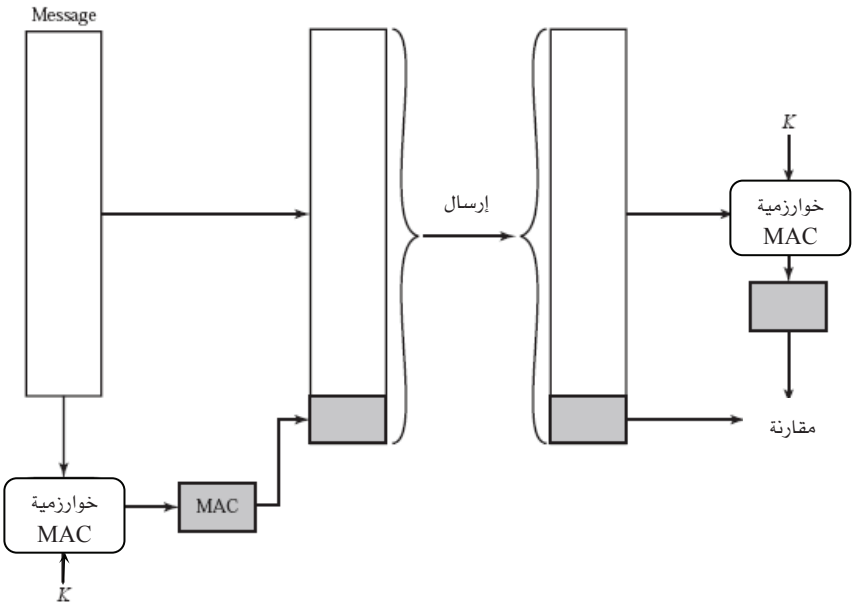
يتضمن التوثيق استخدام مفتاح سري لتوليد كتلة صغيرة من البيانات تُعرَف باسم "كود توثيق الرسالة" (Message Authentication Code (MAC)) تذيّل به الرسالة. ويفترض هذا الأسلوب أن الطرفين المتصلين A وB يستخدمان مفتاحاً سرياً مشتركاً K_{AB}. عندما يريد الطرف A إرسال رسالة إلى B، يقوم بحساب كود توثيق الرسالة من (أي كدالة في) كلٍّ من الرسالة ذاتها والمفتاح السري: $MAC_M = F(K_{AB}, M)$. تُنقل الرسالة بالإضافة إلى الكود إلى الطرف

الآخر. يقوم الطرف الآخر (أي المُستلِم) بنفس العملية الحسابية على الرسالة المُستلَمة مستخدماً نفس المفتاح السري لتوليد كود توثيق جديد. ومن ثم يقارن الكود المُستقبَل بالكود المحسوب (انظر الشكل 3-1). إذا افترضنا أن المُستلِم والمُرسِل يستخدمان نفس المفتاح السري وحدث تطابق بين كلا الكودين فإنه:

1. يتأكد المُستلِم من عدم تغير الرسالة. إذا غير أحد المهاجمين الرسالة بحيث لم يتغير الكود، فإن الكود المحسوب عند المُستلِم سيختلف عن الكود المُستلَم. لأنه من المفترض أن المهاجم لا يعرف المفتاح السري، فلن يمكنه تغيير الكود ليوافق التغيرات التي قام بها في الرسالة.
2. يتحقق المُستلِم من أن الرسالة قادمة من المصدر المزعوم. لأن المفتاح السري غير معروف لأحد آخر غير المُرسِل والمُستلِم، لا يستطيع أحد غيرهما إعداد رسالة بالكود المناسب.
3. إذا تضمّنت الرسالة رقم تسلسل (كالمستخدم في بروتوكولات X.25 وHDLC وTCP)، فيمكن للمُستلِم التأكد من التسلسل الصحيح للرسائل، لأن المهاجم لا يستطيع تعديل رقم التسلسل بنجاح.

يوجد عدد من الخوارزميات التي يمكن استخدمها لتوليد الكود. توصي معايير المعهد القومي للقياسات والتقنية NIST المعروفة بـ FIPS PUB 113 باستخدام DES لتوليد نسخة مُشفَّرة من الرسالة، ثم استخدام عدد من البتات الأخيرة ككود التوثيق. عادةً ما يكون الكود مؤلَّفاً من 16 بتاً أو 32 بتاً.

تشبه العملية التي وصفناها للتوّ عملية التشفير. أحد الفروق أنه ليس من الضروري أن تكون خوارزمية التوثيق انعكاسية (reversible) والذي يُعدُّ مطلباً أساسياً في حالة التشفير كي يتسنى إزالة التشفير؛ ومن ثم فهي أقل عرضة للتأثر (للهجوم) من التشفير بسبب الخصائص الرياضية لعملية التوثيق.



الشكل 3-1: التوثيق باستخدام كود توثيق الرسالة.

❖ دالة التحوير أحادية الاتجاه:

دالة التحوير أحادية الاتجاه هي طريقة بديلة لكود توثيق الرسالة. كما هو الحال مع كود توثيق الرسالة، تأخذ دالة التحوير رسالة M بأي حجم وتنتج خلاصة الرسالة $H(M)$ بحجم ثابت. تختلف دالة التحوير عن كود توثيق الرسالة في أنها لا تستخدم مفتاحاً سرياً. لتوثيق الرسالة، تُرسَل خلاصة الرسالة مع الرسالة بطريقة تكون فيها خلاصة الرسالة محمية (لا يمكن تعديلها).

يوضِّح الشكل 3-2 ثلاث طرق لتوثيق الرسالة. ويمكن أن تُشفَّر خلاصة الرسالة باستخدام طرق التشفير التقليدية (الشكل 3-2 (a))؛ فإذا افترضنا أن المُرسِل والمُستلِم فقط يعرفان مفتاح التشفير فسيكون التوثيق مضموناً. يمكن أيضاً تشفير خلاصة الرسالة باستخدام المفتاح العام (الشكل 3-2 (b))؛ سيتم

توضيح ذلك في الجزء 3-5. ولطريقة التشفير بالمفاتيح العامة فائدتان: فهي تنتج توقيعاً رقمياً بالإضافة إلى توثيق الرسالة؛ وكذلك لا تتطلب توزيعاً للمفاتيح بين الأطراف المتصلة.

لهاتين الطريقتين ميزة عن الطرق التي تُشفِّر الرسالة بكاملها من حيث اختصار الحسابات المطلوبة. ومع ذلك فهناك اهتمام بتطوير أسلوب يتفادى التشفير تماماً. يرجع ذلك لعدة أسباب ذكرت في [TSUD92] على النحو الآتي:

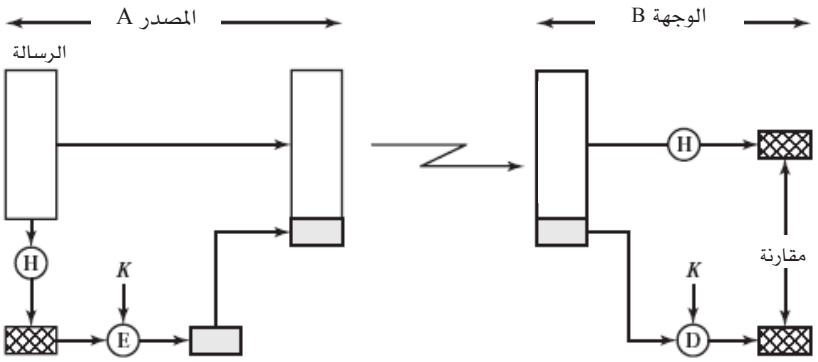
- برامج التشفير بطيئة جداً. بالرغم من صغر كمية البيانات التي تحتاج للتشفير في كل رسالة، فقد يكون هناك سيل متواصل من الرسائل الواردة للنظام والخارجة منه.
- كلفة المكوّنات المادية (العتاد) لأجهزة التشفير ليست بسيطة. وعلى الرغم من توفر شرائح إلكترونية رخيصة لتنفيذ خوارزمية DES، فإن محصّلة الكلفة ستكون عالية إذا ما تعيّن على كل العقد في الشبكة القيام بذلك.
- أجهزة التشفير مصمَّمة للتعامل مع حجوم البيانات الكبيرة. لكن مع كتل البيانات الصغيرة يضيع جزءٌ كبيرٌ من الوقت في التهيئة والاستدعاء مما يشكّل عبئاً إضافياً كبيراً.
- قد تكون خوارزميات التشفير محميّة ببراءات اختراع؛ الأمر الذي قد يُصعِّب استخدمها.

يوضِّح الشكل 3-2 (c) أسلوباً يستخدم دالة التحوير بدون تشفير لتوثيق الرسالة. يفترض هذا الأسلوب أن الطرفين المتصلين (مثلاً A وB) يستخدمان قيمة سرية مشتركة S_{AB}. عندما يريد الطرف A إرسال رسالة إلى B، فإنه يحسب دالة التحوير للرسالة متضمنة القيمة السرية:

$MD_M = \mathrm{H}(S_{AB}||M)$

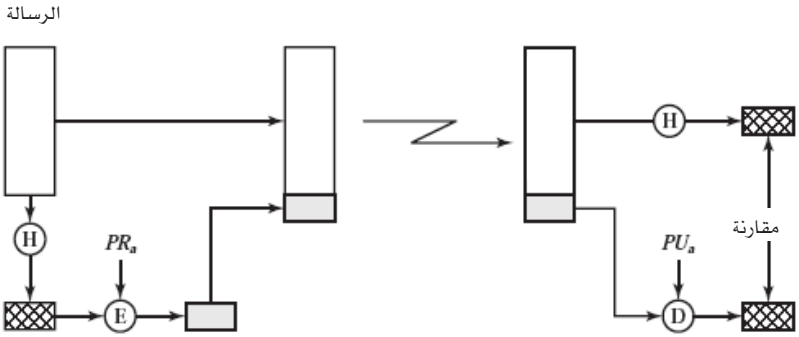
(يعني الرمز || عملية الوصل (concatenation))، ثم يرسل $[M||M_{DM}]$ إلى B. ولأن الطرف B يعرف S_{AB} فسيمكنه حساب $H(S_{AB}||M)$ للرسالة التي وصلته، ومن ثم التحقّق من MD_M. ولأنه لم يتم إرسال القيمة السرية نفسها فلن يتمكن المهاجم من تعديل الرسالة التي يعترضها. وطالما ظلت القيمة سرية، فلن يتمكن المهاجم أيضاً من إنشاء رسالة مزيفة (أي رسالة تعطي نفس ناتج دالة التحوير بنفس القيمة السرية).

هناك أسلوب آخر يُعدُّ تعديلاً للأسلوب الثالث، يطلق عليه HMAC أي Hash-based Message Authentication Code، ويُستخدم في أمن بروتوكول الإنترنت IP (كما سنصف في الفصل السادس)، كما يُنَص عليه أيضاً ضمن مواصفات بروتوكول إدارة الشبكة SNMPv3 (كما سيرد في الفصل الثامن).

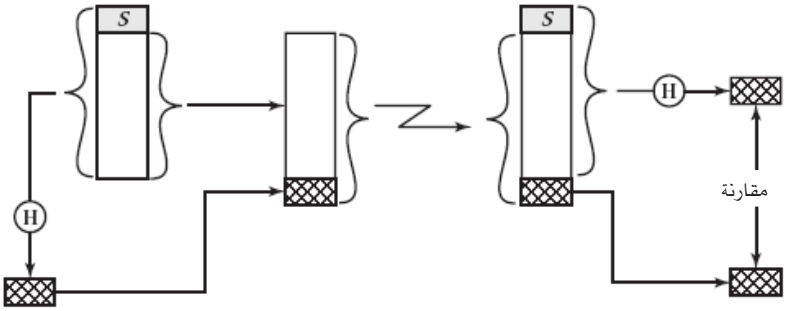


(a) استخدام التشفير التقليدي

الشكل 3-2: توثيق الرسالة بطريقة دالة التحوير أحادية الاتجاه.



(b) استخدام التشفير بالمفاتيح العامة



(c) استخدام قيمة سرية

تابع الشكل 3-2: توثيق الرسالة بطريقة دالة التحوير أحادية الاتجاه.

2-3 دوال التحوير الآمنة وخوارزمية HMAC

لدالة التحوير أحادية الاتجاه أهميتها ليس فقط لتوثيق الرسالة ولكن أيضاً للتوقيع الرقمي. سنبدأ في هذا الجزء بمناقشة متطلبات دالة التحوير الآمنة. بعدها سننظر إلى دالة من أهم دوال التحوير تعرف بـ SHA. في النهاية سنفحص خوارزمية HMAC.

1-2-3 المتطلبات من دالة التحوير

الغرض من دالة التحوير هو إنتاج "بصمة" للملف أو الرسالة أو أي كتلة من البيانات. ولكي تكون مفيدة لتوثيق الرسالة يجب أن يتوفر في دالة التحوير H الخواص الآتية:

1. يمكن تطبيقها على كتلة بيانات بأي حجم.
2. تُنتج مُخرجاً بحجم ثابت.
3. من السهل حساب $H(x)$ لأي معطي x ومن ثَمَّ يمكن تنفيذها عملياً بواسطة "برنامج" أو "مُكوِّن مادي" ("عتاد").
4. لأي قيمة معطاة h، لا يمكن حسابياً إيجاد x بحيث $H(x) = h$. يشار إلى ذلك أحياناً في أدبيات الأمن بخاصية "أحادية الاتجاه".
5. لأيّ كتلة معطاة x، لا يمكن حسابياً إيجاد كتلة بيانات y غير x بحيث $H(y) = H(x)$. ويُشار إلى هذه الخاصية أحياناً بـ "مقاومة الاصطدامات الضعيفة".
6. لا يمكن حسابياً إيجاد أي زوج (x, y) بحيث $H(y) = H(x)$. ويُشار إلى هذه الخاصية أحياناً بـ "مقاومة الاصطدامات القوية".[1]

[1] للأسف لا يوجد استخدام موحد لتلك المصطلحات فهناك مصطلحات أخرى تستخدم في أدبيات التشفير؛ منها دالة تحوير أحادية الاتجاه (الخاصيتان 4، 5)، دالة تحوير مقاومة للاصطدام (الخواص

تُعدُّ الخواص الثلاث الأولى متطلباتٍ للتطبيق العملي لدالة التحوير لتوثيق الرسالة. وبمقتضى الخاصية الرابعة، خاصية "أحادية الاتجاه"، يكون من السّهل توليد الكود لرسالة معطاة ولكن من المستحيل عملياً توليد الرسالة إذا ما عُرِف الكود. وهذه الخاصية مهمة إذا كانت طريقة التوثيق تتضمّن استخدام قيمة سرية (الشكل 3-2 (c)). ورغم أن القيمة السرية نفسها لايتم إرسالها، فإنه إذا كانت دالة التحوير لا تحقق خاصية "أحادية الاتجاه"، فسيمكن للمهاجم اكتشاف القيمة السرية بسهولة. فإذا استطاع المهاجم أن يلاحظ أو يعترض رسالة مُرسَلة، فسيمكنه الحصول على الرسالة M والكود $C=\mathrm{H}(S_{AB}||M)$، وعندها سيمكنه أن يعكس الدالة للحصول على $S_{AB}||M = \mathrm{H}^{-}(C)$[1]. ولأن المهاجم لديه الآن كلٌّ من M و$S_{AB}||M$ تكون مسألة استعادة S_{AB} أمراً بسيطاً.

وأما الخاصية الخامسة فتضمن استحالة إيجاد رسالة بديلة لها نفس قيمة دالة التحوير لتلك الرسالة. ومن شأن ذلك منع التزييف في حالة تشفير قيمة كود التحوير (الشكل 3-2 (a)،(b)). إذا لم تتحقق هذه الخاصية، فسيكون بوسع المهاجم القيام بسلسلة الإجراءات الآتية:

أولاً: ملاحظة أو اعتراض الرسالة والقيمة المُشفَّرة لكود التحوير.

ثانياً: توليد قيمة غير مُشفَّرة لكود التحوير من الرسالة.

ثالثاً: توليد رسالة بديلة لها نفس قيمة كود التحوير.

يُطلَق على دالة التحوير التي تحقق الخواص الخمس الأولى في القائمة السابقة "دالة تحوير ضعيفة". أما إذا تحققت فيها الخاصية السادسة أيضاً، فسيُطلَق عليها "دالة تحوير قوية". تحمي الخاصية السادسة ضدّ فئة متطوّرة من الهجمات تُعرَف بهجمات عيد الميلاد. تقع تفاصيل تلك الهجمات خارج مجال هذا

4، 5، 6)، دالة تحوير أحادية الاتجاه ضعيفة (الخاصيتان 4، 5)، دالة تحوير أحادية الاتجاه قوية (الخواص 4، 5، 6). لذا يجب أن ينتبه القارئ للمعنى المقصود لكلٍّ منها.

الكتاب. تؤدي مثل تلك الهجمات إلى تخفيض قوّة دالة تحوير تستخدم m بتات من 2^m إلى $2^{m/2}$. راجع [YUVA79] و[STAL06a] للمزيد من التفاصيل.

بالإضافة إلى توفير التوثيق تساعد خلاصة الرسالة أيضاً فى التأكد من سلامة البيانات، فهي تؤدي نفس الوظيفة التي تقوم بها سلسلة فحص الإطار (Frame Check Sequence (FCS)): إذا حدث تغيير لأيٍّ من البتات في الرسالة بدون قصد أثناء نقلها، فستكون خلاصة الرسالة الناتجة لدى المستلم خطأ (ومن ثم يستنتج المُستلِم أن الرسالة خطأ).

3-2-2 دوال تحوير بسيطة

تعمل جميع دوال التحوير طبقاً للمبادئ العامة الآتية. يُنظر إلى المُدخلات (رسالة أو ملف أو غير ذلك) كسلسلة من كتل البيانات يتألَّف كلٌّ منها من n بت. يتم معالجة المُدخل كتلةً كتلةً على التوالي لإنتاج قيمة لدالة التحوير تتكون من n بت.

	البت 1	البت 2	• • •	البت n
الكتلة 1	b_{11}	b_{21}		b_{n1}
الكتلة 2	b_{12}	b_{22}		b_{n2}
	•	•	•	•
	•	•	•	•
	•	•	•	•
الكتلة m	b_{1m}	b_{2m}		b_{nm}
رمز التحوير	C_1	C_2		C_n

الشكل 3-3: دالة تحوير بسيطة تستخدم دالة المنطق XOR.

تمثل دالة XOR المنطقية على البتات في كتل البيانات المختلفة إحدى دوال التحوير البسيطة، ويمكن التعبير عنها على النحو الآتي:

$$C_i = b_{i1} \oplus b_{i2} \oplus \ldots \oplus b_{im}$$

حيث:

C_i = البت i للتحوير الناتج، $1 \le i \le n$.

m = عدد كتل البتات، كل كتلة مكوّنة من n بت.

b_{ij} = البت i في كتلة البتات j.

$\oplus$ = دالة XOR.

يوضِّح الشكل 3-3 هذه العملية التي تُنتج بت تكافؤ (parity bit) لكل بت في كتل البيانات وتُعرَف بفحص الفائض الطولي (longitudinal redundancy check). وهي طريقة فعّالة إلى حدّ معقول للتأكد من سلامة البيانات العشوائية. وأي قيمة لدالة التحوير (المكوّنة من n بت) لها نفس الاحتمال. ومن ثَمَّ يكون احتمال أن يؤدي حدوث خطأ في البيانات إلي عدم تغير قيمة دالة التحوير يساوي 2^{-n}. في حالة البيانات بصيغة أكثر نظاماً، يُتوقع أن تكون تلك الدالة أقل فعالية. فعلى سبيل المثال، في معظم الملفات النصية المعتادة تكون البت الأعلى رتبة في كل بايت صفراً دائماً. ولذلك إذا استخدمنا كود تحوير من 128 بتاً ستكون فعالية دالة التحوير لمثل تلك البيانات 2^{-112} بدلاً من 2^{-128}.

من الطرق البسيطة لتحسين هذا الوضع إجراء إزاحة دائرية ببت واحد لقيمة دالة التحوير بعد معالجة كل كتلة بيانات (أي تدوير قيمة الدالة بتاً واحداً). يمكن تلخيص هذا الإجراء على النحو الآتي:

1. في البداية اجعل قيمة دالة التحوير صفراً.
2. قم بمعالجة كل كتلة بيانات على التوالي كالآتي:
   a. قم بتدوير قيمة دالة التحوير الحالية إلى اليسار بتاً واحداً.
   b. قم بحساب XOR للكتلة مع قيمة دالة التحوير.

من شأن هذه الطريقة جعل البيانات المُدخلة أكثر عشوائية، ومن ثَمَّ التغلب على أي انتظام قد يوجد في تلك البيانات.

بالرغم من أن الإجراء الثاني يوفر قدراً جيداً من سلامة البيانات، فإنه عديم الفائدة تقريباً لأمن البيانات عند استخدام كود تحوير مُشفَّر مع رسالة غير مُشفَّرة، كما في الشكل 2-3 (a)، (b). لرسالة معطاة، من السهل إنتاج رسالة جديدة لها نفس كود التحوير: ببساطة جهّز الرسالة البديلة ثم ألحق بها كتلة طولها n بت تجعل كود التحوير الناتج للرسالة + تلك الكتلة هو كود التحوير المطلوب.

رغم أن عملية XOR البسيطة أو عملية XOR مع التدوير (RXOR) غير كافية إذا كان كود التحوير فقط هو المُشفَّر، فقد تشعر بأن مثل تلك الدالة البسيطة قد تكون مفيدة عندما يكون كلٌّ من الرسالة وكود التحوير مُشفَّرين. ولكن يجب عليك توخي الحذر. في طريقة أخرى اقتُرحت في الأصل من قِبَل المكتب الوطني للمعايير استخدمت دالة XOR البسيطة مع كتل الرسالة بطول 64 بتاً ثم تشفير الرسالة بكاملها في نمط سَلسَلة كتل الشفرة (Cipher Block Chaining (CBC)). ويمكن أن نُعرِّف هذه الطريقة كالآتي: لرسالة معطاة مكوّنة من سلسلة من الكتل $X_1, X_2, \ldots, X_N$ طول كلٍّ منها 64 بتاً نحصل على كود التحوير C كناتج عملية XOR على جميع الكتل، ثم نلحق الكود الناتج بالرسالة ككتلة أخيرة:

$$C = X_{N+1} = X_1 \oplus X_2 \oplus \ldots \oplus X_N$$

بعد ذلك نشفر الرسالة بكاملها مع كود التحوير في نمط CBC للحصول على الرسالة المُشفَّرة:

$$Y_1, Y_2, \ldots, Y_{N+1}$$

ذكر [JUEN 85] عدّة طرق يمكن استخدامها للعبث بالرسالة المُشفَّرة دون التأثير على قيمة كود التحوير. على سبيل المثال، من تعريف CBC، كما بالشكل 2-9، نحصل على:

$$X_1 = IV \oplus \mathrm{D}(K, Y_1)$$
$$X_i = Y_{i-1} \oplus \mathrm{D}(K, Y_i)$$
$$X_{N+1} = Y_N \oplus \mathrm{D}(K, Y_{N+1})$$

تُحسب قيمة كود التحوير X_{N+1} عند المُستلِم من:

$$X_{N+1} = X_1 \oplus X_2 \oplus \ldots \oplus X_N$$
$$= [IV \oplus \mathrm{D}(K, Y_1)] \oplus [Y_1 \oplus \mathrm{D}(K, Y_2)] \oplus \ldots \oplus [Y_{N-1} \oplus \mathrm{D}(K, Y_N)]$$

ونظراً لأنه يمكن إجراء XOR للحدود في المعادلة السابقة بأي ترتيب، فلن تتغير قيمة كود التحوير التي يحسبها المُستلِم إذا تم تبديل مواضع كتل الرسالة المُشفَّرة.

3-2-3 دالة التحوير الآمنة SHA-1

تم تطوير خوارزمية التحوير الآمنة (Secure Hash Algorithm (SHA)) بالمعهد الوطني للمعايير والتقنية (NIST) ونُشرت كأحد المعايير الاتحادية (الفيدرالية) لمعالجة البيانات بأمريكا (FIPS 180) في عام 1993. وتم إصدار نسخة معدلة FIPS 180-1 في عام 1995 عُرِفت بشكلٍ عام باسم خوارزمية التحوير الآمنة SHA-1. وتم أيضاً توصيف خوارزمية SHA-1 في RFC 3174، كنسخة طبق الأصل تقريباً من FIPS 180-1، لكن مع إضافة برنامج لها بلغة C.

تحسب SHA-1 قيمة دالة التحوير بطول 160 بتاً. في عام 2002، أصدر معهد NIST نسخة جديدة FIPS 180-2 من المعيار عرّفت ثلاث نوعيات جديدة من خوارزمية SHA هي SHA-256 وSHA-384 وSHA-512 بأطوال 256 و384 و512 بتاً لقيمة دالة التحوير على التوالي (انظر الجدول 3-1). ولهذه النسخ الجديدة نفس البنية التحتية وتستخدم نفس عمليات الحساب المقاسي والعمليات المنطقيّة الثنائية التي تستخدمها خوارزمية SHA-1. في عام 2005، أعلن معهد NIST عن نيته لإلغاء SHA-1 والاعتماد على نسخ SHA الأخرى بحلول العام 2010. بعد ذلك بقليل وصف فريق بحثي هجوماً أمكن فيه إيجاد رسالتين منفصلتين لهما نفس كود التحوير SHA-1 باستخدام 2^{69} عملية وهو أقل بكثير من الـ 2^{80} عملية التي اعتُقد سابقاً أنها مطلوبة لإيجاد اصطدام (collision) بكود تحوير SHA-1 [Wang05]. ويُتوقع أن تعجِّل هذه النتيجة من عملية الانتقال إلى النسخ الأخرى الأقوى من خوارزمية SHA.

الجدول 3-1 مقارنة متغيرات SHA.

	SHA-1	SHA-256	SHA-384	SHA-512
حجم خلاصة الرسالة	160	256	384	512
حجم الرسالة	أقل من 2^{64}	أقل من 2^{64}	أقل من 2^{128}	أقل من 2^{128}
حجم الكتلة	512	512	1024	1024
طول الكلمة	32	32	64	64
عدد الخطوات	80	64	80	80
الأمن	80	128	192	256

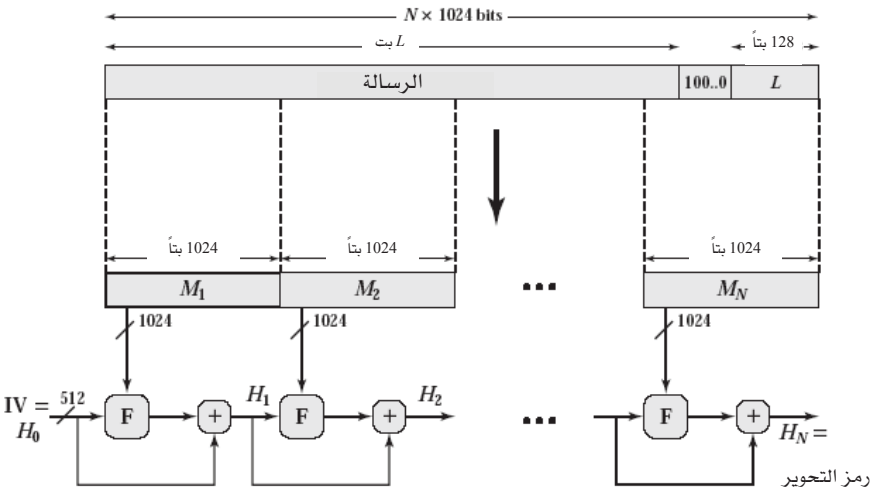
ملاحظات:

1. كل الأطوال مقاسة بالبت.
2. يشير "الأمن" بالصف الأخير إلى حقيقة أن هجوم عيد الميلاد على خلاصة رسالة (message digest) بطول n سوف يُنتج اصطدام بمعامل شغل يساوي تقريباً $2^{n/2}$.

سنقدم في هذا الجزء وصفاً لخوارزمية SHA-512 (النسخ الأخرى مماثلة جداً لها). تعمل تلك الخوارزمية على رسالة مُدخلة بطول أقصى أقل من 2^{128} بتاً وتنتج خلاصة للرسالة بطول 512 بتاً. يتم معالجة الرسالة المُدخلة في كتل بطول 1024 بتاً. ويبيّن الشكل 3-4 خطوات المعالجة التي تتم بشكلٍ عام على رسالة لإنتاج خلاصة لها. يشمل ذلك الخطوات الآتية:

الخطوة 1: إضافة بتات الحشوة (padding bits) إلى الرسالة.

تضاف بتات للرسالة ليصبح طولها بالبت بعد الإضافة يزيد 896 على مضاعفات العدد 1024 (وهو ما يعرف رياضياً بالتطابق congruence مع العدد 896 عند القسمة على 1024، أي أن: [length ≡ 896 modulo 1024]). دائماً تضاف بتات حتى لو كان طول الرسالة هو الطول المطلوب. ومن ثَمَّ يتراوح عدد بتات الحشو من 1 إلى 1024، وتبدأ ببت قيمته 1 يتبعه بتات كلها أصفار.



الشكل 3-4: توليد خلاصة الرسالة باستخدام خوارزمية SHA-512.

الخطوة 2: إضافة طول الرسالة.

تضاف كتلة من 128 بتاً إلى الرسالة. تُعامل هذه الكتلة كعدد صحيح بدون إشارة طوله 128 بتاً، حيث تأتي البايت الأكبر قيمة أولاً، وتمثل طول الرسالة الأصلية (أي بدون بتات الحشوة) بالبتات.

تنتج الخطوتان 1 و2 رسالة بطول من مضاعفات 1024 بتاً. يبيّن الشكل 3-4 رسالة تم تمديدها بهذا الشكل ومن ثم تُمثّل كسلسلة من الكتل، طول كل كتلة منها 1024 بتاً: $M_1, M_2, \ldots, M_N$. وبذلك يكون الطول الكلي للرسالة الممددة $N \times 1024$.

الخطوة 3: تهيئة المخزن المؤقت لكود التحوير.

يُستخدَم مخزن مؤقت سعته 512 بتاً لحفظ النتائج المرحلية والنتائج النهائية لدالة التحوير. يمكن تمثيل المخزن المؤقت بثمانية مُسجِّلات (registers) (a, b, c, d, e, f, g, h) طول كلٍّ منها 64 بتاً. في البداية توضع القيم الابتدائية التالية في تلك المُسجِّلات الثمانية:

a = 6A09E667F3BCC908
b = BB67AE8584CAA73B
c = 3C6EF372FE94F82B
d = A54FF53A5F1D36F1
e = 510E527FADE682D1
f = 9B05688C2B3E6C1F
g = 1F83D9ABFB41BD6B
h = 5BE0CD19137E2179

يتم تخزين تلك القيم في صيغة big-endian، أي يوضع البايت الأعلى رتبة في الكلمة في موضع البايت ذات العنوان الأقل (أقصى اليسار). تم الحصول على هذه الكلمات بأخذ الأربعة والستّين بتاً الأولى من الأجزاء العشرية للجذور التربيعية للأعداد الأولية (prime numbers) الثمانية الأولى.

الخطوة 4: معالجة الرسالة كتلةً كتلةً بطول 1024 بتاً (أي 128 كلمة).

يشكِّل قلب الخوارزمية وحدة تتضمن 80 جولة (مرحلة)؛ يشار إليها بالرمز F في الشكل 3-4. يبيّن الشكل 3-5 المنطق المُستخدم. تأخذ كل جولة قيمة المخزن المؤقت بطول 512 بتاً من المُسجِّلات الثمانية abcdefgh وتقوم بتعديل تلك القيمة. عند الدخول إلى الجولة الأولى يحتوي المخزن المؤقت على القيمة المرحلية لدالة التحوير H_{i-1}. تستخدم كل جولة t قيمة W_t بطول 64 بتاً مشتقة من كتلة الرسالة المكوَّنة من 1024 بتاً والجاري معالجتها حالياً (M_i). تستخدم كل جولة أيضاً ثابتاً جمعياً (additive constant) K_t، حيث تشير t ($0 \le t \le 79$) إلى جولة من الجولات الثمانين. تمثّل هذه الكلمات الـ 64 بتاً الأولى من الأجزاء العشرية

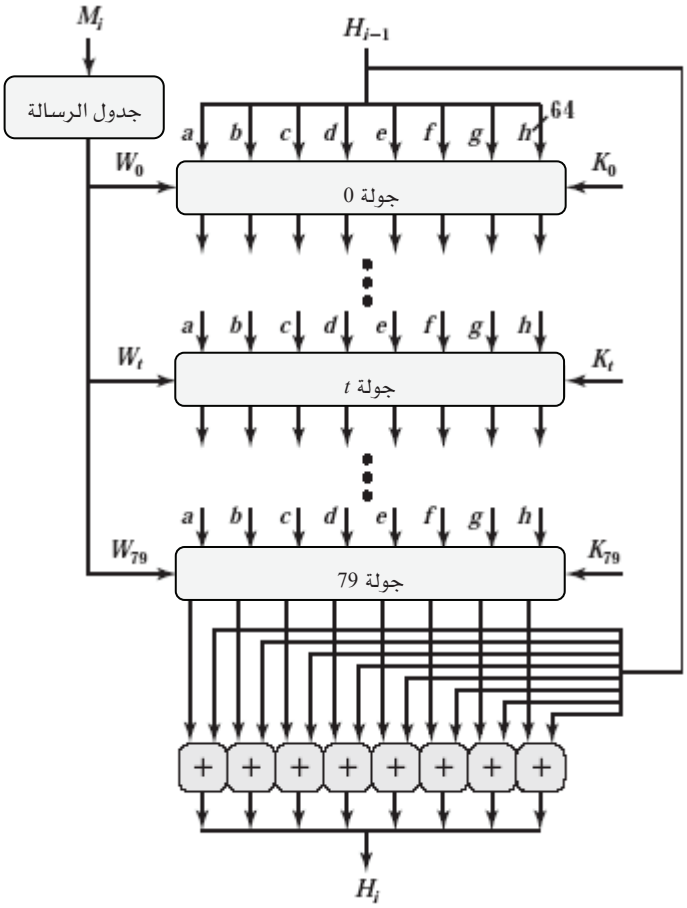
للجذور التكعيبية للأعداد الأولية الثمانين الأولى. توفر تلك الثوابت مجموعة "عشوائية" من أنماط من 64 بتاً، والتي يُتوقع أن تُزيل أي انتظام (نمطية) قد يوجد في البيانات الداخلة.

يضاف ناتج الجولة الثمانين إلى مُدخل الجولة الأولى H_{i-1} لإنتاج H_i. تتم عملية الإضافة بشكلٍ منفصل لكلٍّ من الكلمات الثمانية في المخزن المؤقت (buffer) مع الكلمات المناظرة في H_{i-1} باستخدام عملية الجمع في مجال باقي القسمة على 2^{64}.

الخطوة 5: إخراج الناتج.

بعد معالجة جميع الكتل N والمكوّنة من 1024 بتاً لكلٍّ، يمثل الناتج من المرحلة N خلاصة الرسالة بطول 512 بتاً.

تمتاز الخوارزمية SHA-512 بالخاصية أن كل بت من كود التحوير دالة في كل بت في المُدخل. يولّد التكرار المعقّد للدالة الأساسية F نتائج مخلوطة بشكلٍ جيد؛ أي أنه من غير المحتمل أن يكون لرسالتين مختارتين بشكلٍ عشوائي (حتى إذا كان بهما نفس الانتظام) نفس كود التحوير. ما لم يكن هناك بعض الضعف المخفي – الذي لم يُنشر حتى الآن – في SHA-512، تُقدَّر صعوبة المجيء برسالتين لهما نفس خلاصة الرسالة بحوالي 2^{256} عملية، في حين تقدّر صعوبة الحصول على رسالة من كود تحوير مُعطى بحوالي 2^{512} عملية.



الشكل 3-5: المعالجة التي تتم في خوارزمية SHA-512 لكتلة من 1024 بتاً.

4-2-3 دوال التحوير الآمنة الأخرى

كما هو الحال في التشفير المتماثل لكتل البيانات، يمانع مصمِّمو دوال التحوير الآمنة في استبدال هيكلية مُجرَّبة أثبتت جدواها. فمثلاً تعتمد خوارزمية DES للتشفير على هيكلية فيستيل (Feistel). وتتبع كل الأنظمة المهمة لتشفير الكتل – التي ظهرت بعد ذلك – تقريباً تصميم فيستيل أو تعميماً لهذا التصميم يتضمن جولات متعدّدة لدوال التعويض والتبديل. يمكن تكييف هذا التصميم لمقاومة تهديدات التشفير التي تُكتشف حديثاً. أما إذا استخدم تصميم جديد تماماً للتشفير المتماثل للكتلة، فسيكون هناك مخاوف من أن تفتح الهيكلية الجديدة على نفسها دروباً جديدة من الهجوم لم تخطر بعد على بال. بالمثل، تتبع معظم دوال التحوير الحديثة المهمة الهيكلية الأساسية المبينة في الشكل 3-4، وتعرف باسم دالة التحوير التكرارية (iterative hash function)، وقد اقترحها في البداية ميركل [MERK89, MERK79]. يرجع الحافز لهذا الهيكل التكراري إلى ملاحظة ميركل [MERK89] ودامجارد [DAMG89] أنه إذا كانت الدالة لكتلة واحدة (تُعرَف بدالة الضغط (compression function)) مقاومة للتصادم، فإن دالة التحوير التكرارية الناتجة تكون كذلك أيضاً. ومن ثَمَّ يمكن استخدام ذلك الهيكل لإنتاج دالة تحوير آمنة للعمل على رسائل بأي طول. وهكذا تُختزل مشكلة تصميم دالة تحوير آمنة إلى تصميم دالة ضغط مقاومة للتصادم تعمل على مُدخلات بطول معين ثابت. لقد أثبتت هذه الطريقة جدارتها بشكلٍ أساسي، وتقوم التصاميم الحديثة فقط بتعديلات طفيفة على الهيكل وزيادة طول كود التحوير.

نستعرض في هذا الجزء دالتين أخريين لدوال التحوير الآمنة التي لاقت قبولاً جيداً على المستوى التجاري بالإضافة إلى خوارزميات SHA.

❖ خوارزمية خلاصة الرسالة (MD5):

تم تطوير خوارزمية خلاصة الرسالة MD5 (Message Digest 5) من قِبَل رون ريفيست (Ron Rivest) (RFC 1321). كانت خوارزمية MD5 أكثر دوال التحوير الآمنة انتشاراً إلى سنوات قليلة مضت، عندما ظهرت مخاوف تتعلق بأنواع الهجوم الاستقصائي (brute-force attacks) وعمليات تحليل الشفرة (cryptanalysis). تأخذ تلك الخوارزمية رسالةً بأي طول كمُدخل لتُنتج خلاصةً للرسالة بطول 128 بتاً. تتم معالجة المُدخل في كتل بطول 512 بتاً.

مع زيادة سرعة المعالجات، أصبحت القدرة الأمنية لدوال التحوير بطول 128 بتاً مشكوكاً فيها. يمكن إثبات أن صعوبة إيجاد رسالتين لهما نفس الخلاصة تقدَّر بحوالي 2^{64} عملية، بينما تقدَّر صعوبة إيجاد رسالة من خلاصة مُعطاه بحوالي 2^{128} عملية. يُعدُّ العدد الأول من العمليات صغيراً جداً من الناحية الأمنية، كما تم تطوير عدد من الهجمات أظهرت ضعف MD5 لمقاومة عمليات تحليل الشفرة [BERS92, BOER93, DOBB96].

❖ خوارزمية الدوّامة (Whirlpool):

تم تطوير هذه الخوارزمية من قِبَل فنسينت ريجمين البلجيكي الذي شارك في تطوير خوارزمية ريجندايل والتي استُخدمت كمعيار التشفير المتقدّم (Advanced Encryption Standard (AES))، وعالم التشفير البرازيلي بولو باريتو (Paulo Pareto). تُعدُّ خوارزمية الدوّامة (Whirlpool) إحدى دالتي التحوير الوحيدتين المعتمدتين من قبل الهيئة الأوروبية للأساليب الجديدة للتوقيعات والسلامة والتشفير (New European Schemes for Signatures, Integrity, and Encryption (NESSIE)) [PREN02] (تشمل الأساليب المعتمدة الأخرى ثلاث نوعيات من SHA هي SHA-256 وSHA-384 وSHA-512). NESSIE هو مشروع مموَّل من قبل الاتحاد الأوروبي لتقديم حقيبة من أنواع مختلفة من أساليب

التشفير الأساسية القوية، بما في ذلك تشفير الكتلة، والتشفير المتماثل، ودوال التحوير، وأكواد توثيق الرسالة.

تعتمد خوارزمية الدوّامة على استخدام تشفير كتلة كدالة ضغط. صُمِّم ذلك النظام لتشفير الكتلة خصيصاً للاستخدام في دالة التحوير ومن غير المحتمل على الإطلاق استخدامه كدالة تشفير قائمة بذاتها. يرجع السبب في ذلك إلى أن المصممين أرادوا استخدام نظام تشفير كتلة بنفس القدرة الأمنية والكفاءة لـ AES لكن مع تحوير بطول يوفر إمكانيات أمن مكافئة لـ SHA-512. كانت النتيجة خوارزمية تشفير كتلة W تستخدم نفس الهيكل ونفس الدوال الأولية لخوارزمية AES، ولكن بكتل ومفاتيح طولها 512 بتاً.

تأخذ خوارزمية الدوّامة رسالة بطول لا يزيد عن 2^{256} بتاً وتُنتج خلاصة للرسالة بطول 512 بتاً. تتم معالجة رسالة المُدخل كتلةً كتلةً بطول 512 بتاً.

5-2-3 خوارزمية HMAC

تزايد في السنوات الأخيرة الاهتمام بتطوير أكواد لتوثيق الرسائل (MAC) مشتقة من دوال تحوير تشفيرية، كـ SHA-1. ومما شجع على هذا الإهتمام ما يأتي:

- تُنفَّذ وظائف دوال التحوير التشفيرية عموماً بشكلٍ أسرع من خوارزميات التشفير التقليدية كـ DES.
- تتوافر مكتبات برمجية لدوال التحوير التشفيرية على نطاق واسع.

لم تُصمَّم دالة التحوير (مثلاً SHA-1) لاستخدامها ككود لتوثيق الرسالة (MAC)، ولا يمكن استخدامها لهذا الغرض بشكلٍ مباشر؛ لأنها لا تعتمد على مفتاح سرّي. ظهرت عدة مقترحات لتضمين مفتاح سرّي مع إحدى خوارزميات التحوير المتوفرة. في هذا الصدد تُعدُّ خوارزمية HMAC [BELL96a, BELL96b] أكثر الطرق التي لاقت دعماً. تم إصدار HMAC في طلب التعليقات RFC

2104، وتم اختيارها أسلوباً يجب تنفيذه لأمن بروتوكول الإنترنت IP، كما استُخدِمت في بروتوكولات أخرى للإنترنت كبروتوكول أمن طبقة النقل (Transport Layer Security (TLS)) (والذي سيحل قريباً محل بروتوكول طبقة المقابس الآمنة SSL (Secure Sockets Layer)) وبروتوكول المُعامَلات الإلكترونية الآمنة (Secure Electronic Transaction (SET)).

❖ الأهداف التصميمية لخوارزمية HMAC

يسرد طلب التعليقات RFC 2104 الأهداف التصميمية لخوارزمية HMAC كالآتي:

- استخدام دوال التحوير المتوفرة بدون تعديلات، وخاصةً دوال التحوير ذات الأداء الجيد المتوفرة على شكل برمجيات يتوفر الكود الخاص بها مجاناً وعلى نطاق واسع.
- السماح باستبدال دالة التحوير المضمّنة بسهولة في حالة توفر دوال تحوير أسرع أو أكثر أمناً.
- الحفاظ على مستوى الأداء الأصلي لدالة التحوير دون أي انخفاض ملحوظ.
- استخدام المفاتيح ومناولتها بطريقة بسيطة.
- وجود تحليل تشفيري مفهوم بشكلٍ جيد لأوجه القوة لآلية التوثيق يستند على فرضيات معقولة فيما يتعلق بدالة التحوير المضمّنة.

يُعدُّ الهدف الأول والثاني مهمين في تحديد مدى القبول لـ HMAC. تتعامل خوارزمية HMAC مع دالة التحوير كـ "صندوق أسود"؛ ولهذه الطريقة فائدتان: أولاً، يمكن استخدام تنفيذ موجود بالفعل لدالة التحوير بمثابة وحدة ضمن بنية HMAC. بهذه الطريقة، يكون معظم كود HMAC مُعداً مسبقاً وجاهزاً للاستخدام بدون تعديل. ثانياً، إذا حدث وظهرت الحاجة لاستبدال دالة التحوير

في تطبيق HMAC بأخرى، فسيكون المطلوب هو فقط إزالة الوحدة الخاصة بالدالة الحالية وإحلالها بوحدة أخرى لدالة التحوير الجديدة. يمكن اللجوء لذلك عند الرغبة في دالة تحوير أسرع. من الأسباب الأكثر أهمية تعرّض أمن دالة التحوير المضمّنة للخطر، حيث يمكن استعادة أمن HMAC ببساطة باستبدال دالة التحوير المضمّنة بأخرى أكثر أمناً.

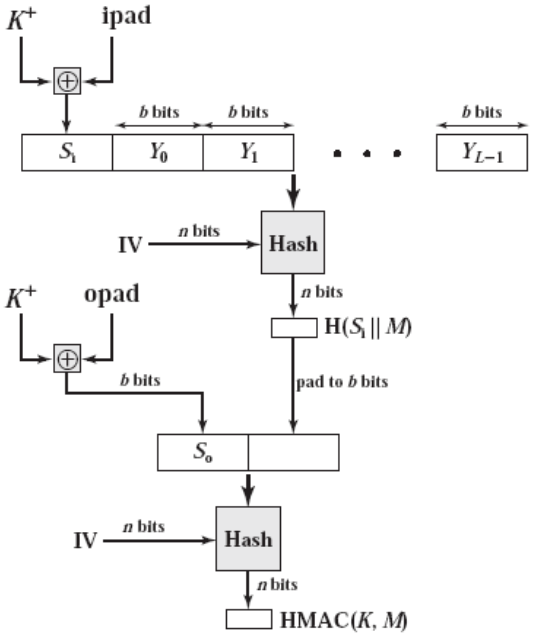
في الحقيقة، يُعدُّ هدف التصميم الأخير في القائمة السابقة الميزة الرئيسة لـ HMAC على كل الأساليب الأخرى التي اقتُرحت. يمكن إثبات أن خوارزمية HMAC آمنة بشرط توفر بعض جوانب القوة التشفيرية لدالة التحوير المضمّنة بدرجة معقولة. سنعود لهذه النقطة لاحقاً في هذا الجزء، لكننا سنتناول أولاً هيكل خوارزمية HMAC.

❖ الخوارزمية

يوضّح الشكل 3-6 طريقة العمل العامة لـ HMAC. لنُعرِّف الحدود الآتية:

- H: دالة التحوير المضمّنة (كـ SHA-1).
- M: رسالة المُدخل لخوارزمية HMAC (بما في ذلك بتات الحشو المحددة في دالة التحوير).
- Y_i: الكتلة i في الرسالة M، $0 \le I \le L - 1$.
- L: عدد الكتل في الرسالة M.
- b: عدد البتات في الكتلة.
- n: طول كود التحوير الناتج من دالة التحوير المضمّنة.
- K: المفتاح السرّي؛ إذا كان طول المفتاح أكبر من b، يتم إدخال المفتاح إلى دالة التحوير لإنتاج مفتاح بطول n بت؛ الطول الموصى به $\ge n$.

- K^+: عبارة عن K بعد إضافة بتات حشوة على اليسار ليكون طول الناتج b بت.
- ipad حشوة تتكون من 00110110 (36 بالنظام الست عشري) مكررة $b/8$ مرة.
- opad حشوة تتكون من 01011100 (5C بالنظام الست عشري) مكررة $b/8$ مرة.



الشكل 3-6: تركيبة HMAC.

يمكن التعبير عن الـ HMAC على النحو الآتي:

$$HMAC(K, M) = H[(K^{+} \oplus opda) \,||\, H[(K^{+} \oplus ipad) \,||\, M]]$$

يمكن التعبير عن ذلك بشكلٍ آخر كما يأتي:

1. أضف أصفاراً على يسار المفتاح K لتكوين سلسلة من b بت (مثلاً إذا كان طول K = 160 بتاً وb = 512 بتاً، فعندئذ يُزاد طول K بعدد من الأصفار يعادل 44 بايتاً كلٌّ منها قيمته صفر 0x00.
2. قم بإجراء عملية XOR بين K^{+} وipad (لكل بت على حدة) لإنتاج كتلة S_i تضم b بت.
3. ألحق M بـ S_i.
4. احسب دالة التحوير H للسلسلة الناتجة من الخطوة 3.
5. قم بإجراء عملية XOR بين K^{+} وopad (لكل بت على حدة) لإنتاج كتلة S_o من b بت.
6. ألحق ناتج التحوير في الخطوة 4 بـ S_o.
7. احسب دالة التحوير H للسلسلة الناتجة من الخطوة 6 وأخرِج النتيجة.

لاحظ أن عملية XOR مع ipad تؤدي إلى قلب قيمة كل بت لنصف بتات K. بالمثل تؤدي عملية XOR مع opad إلى قلب قيمة كل بت لنصف بتات K ولكنها تعمل على مجموعة مختلفة من البتات. ينتج عن تمرير S_i وS_o خلال دالة التحوير الحصول على مفتاحين من K بطريقة شبة عشوائية (pseudorandom). يُتوقع أن يستغرق تنفيذ HMAC على الرسائل الطويلة نفس الوقت تقريباً الذي تستغرقه دالة التحوير المضمّنة.

3-3 مبادئ التشفير بالمفاتيح العامة

لا يقل التشفير بالمفاتيح العامة أهميةً عن التشفير التقليدي، حيث يُستخدم في توثيق الرسائل وتوزيع المفاتيح. نستعرض في هذا الجزء أولاً المفهوم الأساسي للتشفير بالمفاتيح العامة، ثم نلقي نظرة تمهيدية على قضايا توزيع المفاتيح. أما الجزء 3-4 فيتناول أهم خوارزميتين للتشفير بالمفاتيح العامة: خوارزمية RSA، وخوارزمية ديفي - هيلمان (Diffie-Hellman). وسيعطي الجزء 3-5 مقدمة عن التوقيعات الرقمية.

3-3-1 تركيبة التشفير بالمفاتيح العامة

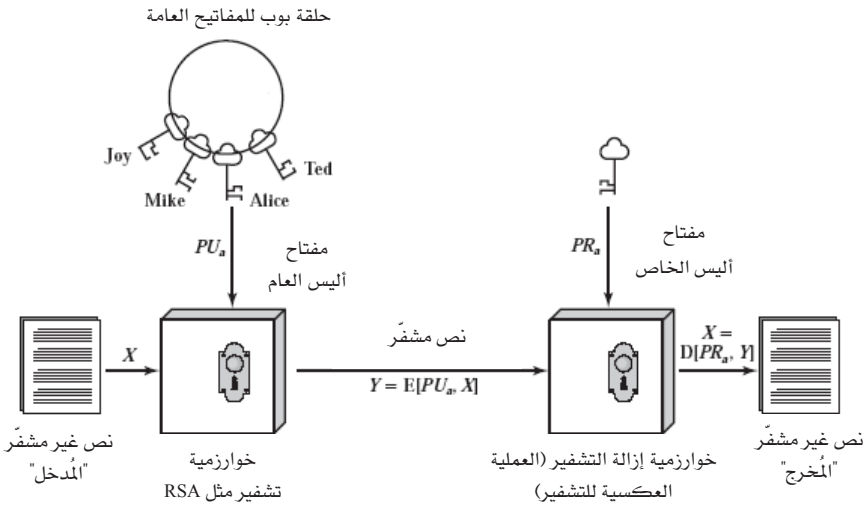
يُعدُّ التشفير بالمفاتيح العامة – الذي اقتُرح أولاً للجمهور من قِبَل ديفي وهيلمان في عام 1976 [DIFF76] – أول تقدم ثوري حقيقي في التشفير على مدى آلاف السنين. تعتمد خوارزميات المفاتيح العامة على دوال رياضية بدلاً من العمليات البسيطة على أنماط البتات كالتي تستخدم في خوارزميات التشفير المتماثل. كما أن التشفير بالمفاتيح العامة لاتماثلي (asymmetric) (أي يتضمّن استخدام مفتاحين منفصلين) بالمقارنة مع التشفير التقليدي المتماثل (الذي يستخدم مفتاحاً واحداً فقط). لاستخدام مفتاحين نتائج عميقة الأثر فيما يتعلق بتطبيقات السريّة، وتوزيع المفاتيح، والتحقق من الهوية.

قبل متابعة الشرح، يجب أن نذكر أولاً عدّة مغالطات شائعة حول التشفير بالمفاتيح العامة. الأولى هي أن التشفير بالمفاتيح العامة أكثر أمناً ضد تحليل الشفرة من التشفير التقليدي. وفي الحقيقة، يعتمد أمن أي أسلوب للتشفير على (1) طول المفتاح و(2) كم الحسابات المطلوب لكسر الشفرة. من حيث المبدأ لا يوجد شيء يتعلق بالتشفير التقليدي أو التشفير بالمفاتيح العامة يجعل أحدهما أفضل من الآخر من حيث مقاومة تحليل الشفرة. ومغالطة ثانية هي أن التشفير بالمفاتيح العامة طريقة متعددة الأغراض أدت إلى جعل التشفير التقليدي طريقة

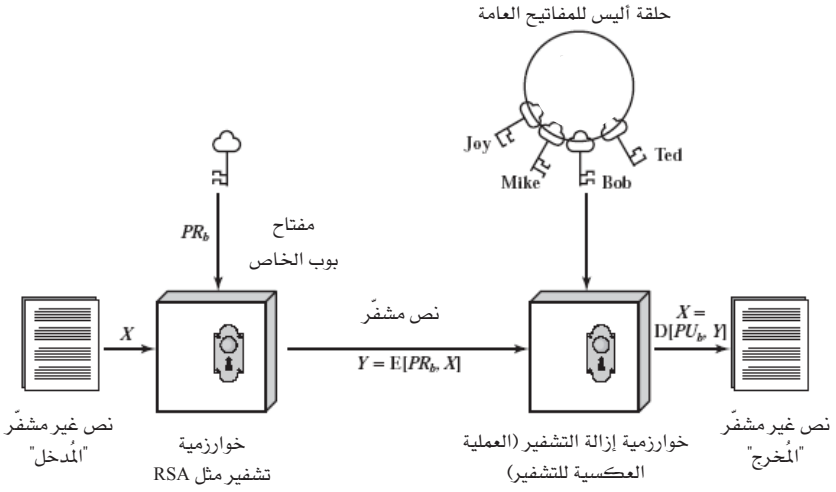
متقادمة. على عكس ذلك، بسبب عبء الحسابات الإضافية في الأساليب الحالية للتشفير بالمفاتيح العامة، لا يبدو في الأفق أي احتمال لإلغاء طرق التشفير التقليدية في المستقبل القريب. وأخيراً، هناك شعور بأن توزيع المفاتيح أمر بسيط عند استخدام التشفير بالمفاتيح العامة، مقارنةً بالمتاعب المرتبطة بالمصافحة المطلوبة بمراكز توزيع المفاتيح التي تستخدم التشفير التقليدي. في الواقع، نحتاج لشكلٍ من أشكال البروتوكولات لذلك، يتضمّن في الغالب استخدام وكيل مركزي وإجراءات ليست أبسط ولا أكثر كفاءة من تلك المُستخدَمة في التشفير التقليدي.

يتكون نظام التشفير بالمفاتيح العامة من ستة مكوّنات (انظر الشكل 3-7 (a)):

- الرسالة غير المُشفَّرة (plaintext): الرسالة الأصلية أو البيانات المقروءة التي تُغذَّي بها الخوارزمية كمُدخل.
- خوارزمية التشفير: تقوم خوارزمية التشفير بتحويلات مختلفة على نص الرسالة الأصلية.
- مفتاح عام ومفتاح خاص: يُختار هذا الزوج من المفاتيح بحيث إذا استُخدِم أحدهما للتشفير، يُستخدَم الآخر في إزالة التشفير. تعتمد التحويلات المحددة التي تؤديها خوارزمية التشفير على المفتاح العام أو الخاص والذي يُعطى كمُدخل.
- الرسالة المُشفَّرة (ciphertext): الرسالة المخلوطة التي تنتج كمُخرج من الخوارزمية وتعتمد على النص الأصلي للرسالة وعلى المفتاح. لرسالة معطاة ومفتاحين مختلفين للتشفير، تُنتِج خوارزمية التشفير نصين مُشفَّرين مختلفين.
- خوارزمية إزالة التشفير: تأخذ تلك الخوارزمية النص المُشفَّر والمفتاح المناظر وتعطي النص الأصلي.



(a) تشفير



(b) توثيق

الشكل 3-7: التشفير بالمفاتيح العامة.

كما يظهر من الأسماء، يُعلَن عن المفتاح العام للآخرين لكي يستخدموه، بينما يُحفَظ المفتاح الخاص مع مالكه فقط. وتعتمد خوارزمية التشفير بالمفاتيح العامة متعددة الأغراض على مفتاح للتشفير وآخر مختلف - لكنه ذو علاقة بالمفتاح الأول - لإزالة التشفير.

تشمل خوارزمية التشفير بالمفاتيح العامة الخطوات الأساسية الآتية:

1. يولّد كل مستخدم زوجاً من المفاتيح لاستخدامه في تشفير الرسائل وإزالة تشفيرها.
2. يضع كل مستخدم أحد المفتاحين في مُسجِّل عام أو ملف يسهل الوصول إليه من قبل الآخرين (يُمثل المفتاح العام)، بينما يحتفظ المستخدم بالمفتاح الآخر المناظر سرياً. كما يظهر من الشكل 3-7 (a)، يحتفظ كل مستخدم أيضاً بمجموعة من المفاتيح العامة التي حصل عليها من الآخرين.
3. إذا أراد بوب إرسال رسالة خاصّة إلى أليس، يقوم بوب بتشفير الرسالة مستخدماً مفتاح أليس العام.
4. عندما تتلقى أليس الرسالة، تستخدم مفتاحها الخاص لإزالة التشفير. لا يمكن لأي شخص آخر يتلقى الرسالة (بما في ذلك بوب) أن يحل شفرتها، لأن أليس هي الوحيدة التي تعرف مفتاحها الخاص.

بهذه الطريقة، يمكن لكل المشاركين الوصول للمفاتيح العامة، أما المفاتيح الخاصة فيتم توليدها محلياً من قبل كل مشارك ولا تحتاج أبداً إلى توزيع. طالما أن المستخدم يحمي مفتاحه الخاص، فسيكون الاتصال التالي آمناً. في أي وقت، يمكن أن يغيّر المستخدم المفتاح الخاص ويقوم بنشر المفتاح العام المناظر ليحل محل المفتاح العام القديم. عادةً ما يُطلق على المفتاح المستخدم في التشفير التقليدي اسم المفتاح السرّي. ويُطلق على المفتاحين المستخدمين في التشفير بالمفاتيح العامة المفتاح العام والمفتاح الخاص. يبقى المفتاح الخاص سرّياً دائماً، لكنّه يُعرف باسم المفتاح الخاص بدلاً من المفتاح السري لتفادي الخلط مع التشفير التقليدي.

2-3-3 تطبيقات أنظمة التشفير بالمفاتيح العامة

قبل أن نواصل، نحتاج لتوضيح إحدى سمات أنظمة المفاتيح العامة والتي بدونها يُحتمل أن يحدث خلط في المفاهيم. تتميز أنظمة التشفير بالمفاتيح العامة باستخدام نوع من خوارزميات التشفير بمفتاحين: أحدهما خاص والآخر عام يُنشر بشكلٍ علني. حسب التطبيق، يستخدم المُرسِل مفتاحه الخاص أو المفتاح العام للمُستلِم أو كليهما لأداء بعض عمليات التشفير. بشكلٍ عام، يمكن أن نصنّف استخدام أنظمة التشفير بالمفاتيح العامة إلى ثلاثة أصناف:

- التشفير/إزالة التشفير: يشفّر المُرسِل الرسالة باستخدام المفتاح العام للمُستلِم.
- التوقيع الرقمي: "يوقّع" المُرسِل على الرسالة باستخدام مفتاحه الخاص. يتم ذلك باستخدام خوارزمية تشفير تُطبّق على الرسالة كلها أو على كتلة بيانات صغيرة يتم إنشاؤها كدالة في الرسالة.
- تبادل المفاتيح: يتعاون طرفان لتبادل مفتاح جلسة. يمكن ذلك بعدة طرق مختلفة تتضمّن استخدام المفاتيح الخاصة لأحد الطرفين أو كليهما.

تناسب بعض الخوارزميات التطبيقات الثلاثة، بينما يمكن استخدام بعضها الآخر فقط مع واحد أو اثنين من هذه التطبيقات. ويبيّن الجدول 3-2 التطبيقات التي تدعمها الخوارزميات التي نوقشت في هذا الفصل: RSA، وديفي - هيلمان. يبيّن الجدول أيضاً معيار التوقيع الرقمي (Digital Signature Standard (DSS)) وتشفير المنحنى الإهليليجي (Elliptic-Curve Cryptography (ECC))، وسنتناولها أيضاً لاحقاً في هذا الفصل.

الجدول 3-2: تطبيقات أنظمة التشفير بالمفاتيح العامة.

الخوارزمية	التشفير/إزالة التشفير	التوقيع الرقمي	تبادل المفاتيح
خوارزمية RSA	نعم	نعم	نعم
خوارزمية ديفي - هيلمان	لا	لا	نعم
خوارزمية DSS	لا	نعم	لا
خوارزمية المنحنى الإهليليجي	نعم	نعم	نعم

3-3-3 متطلبات التشفير بالمفاتيح العامة

تعتمد أنظمة التشفير الموضحة في الشكل 3-7 على خوارزمية تشفير تستخدم مفتاحين متعلقين ببعضهما. افترض ديفي وهيلمان هذا النظام بدون توضيح أنّ مثل هذه الخوارزميات موجودة، غير أنهما عرضا الشروط التي يجب توفرها في تلك الخوارزميات [DIFF 76]:

1. أن يكون من السهل حسابياً قيام أحد الأطراف المتصلة B بتوليد زوج من المفاتيح (مفتاح عام PU_b وآخر خاص PR_b).
2. أن يكون من السهل حسابياً قيام المُرسِل A (بمعرفة المفتاح العام للمُستلِم والرسالة M المطلوب تشفيرها) بتشفير الرسالة وتوليد النص المُشفَّر المناظر:

$$C = \mathrm{E}(PU_b, M)$$

3. أن يكون من السهل حسابياً قيام المُستلِم B بإزالة تشفير الرسالة المُستقبَلة باستخدام مفتاحه الخاص لاستعادة الرسالة الأصلية:

$$M = \mathrm{D}(PR_b, C) = \mathrm{D}[PR_b, \mathrm{E}(PU_b, M)]$$

4. أن يكون من الصعب حسابياً أن يتوصل مهاجم لمعرفة المفتاح الخاص PR_b بمعرفة المفتاح العام PU_b.

5. أن يكون من الصعب حسابياً أن يستعيد مهاجم الرسالة الأصلية M بمعرفة المفتاح العام PU_b والنص المُشفَّر C.

يمكننا إضافة متطلب سادس، ورغم أهميته فإنه ليس ضرورياً لكل تطبيقات المفاتيح العامة:

6. يمكن أن يستخدم أي من المفتاحين للتشفير ويستخدم الآخر في إزالة التشفير:

$$M = \mathrm{D}[PU_b, \mathrm{E}(PR_b, M)] = \mathrm{D}[PR_b, \mathrm{E}(PU_b, M)]$$

4-3 خوارزميات التشفير بالمفاتيح العامة

الخوارزميتان الأكثر انتشاراً للتشفير بالمفاتيح العامة هما RSA وديفي - هيلمان. سنناقش كلتا الطريقتين في هذا الجزء، ثم نقدم لمحة سريعة عن خوارزميتين أخريين. سنستخدم هنا بعض المفاهيم الأولية لنظرية الأعداد؛ ويمكنك مراجعة الملحق A في نهاية الكتاب.

1-4-3 خوارزمية RSA للتشفير بالمفاتيح العامة

خوارزمية RSA هي إحدى طرق التشفير بالمفاتيح العامة والتي طُوِّرت في عام 1977 من قِبَل Ron Rivest وAdi Shamir وLen Adleman في معهد MIT ونشرت لأول مرة في عام 1978 [RIVE78]. منذ ذلك الحين وخوارزمية RSA تتربع بلا منافس كأكثر خوارزميات التشفير بالمفاتيح العامة قبولاً وانتشاراً. وRSA هي خوارزمية تشفير كتلة يكون فيها النص الأصلي والنص المُشفَّر أعداداً صحيحة ما بين 0 وn-1 (لقيمة معينة لـ n). يتم تشفير الرسالة M وإزالة تشفير النص المُشفَّر C كما يأتي:

$$C = M^e \bmod n$$
$$M = C^d \bmod n = (M^e)^d \bmod n = M^{ed} \bmod n$$

يجب أن يعرف كلٌّ من المُرسِل والمُستلِم قيم n وe، ولكن المُستلِم فقط هو الذي يعرف قيمة d. هذه خوارزمية تشفير بالمفاتيح العامة حيث المفتاح العام $KU = \{e, n\}$ والمفتاح الخاص $KR = \{d, n\}$. لكي تكون تلك الخوارزمية مُرضية للتشفير بالمفاتيح العامة، يجب أن تحقق المتطلبات الآتية:

1. من الممكن إيجاد قيم e، d، n بحيث $M^{ed} = M \bmod n$ لجميع القيم $M < n$.
2. من السهل نسبياً حساب M^e وC^d لجميع القيم $M < n$.
3. عدم القدرة على تعيين قيمة d بمعرفة e وn.

يمكن تحقيق المتطلب الأول والثاني بسهولة، أما المتطلب الثالث فيمكن تحقيقه باستخدام قيم كبيرة لـ e وn.

توليد المفاتيح

اختر عددين أوليين p وq بحيث $p \neq q$

احسب $n = p \times q$

احسب $\phi(n) = (p-1)(q-1)$

اختر e بحيث $\gcd(\phi(n),e)=1,\ 1<e<\phi(n)$

احسب d بحيث $de \bmod \phi(n) = 1$

المفتاح العام: $KU=\{e, n\}$، المفتاح الخاص: $KR=\{d, n\}$

المفتاح الخاص: $KR=\{d, n\}$

التشفير

النص غير المشفّر: $M < n$

النص المشفّر: $C = M^e \bmod n$

إزالة التشفير

النص المشفّر: C

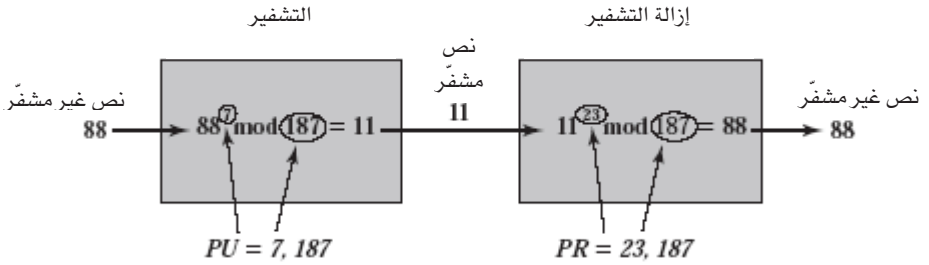
النص غير المشفّر: $M = C^d \bmod n$

الشكل 3-8: خوارزمية RSA.

يلخص الشكل 3-8 خوارزمية RSA. تبدأ الخوارزمية باختيار عددين أوليين p وq، ثم تحسب حاصل ضربهما n (والذي يُستخدم كمعامل الباقي (modulus) للتشفير وإزالة التشفير). بعد ذلك نحتاج الكمية $\phi(n)$ والتي تُعرف باسم بدالة أويلر (Euler totient) للعدد n، وهي تمثل عدد الأعداد الصحيحة الموجبة الأقل من n وتُعدُّ أعداداً أولية نسبياً مع n. ثم نختار عدداً صحيحاً e يكون أولياً نسبياً مع $\phi(n)$ (أي أن القاسم المشترك الأكبر بين $\phi(n)$ وe هو 1). وأخيراً نحسب d كالمعكوس الضربي لـ e في مجال باقي القسمة على $\phi(n)$. يمكن إثبات توفر الخصائص المطلوبة في d وe.

افترض أن المستخدم A نشر مفتاحه العام وأن المستخدم B يريد إرسال الرسالة M إلى A. عندئذ يقوم B بحساب $C = M^e \pmod{n}$ ويرسل C. بعد استقبال ذلك النص المُشفَّر، يقوم المستخدم A بحساب $M = C^d \pmod{n}$. يوضِّح الشكل 3-9 مثالاً مشتقاً من [SING99]. لهذا المثال، يتم توليد المفاتيح على النحو الآتي:

1. اختر عددين أوليين $p = 17$ و$q = 11$.
2. احسب $n = pq = 17 * 11 = 187$.
3. احسب $\phi(n) = (p\text{-}1)(q\text{-}1) = 16 \times 10 = 160$.
4. اختر العدد e بحيث يكون أولياً نسبياً مع $\phi(n) = 160$ ويكون أقل من $\phi(n)$؛ دعنا نختار $e = 7$.
5. أوجد d بحيث $de \bmod 160 = 1$ و$d < 160$؛ تكون قيمة $d = 23$ لأن $23 \times 7 = 161 = 1 \times 160 + 1$.



الشكل 3-9: مثال لخوارزمية RSA.

المفتاحان الناتجان هما العام $PU = \{7, 187\}$ والخاص $PR=\{23, 187\}$. يبيّن المثال الآتي استخدام هذين المفاتيح في تشفير الرسالة $M = 88$:

$88^7 \bmod 187 = [(88^4 \bmod 187)\times(88^2 \bmod 187)\times(88^1 \bmod 187)] \bmod 187$
$88^1 \bmod 187 = 88$
$88^2 \bmod 187 = 7744 \bmod 187 = 77$
$88^4 \bmod 187 = 59{,}969{,}536 \bmod 187 = 132$
$88^7 \bmod 187 = (88\times77\times132) \bmod 187 = 894{,}432 \bmod 187 = 11$

ولإزالة التشفير نقوم بالآتي:

$11^{23} \bmod 187 = [(11^1 \bmod 187)\times(11^2 \bmod 187)\times(11^4 \bmod 187)\times$
$(11^8 \bmod 187)\times(11^8 \bmod 187)] \bmod 187$
$11^1 \bmod 187 = 11$
$11^2 \bmod 187 = 121 \bmod 187 = 121$
$11^4 \bmod 187 = 14{,}641 \bmod 187 = 55$
$11^8 \bmod 187 = 214{,}358{,}881 \bmod 187 = 33$
$11^{23} \bmod 187 = (11\times121\times55\times33\times33) \bmod 187 = 79{,}720{,}245 \bmod 187 = 88$

هناك طريقتان لاختراق خوارزمية RSA. الأولى هي الطريقة الاستقصائية (brute-force): أي بتجريب كل قيم المفاتيح الخاصة، ولذا فكلما زاد عدد البتات في e وd، ازداد أمن الخوارزمية. غير أنه بسبب الحسابات المعقدة

المستخدمة لتوليد المفتاح وفي عمليتي التشفير وإزالة التشفير، فكلما زاد طول المفتاح كان النظام أبطأ في التشغيل.

تركّز معظم مناقشات تحليل الشفرة لخوارزمية RSA على مهمة تحليل n إلى عاملين أوليين. وعندما تكون قيم n كبيرة ولها عوامل أولية كبيرة يكون تحليلها إلى عواملها الأولية من المسائل الصعبة حسابياً، لكن ليس بنفس الدرجة التي كانت عليها من قبل. شهد عام 1977 برهاناً رائعاً على ذلك، حيث تحدى مخترعو خوارزمية RSA الثلاثة قراء مجلة Scientific American الأمريكية لفك شفرة نشروها في عمود "ألعاب رياضية" بمجلة Martin Gardner [GARD77]. وعرض المخترعون مائة دولار جائزة لكل جملة يمكن استعادتها من الرسالة المُشفَّرة، وكانوا قد توقعوا أن مثل هذا الحدث قد يستغرق حوالي 40 كوادرليون سنة (أي 40×10^{15} سنة). في أبريل/نيسان عام 1994, عملت مجموعة من خلال الإنترنت واستخدمت أكثر من 1600 جهاز حاسب واستطاعت أن تكسب الجائزة بعد ثمانية شهور فقط من العمل [LEUT94]. استخدم هذا التحدي مفتاحاً عاماً طوله 129 رقماً عشرياً (أي حوالى 428 بتاً). وهذه النتيجة لن تبطل استخدام RSA؛ ولكنها تعني ببساطة أنه يجب استخدام مفاتيح بأطوال كبيرة. حالياً تُعدُّ المفاتيح بأطوال 1024 بتاً (أي حوالي 300 رقم عشري) قوية بما فيه الكفاية لكل التطبيقات تقريباً.

2-4-3 تبادل المفاتيح بطريقة ديفي - هيلمان (Diffie-Hellman)

ظهرت أول خوارزمية للمفاتيح العامة في الورقة البحثية المهمة من قِبَل ديفي وهيلمان، وقد عَرَّفت التشفير بالمفاتيح العامة [DIFF76] الذي عُرِف بشكلٍ عام بطريقة ديفي - هيلمان لتبادل المفاتيح. وتُستخدم هذه الطريقة في عدد من المنتجات التجارية.

الغرض من الخوارزمية هو تمكين المستخدمين من تبادل مفتاح سري بشكلٍ آمن، ومن ثم يكون بوسعهما بعد ذلك استخدام المفتاح في تشفير

الرسائل اللاحقة. يقتصر دور الخوارزمية نفسها على تبادل المفاتيح. تعتمد خوارزمية ديفي - هيلمان في فعاليتها على صعوبة حساب اللوغاريتمات المنفردة (discrete logarithms). باختصار، يمكننا تعريف اللوغاريتم المنفرد بالطريقة الآتية: أولاً، نُعرِّف الجذر البدائي لعدد أولي p بأنه الجذر الذي تولِّد القوى الأسيِّة له كل الأعداد الصحيحة من 1 إلى p-1. أي إذا كان a جذراً بدائياً للعدد الأولي p، فإن الأعداد الآتية:

$a \bmod p, a^2 \bmod p, \ldots, a^{p-1} \bmod p$

أعداد منفردة وتمثل الأعداد الصحيحة من 1 إلى p-1 بتبديل ما.

لأي عدد صحيح b أقل من p وجذر بدائي a للعدد الأولي p، يمكن أن نوجد أسًّا فريداً i بحيث $0 \le i \le (p - 1)$ ويحقق العلاقة التالية:

$b = a^i \bmod p$

يُعرَف الأس i باسم اللوغاريتم المنفرد، أو دليل العدد b للأساس a في مجال باقي القسمة على p. سنشير للوغاريم المنفرد بالرمز:

$\mathrm{dlog}_{a,p}(b)$

❖ الخوارزمية:

بهذه الخلفية يمكننا أن نُعرِّف خوارزمية ديفي - هيلمان لتبادل المفاتيح كما هي موضحة باختصار بالشكل 3-10. في هذه الطريقة، هناك عددان معروفان للجميع: العدد الأولي q والعدد الصحيح α الذي يمثل الجذر البدائي للعدد q. افترض أن مستخدمين A وB يريدان تبادل مفتاح. يختار المستخدم A عدداً صحيحاً عشوائياً X_A أقل من q، ويحسب

$$Y_A = \alpha^{X_A} \bmod q$$

بنفس الطريقة وبشكلٍ مستقل يختار المستخدم B عدداً صحيحاً عشوائياً X_B أقل من q ويحسب

$$Y_B = \alpha^{X_B} \bmod q$$

يحتفظ كل طرف بقيمة X الخاصة به ويعلن للطرف الآخر عن القيمة Y الناتجة لديه من الحساب السابق. يحسب المستخدم A المفتاح من

$$K = (Y_B)^{X_A} \bmod q$$

ويحسب المستخدم B المفتاح من

$$K = (Y_A)^{X_B} \bmod q$$

لاحظ أن كلا المستخدمين يحصل على نفس النتجة (أي نفس المفتاح) من الحساب السابق؛ يمكن إثبات ذلك كما يأتي:

$$K = (Y_B)^{X_A} \bmod q$$
$$= (\alpha^{X_B} \bmod q)^{X_A} \bmod q$$
$$=(\alpha^{X_B})^{X_A} \bmod q$$
$$=\alpha^{X_B X_A} \bmod q$$
$$=(\alpha^{X_A})^{X_B} \bmod q$$
$$=(\alpha^{X_A} \bmod q)^{X_B} \bmod q$$
$$=(Y_A)^{X_B} \bmod q$$

المحصّلة النهائية أن كلا الطرفين تمكنا من تبادل قيمة سرية. علاوة على ذلك، فلأن X_A وX_B قيم خاصة بكل طرف، فلا يتوفر لخصم سوى القيم q وα وY_A وY_B فقط. وعليه يحتاج الخصم أن يقوم بحساب لوغاريتم منفرد لتحديد قيمة المفتاح. على سبيل المثال، لتحديد قيمة المفتاح الخاص بالمستخدم B، يجب أن يحسب الخصم

$$X_B = \text{dlog}_{\alpha, q}(Y_B)$$

ومن ثم يمكن للخصم بعد ذلك حساب المفتاح K بنفس الطريقة التي يحسب بها المستخدم B.

يكمن أمن تبادل المفاتيح بطريقة ديفي - هيلمان في الحقيقة الآتية: "بينما من السهل نسبياً حساب باقي قسمة رقم أسي على عدد أولي، فإنه من الصعب جداً حساب قيم اللوغاريتمات المنفردة". وعند استخدام أعداد أولية كبيرة تُعدُّ هذه المهمة الأخيرة عملية غير ممكنة.

العناصر العمومية للجميع

q عدد أولي

α عدد أقل من q ويمثل جذراً بدائياً لـ q

توليد مفتاح المستخدم A

اختر عدداً خاصاً X_A أقل من q

احسب المفتاح العام Y_A من $Y_B = \alpha^{X_B} \bmod q$

توليد مفتاح المستخدم B

اختر عدداً خاصاً X_B أقل من q

احسب المفتاح العام Y_B من $Y_A = \alpha^{X_A} \bmod q$

توليد المفتاح السري للمستخدم A

$$K = (Y_B)^{X_A} \bmod q$$

توليد المفتاح السري للمستخدم B

$$K = (Y_A)^{X_B} \bmod q$$

الشكل 3-10: خوارزمية ديفي - هيلمان لتبادل المفاتيح.

لنعرض هنا مثالاً. افترض أن تبادل المفاتيح يعتمد على استخدام العدد الأولي $q = 353$ وجذر بدائي للعدد 353، في هذه الحالة $\alpha = 3$. وافترض أن الطرفين A وB يختاران عددين خاصين $X_A = 97$ و$X_B = 233$ على التوالي. يقوم كل طرف بحساب مفتاحه العام على النحو الآتي:

الطرف A:

$$Y_A = 3^{97} \; mod \; 353 = 40$$

الطرف B:

$$Y_B = 3^{233} \; mod \; 353 = 248$$

بعد أن يتبادلا هذين القيمتين، يمكن لكل واحد منهما أن يحسب المفتاح السري المشترك:

الطرف A:

$$K = (Y_B)^{X_A} mod \; 353 = \; 248^{97} mod 353 = 160$$

الطرف B:

$$K = (Y_A)^{X_B} mod \; 353 = \; 40^{233} mod 353 = 160$$

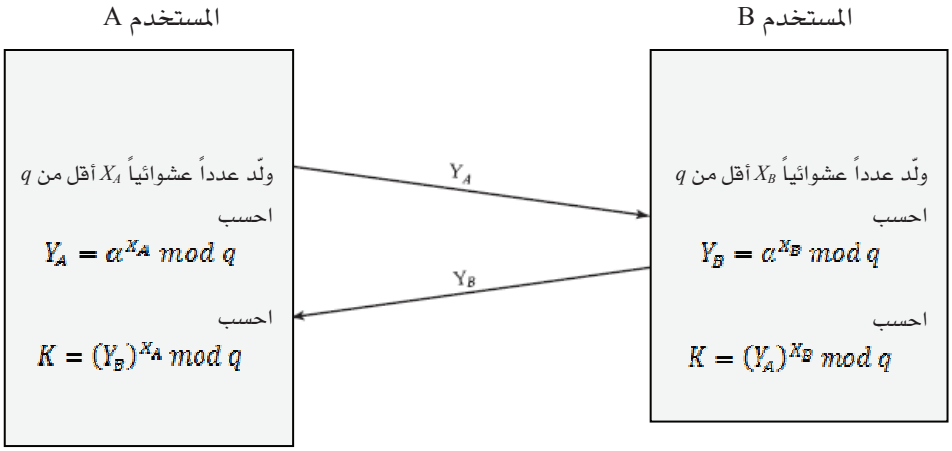
افترض أن مهاجماً E توفرت لديه المعلومات الآتية:

$q = 353$; $\alpha = 3$; $Y_A = 40$; $Y_B = 248$

في هذا المثال البسيط يمكن تحديد قيمة المفتاح السرّي المشترك على أنها 160 بالطريقة الاستقصائية (أي بتجريب كل الاحتمالات الممكنة). على وجه التحديد يمكن للمهاجم E أن يحدد قيمة المفتاح المشترك باكتشاف حل للمعادلة $3^a \bmod 353 = 40$ أو المعادلة $3^b \bmod 353 = 248$. ينطوي الحل الاستقصائي على تجريب القيم المختلفة لقوى 3 وحساب باقي قسمة الناتج على 353، ثم التوقف عندما يكون ذلك الباقي 40 أو 248. تتحقق الإجابة المطلوبة بالقيمة $a = 97$ التي عندها $3^{97} \bmod 353 = 40$. عند استخدام أعداد كبيرة تصبح مسألة الحصول على المفتاح السرّي المشترك غير عملية.

❖ بروتوكولات تبادل المفاتيح:

يبيّن الشكل 3-11 بروتوكولاً بسيطاً يستخدم حسابات ديفي - هيلمان . افترض أن المستخدم A يريد بدء اتصال مع المستخدم B ويريد أن يستخدم مفتاحاً سرياً لتشفير الرسائل على ذلك الاتصال. يمكن أن يولّد المستخدم A مفتاحاً خاصاً X_A للاستخدام مرة واحدة ويحسب Y_A ويرسلها إلى المستخدم B. يقوم المستخدم B بالرد بتوليد مفتاح خاص X_B وحساب Y_B وإرسال قيمتها إلى A. الآن يمكن أن يحسب كل مستخدم قيمة المفتاح المشترك. من الضروري معرفة القيم العامة q وα مسبقاً. أيضاً يمكن بدلاً من ذلك أن يختار المستخدم A قيماً لـ q وα ويضمنها في الرسالة الأولى التي يرسلها إلى B.



الشكل 3-11: تبادل المفاتيح بطريقة ديفي - هيلمان.

ومثال آخر لاستخدام خوارزمية ديفي - هيلمان، افترض وجود مجموعة من المستخدمين (مثلاً كل المستخدمين على شبكة اتصالات محلية)، وأن كل واحد منهم يولّد قيمة خاصة X_A للاستخدام على المدى الطويل ويحسب قيمة عامة Y_A. وافترض أن هذه القيم العامة (واحدة لكل مستخدم) والقيمتين

العموميتين q وα يتم تخزينها في دليل مركزي على الشبكة. في أي وقت يمكن للمستخدم B الوصول القيمة العامة للمستخدم A ويحسب مفتاحاً سرّياً يستخدمه في تشفير الرسائل التي يرسلها إلى المستخدم A. إذا توفرت الثقة في أمن الدليل المركزي، يمكن لهذه الطريقة توفير السرية ودرجة من درجات التوثيق (التحقق من الهوية). يرجع ذلك إلى أن A وB فقط يمكنهما تعيين المفتاح، ومن ثَمَّ لا يمكن لمستخدم آخر أن يقرأ الرسالة (ومن ثم تتحقق السرية). أيضاً المستخدم A يعرف أن المستخدم الوحيد الذي يمكنه توليد تلك الرسالة المُشفَّرة باستخدام هذا المفتاح هو B (ومن ثم يتحقق التوثيق). غير أن هذه الطريقة لا تحمي ضد هجمات إعادة التشغيل (replay).

❖ هجوم الرجل في الوسط (Man-in-the-Middle):

البروتوكول المبين في الشكل 3-11 غير آمن ضد هجوم الرجل في الوسط. لتوضيح ذلك افترض أن أليس وبوب يريدان تبادل المفاتيح، وأن دارث (Darth) خصم لهما. يتم الهجوم على النحو الآتي:

1. تستعد دارث للهجوم بتوليد مفتاحين خاصين عشوائيين X_{D1} وX_{D2}، وبعد ذلك تحسب المفاتيح العامة المناظرة Y_{D1} وY_{D2}.
2. ترسل أليس Y_A إلى بوب.
3. تعترض دارث الرسالة وتحصل على Y_A، وترسل Y_{D1} بدلاً منه إلى بوب. تحسب دارث أيضاً

$$K2 = (Y_A)^{X_{D2}} \bmod q$$

4. يتلقى بوب Y_{D1} ويحسب $K1 = (Y_{D1})^{X_B} \bmod q$.
5. يرسل بوب Y_B إلى أليس.
6. مرة أخرى تعترض دارث الرسالة وتحصل على Y_B، وترسل Y_{D2} إلى أليس. تحسب دارث

$$K1 = (Y_B)^{X_{D1}} \bmod q$$

7. تتلقى أليس Y_{D2} وتحسب $K2 = (Y_{D2})^{X_A} \bmod q$.

في هذه النقطة، يعتقد كلٌّ من بوب وأليس أنهما يشتركان في المفتاح السري، لكن الحقيقة أن بوب يشارك دارث في المفتاح السري $K1$ وأليس تشارك دارث في المفتاح السري $K2$. كل المراسلات المستقبلية بين بوب وأليس تمر على دارث كما يلي:

1. ترسل أليس الرسالة M مُشفَّرة بالمفتاح $K2$.
2. تعترض دارث الرسالة المُشفَّرة من أليس وتزيل التشفير لاستعادة M.
3. على حسب هدف دارث قد ترسل دارث الرسالة M أو أي رسالة أخرى M' مُشفَّرة بالمفتاح $K1$ إلى بوب. في الحالة الأولى تريد دارث التنصت فقط على الاتصال بين بوب وأليس دون تعديل للرسائل. أما في الحالة الثانية فتريد دارث تعديل الرسائل بين بوب وأليس.

بروتوكول تبادل المفاتيح عرضة لمثل هذا الهجوم لأنه لا يتحقق من هوية المشاركين. ويمكن التغلب على هذا الضعف باستخدام التوقيع الرقمي وشهادات المفاتيح العامة؛ وسوف نستعرض تلك الموضوعات لاحقاً في هذا الفصل وفي الفصل الرابع.

3-4-3 الخوارزميات الأخرى للتشفير بالمفاتيح العامة

لاقت خوارزميتان أخريان للتشفير بالمفاتيح العامة قبولاً تجارياً؛ هما: معيار التوقيع الرقمي DSS والتشفير بالمنحنى الإهليليجي ECC.

❖ معيار التوقيع الرقمي (DSS):

نشر المعهد الوطني للمعايير والتقنية (NIST) معياراً فيدرالياً لمعالجة البيانات برقم FIPS PUB 186، وعُرِف بمعيار التوقيع الرقمي (DSS). ويَستخدَم معيار التوقيع الرقمي دالة التحوير SHA-1 ويوفِّر طريقة جديدة للتوقيع الرقمي

تُعرَف بخوارزمية التوقيع الرقمي (DSA). وقد تم اقتراح معيار التوقيع الرقمي في الأصل في عام 1991 وعُدِّل في عام 1993 استجابةً لتعليقات الجمهور حول أمن هذه الطريقة. تم تعديله مرة أخرى بشكلٍ بسيط في عام 1996. ويَستخدَم معيار التوقيع الرقمي خوارزمية مصممة لتوفير وظيفة التوقيع الرقمي فقط. فعلى خلاف RSA، لا يمكن استخدامه للتشفير أو تبادل المفاتيح.

❖ التشفير بالمنحنى الإهليليجي (ECC):

تستخدم الأغلبية العظمى من المنتجات والمعايير التي توظِّف المفاتيح العامة في التشفير والتوقيع الرقمي خوارزمية RSA. زاد طول البتات اللازم للاستخدم الآمن لخوارزمية RSA في السنوات الماضية، مما زاد من عبء المعالجة المطلوبة على التطبيقات التي تستخدم RSA. وتظهر آثار ذلك على وجه الخصوص في مواقع التجارة عبر الإنترنت حيث تُجري أعداد كبيرة من المُعامَلات التجارية الآمنة. بدأ مؤخراً نظام جديد في تحدي RSA، وهو التشفير بالمنحنى الإهليليجي (ECC). بدأ ECC في الظهور بالفعل في جهود المعايير الموحدة، بما في ذلك معيار IEEE P1363 للتشفير بالمفاتيح العامة.

عامل الجذب الرئيس لـ ECC مقارنةً بـ RSA هو قدرته على توفير أمن مماثل لكن باستخدام عدد أقل جداً من البتات، مما يؤدي إلى تخفيض عبء المعالجة. من ناحية أخرى، رغم أن نظرية ECC كانت معروفة لفترة ليست بالقصيرة، فإن منتجات ECC لم تبدأ بالظهور، ولم يستحوذ الأسلوب الجديد على اهتمام كافٍ في مجال تحليل الشفرة لتقصي نقاط الضعف فيه، إلا مؤخراً. وعليه فمستوى الثقة في ECC ليس بنفس درجة الثقة في RSA.

يُعدُّ شرح ECC أكثر صعوبة بشكلٍ جوهري من RSA وديفي - هيلمان، ومعالجته رياضياً بشكلٍ كامل خارج نطاق هذا الكتاب. تعتمد تلك الطريقة على استخدام تركيب رياضي يُعرف بالمنحنى الإهليليجي.

5-3 التوقيع الرقمي

يمكن استخدام التشفير بالمفاتيح العامة بطريقة أخرى، كما هو موضَّح بالشكل 3-7 (b). افترض أن بوب يريد إرسال رسالة إلى أليس، رغم أنّه ليس من المهم أن تبقى الرسالة سرية، يريد بوب من أليس أن تكون متأكّدة من أن الرسالة هي في الحقيقة منه. وفي هذه الحالة يستخدم بوب مفتاحه الخاص لتشفير الرسالة. وعندما تستلم أليس النص المُشفَّر، تجد أنه بإمكانها أن تزيل التشفير بمفتاح بوب العام، وعليه تثبت أن الرسالة لا بد أنها شُفّرت من قِبَل بوب. لا أحد غيره لديه مفتاح بوب الخاص ولذا لا أحد سواه يمكنه توليد النص المُشفَّر الذي يمكنها أن تزيل تشفيره بمفتاح بوب العام. لذا، تخدم الرسالة المُشفَّرة بكاملها كـ "توقيع رقمي". وعلاوةً على ذلك، من المستحيل تعديل الرسالة بدون الوصول إلى مفتاح بوب الخاص، لذا يتم التحقق من الرسالة من ناحية المصدر ومن ناحية سلامة البيانات.

في الطريقة السابقة، تم تشفير الرسالة بكاملها. ورغم أن ذلك يُفيد في التحقق من هوية المُرسِل والتأكد من سلامة محتويات الرسالة، فإنه يتطلب كثيراً من حيز التخزين. وينبغي الاحتفاظ بكل مستند في شكله الأصلي (غير المُشفَّر) للاستخدام في الأغراض العملية. ويجب أيضاً الاحتفاظ بنسخة من النص المُشفَّر كي يتسنى التحقق من هوية الأصل ومن المحتويات في حالة نشوب نزاع. وطريقة أخرى أكثر كفاءة لتحقيق نفس النتيجة هي تشفير كتلة صغيرة من البتات تحسب كدالة في الرسالة. ويُطلق على مثل هذه الكتلة الموثّق (authenticator)، ويجب أن يتوفر لها خاصية عدم إمكانية تغيير المستند دون الحاجة إلى تغيير كتلة الموثّق. إذا تم تشفير الموثّق بمفتاح المُرسِل الخاص، فيمكن استخدمه بمثابة توقيع للتحقق من هوية الأصل وكذلك المحتوى والتسلسل. يمكن أن تستخدم خوارزمية لكود التحوير الآمن، كـ SHA-1 مثلاً، لهذا الغرض. يبيّن الشكل 3-2 (b) هذا السيناريو.

من المهم التأكيد على أن عملية التشفير التي وصفت للتو لا توفر السرية. بمعنى أن الرسالة المرسلة آمنة من التعديل لكنها ليست آمنة من التنصت. وهذا الأمر واضح في حالة التوقيع المعتمد على جزء من الرسالة، لأن بقيّة الرسالة ترسل واضحة. حتى في حالة التشفير الكامل، ليس هناك حماية للسريّة لأن أيّ مراقب يمكن أن يزيل تشفير الرسالة باستخدام المفتاح العام للمُرسِل.

6-3 إدارة المفاتيح

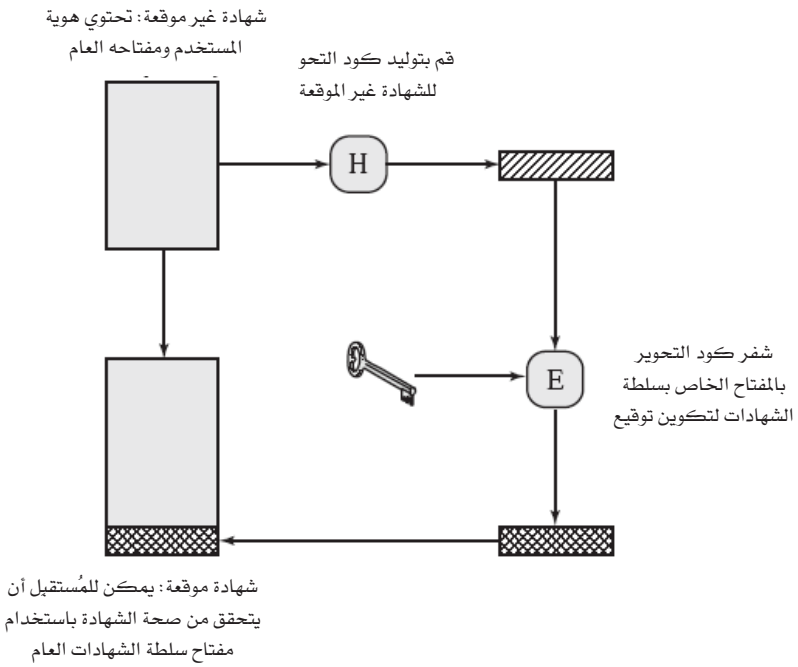
أحد الأدوار الرئيسة للتشفير بالمفاتيح العامة هو تناوله لمسألة توزيع المفاتيح. في الحقيقة هناك جانبان محدَّدان لاستخدام التشفير بالمفاتيح العامة لهذا الغرض:

- توزيع المفاتيح العامة.
- استخدام التشفير بالمفاتيح العامة لتوزيع المفاتيح السرية.

سنتناول كلاً من هذين الموضوعين تباعاً.

1-6-3 شهادات المفاتيح العامة

بحسب الظاهر، النقطة المحورية في التشفير بالمفاتيح العامة هي أن المفتاح العام معلن. وعليه، إذا توفرت خوارزمية مفتاح عام بشكلٍ واسع (كـ RSA)، يمكن لأحد المشتركين أن يرسل مفتاحه العام إلى أي مشارك آخر أو أن يذيع المفتاح إلى جميع المشتركين. رغم أن هذه الطريقة مريحة، فإنها تعاني من ضعف أساسي. يمكن لأي شخص أن يزيف مثل هذا الإعلان العام. بمعنى يمكن أن يتظاهر مستخدم ما بأنه المستخدم A ويرسل مفتاحه العام إلى مشارك آخر أو يذيع مثل ذلك المفتاح العام. إلى أن يحين الوقت الذي يكتشف فيه المستخدم A التزييف الذي حصل وينبه المشاركين الآخرين إليه، يكون المزوِّر قد تمكن من قراءة كل الرسائل المُشفَّرة الموجهة إلى A، كما يمكنه استخدام المفاتيح المزوَّرة للتحقّق.



الشكل 3-12: استخدام شهادة المفتاح العام.

يكمن حل هذه المشكلة في استخدام شهادة المفتاح العام. وبشكلٍ أساسي تتكون الشهادة من مفتاح عام بالإضافة إلى هوية المستخدم مالك المفتاح، وتوقَّع من طرف ثالث مؤتمن. عادةً ما يكون الطرف الثالث هيئة شهادات ((Certificate Authority (CA) مؤتمنة من قِبَل المستخدمين، كجهاز حكومي أو مؤسسة مالية. يستطيع المستخدم أن يقدم مفتاحه العام إلى السلطة بطريقة آمنة ويحصل على شهادة يمكنه بعد ذلك نشرها. ويمكن لأي شخص يحتاج المفتاح العام لذلك المستخدم الحصول على الشهادة والتحقق من صحتها من التوقيع المؤتمن الملحق بها. يبيّن الشكل 3-12 هذه العملية.

إحدى الطرق التي انتشرت بشكلٍ عام لإصدار شهادات المفاتيح العامة هي معيار X.509. تُستخدم شهادات X.509 في أكثر تطبيقات أمن الشبكات، بما في ذلك أمن بروتوكول الإنترنت (IP)، وبروتوكول طبقة المقابس الآمنة (SSL)، وبروتوكول المُعامَلات الإلكترونية الآمنة (SET)، وبروتوكول امتدادات بريد الإنترنت متعددة الأغراض الآمنة (S/MIME)، وسنناقشها جميعاً في الباب الثاني من هذا الكتاب. سنتناول معيار X.509 بالتفصيل في الفصل الرابع.

3-6-2 استخدام لمفاتيح العامة لتوزيع المفاتيح السرية

أحد المتطلبات الأساسية لكي يتمكن طرفان من الاتصال بشكلٍ آمن باستخدام التشفير التقليدي هو أن يشترك الطرفان في مفتاح سري. افترض أن بوب يريد عمل برنامج مراسلات يمكّنه من تبادل رسائل البريد الإلكتروني بشكلٍ آمن مع أي شخص آخر لديه اتصال بالإنترنت أو بشبكة أخرى مشتركة بينهما. افترض أن بوب يريد أن يقوم بذلك مستخدماً التشفير التقليدي. في التشفير التقليدي يجب أن يتوصل بوب والشخص الذي يراسله (مثلاً أليس) إلى طريقة للاشتراك معاً في مفتاح سري فريد لا يعرفه أحد غيرهما. كيف يمكنهما القيام بذلك؟

إذا كانت أليس في الغرفة المجاورة لبوب، يمكن أن ينشئ بوب مفتاحاً ويكتبه على قطعة من الورق أو يخزنه على قرص ويسلمه إلى أليس. لكن إذا كانت أليس على الجانب الآخر من القارة أو العالم، فماذا يمكن لبوب فعله؟ يمكن أن يُشفِّر بوب هذا المفتاح باستخدام التشفير التقليدي ويرسله بالبريد الإلكتروني إلى أليس، لكن هذا يعني أن بوب وأليس ينبغي أن يشتركا في مفتاح سري لتشفير هذا المفتاح السري الجديد. وعلاوة على ذلك، يعاني بوب والآخرون الذين يستخدمون نفس حزمة البريد الإلكتروني الجديدة من

المشكلة نفسها مع كل شخص آخر يُحتمل أن يراسلوه؛ حيث يتعين أن يشترك كل زوج من المراسلين في مفتاح سري فريد.

إحدى الطرق لحل هذه المشكلة هي استخدام طريقة ديفي - هيلمان لتبادل المفاتيح. هذه الطريقة واسعة الانتشار بحق، لكنها مع ذلك تعاني في شكلها البسيط من قصور واضح، حيث إنها لا توفر أي تحقق من هوية الأطراف المتصلة.

من البدائل القوية استخدام شهادات المفاتيح العامة. عندما يريد بوب أن يتصل بأليس، يمكن أن يقوم بالآتي:

1. يُجهِّز الرسالة.
2. يُشفِّر تلك الرسالة بالتشفير التقليدي بمفتاح جلسة يُستخدم لمرة واحدة.
3. يُشفِّر مفتاح الجلسة بطريقة التشفير بالمفاتيح العامة مستخدماً مفتاح أليس العام.
4. يلحق مفتاح الجلسة المُشفَّر بالرسالة ويرسلها إلى أليس.

فقط أليس تكون قادرة على إزالة تشفير مفتاح الجلسة في الرسالة التي تلقتها، ومن ثَمَّ يمكنها استعادة الرسالة الأصلية. إذا حصل بوب على مفتاح أليس العام من خلال شهادة المفتاح العام لأليس، فإنه سيكون متأكداً من صحة المفتاح.

7-3 توصيات للمطالعة

[CORM01] Cormen, T.; Leiserson, C.; Rivest, R.; and Stein, C. *Introduction to Algorithms.* Cambridge, MA: MIT Press, 2001.

[DIFF88] Diffie, W. "The First Ten Years of Public-Key Cryptography," *Proceedings of the IEEE*, May 1988. Reprinted in [SIMM92].

[MENE97] Menezes, A.; Oorshcot, P.; and Vanstone, S. *Handbook of Applied Cryptography.* Boca Raton, FL: CRC Press, 1997.

[SIMM92] Simmons, G., ed. *Contemporary Cryptology: The Science of Information Integrity.* Piscataway, NJ: IEEE Press, 1992.

[STIN06] Stinson, D. *Cryptography: Theory and Practice.* Boca Raton, FL: CRC Press, 2006.

8-3 مصادر للمعلومات على الويب

- NIST Secure Hashing Page: صفحة المعهد القومي للقياسات والتقنية عن التحوير الآمن تتضمن SHA FIPS والمستندات المتعلقة بها.
- Whirlpool: يحتوي على معلومات عن Whirlpool.
- RSA Laboratories: قائمة واسعة من المواد الفنية عن RSA وموضوعات أخرى عن التشفير.

9-3 مصطلحات رئيسة

Diffie-Hellman key exchange	تبادل المفاتيح بطريقة ديفي - هيلمان
digital signature	توقيع رقمي
Digital Signature Standard (DSS)	معيار التوقيع الرقمي
Elliptic-Curve Cryptography (ECC)	تشفير بالمنحنى الإهليلجي
HMAC	خوارزمية HMAC لتوثيق الرسائل
key exchange	تبادل المفاتيح
MD5	خوارزمية MD5 لخلاصة الرسالة
message authentication	توثيق الرسالة
Message Authentication Code (MAC)	كود توثيق الرسالة
message digest	خلاصة الرسالة
one-way hash function	دالة تحوير أحادية الاتجاه
private key	مفتاح خاص
public key	مفتاح عام
public-key certificate	شهادة مفتاح عام
public-key encryption	تشفير بالمفاتيح العامة
RSA	خوارزمية RSA
secret key	مفتاح سرّي
secure hash function	دالة تحوير آمنة
SHA-1	خوارزمية SHA-1
strong collision resistance	مقاومة قوية للتصادمات
weak collision resistance	مقاومة ضعيفة للتصادمات

10-3 أسئلة للمراجعة ومسائل

1-10-3 أسئلة للمراجعة

3-1 اذكر ثلاث طرق لتوثيق الرسائل.

3-2 عرِّف كود توثيق الرسائل.

3-3 صف باختصار الأساليب الثلاثة المبيَّنة بالشكل 3-2.

3-4 ما الخصائص التي يجب أن تتوفر لدالة التحوير لتكون مفيدة في توثيق الرسائل؟

3-5 ضمن سياق دالة التحوير، ما هي دالة الضغط؟

3-6 ما العناصر الرئيسة لنظام التشفير بالمفاتيح العامة؟

3-7 اذكر مع التعريف باختصار ثلاثة استخدامات للتشفير بالمفاتيح العامة.

3-8 ما الفرق بين المفتاح الخاص والمفتاح السرّي؟

2-10-3 مسائل

3-1 أحد الطرق الشائعة لكود توثيق الرسائل يُعرَف بخوارزمية توثيق البيانات (Data Authentication Algorithm) تعتمد على DES. وهذه الخوارزمية مُعرَّفة كمعيار FIPS PUB 113 من FIPS وكذلك كمعيار X9.17 من ANSI. يمكن تعريف تلك الخوارزمية على أنها استخدام نمط سَلسَلة كتل التشفير (CBC) لخوارزمية DES مع متجه تهيئة (IV) صفري (انظر الشكل 2-9). تقسَّم البيانات المراد توثيقها إلى كتل تتكون كتلة منها من 64 بتاً متجاورة: P_1، P_2، ... ، P_N. إذا تتطلب الأمر تزاد بتات حشوة قيمتها أصفار للكتلة النهائية من ناحية اليمين ليصبح طولها 64 بتاً. يتكون الـ MAC لكلٍّ منها من النص المُشفَّر بالكامل C_N أو عدد M من البتات تؤخذ من يسار كل كتلة حيث تتراوح قيمة M بين 16 و64. بيِّن أنه يمكن الحصول على نفس النتيجة باستخدام نمط التشفير بالتغذية المُرتدَّة (cipher feedback mode).

3-2 افترض دالة تحوير من 32 بتاً مُعرَفة كسلسلة من دالتين من 16 بتاً: XOR وRXOR، والمعرفتين في الجزء 3-2 كدالتين بسيطتين للتحوير.

a. هل يمكن لهذا المجموع التدقيقي كشف كل الأخطاء المكوّنة من عدد فردي من بتات الخطأ؟ وضح ذلك.

b. هل يمكن لهذا المجموع التدقيقي كشف كل الأخطاء المكوّنة من عدد زوجي من بتات الخطأ؟ في حالة الإجابة بلا، اذكر خصائص نمط الأخطاء التي يفشل معها هذا المجموع التدقيقي.

c. ما مدى فعالية هذه الدالة عند استخدمها كدالة تحوير للتوثيق؟

3-3 افترض أن $H(m)$ دالة تحوير مقاومة للتصادم تقوم بتحويل رسالة بطول اعتباطي إلى كود تحوير طوله n بت. هل صحيح أنه لجميع الرسائل x وx' بحيث $x' \neq x$ يكون $H(x') \neq H(x)$؟ وضح إجابتك.

4-3 a. افترض دالة تحوير مُعرَّفة كالتالي: تتكون الرسائل من سلسلة من الأعداد العشرية $M = (a_1, a_2, \ldots, a_t)$ ويُحسب كود التحوير h من

$$h = \left(\sum_{i=1}^{t} a_i \right) \bmod n$$

لقيمة معطاة n. هل تحقق دالة التحوير تلك أيًّا من متطلبات دوال التحوير المذكورة في الجزء 11-4؟ وضح إجابتك.

b. أعد إجابة الجزء السابق مع تغيير تعريف دالة التحوير كالتالي

$$h = \left(\sum_{i=1}^{t} (a_i)^2 \right) \bmod n$$

c. إحسب دالة التحوير المعرفة بالجزء (b) إذا كانت M=(189, 632, 900, 722, 349) وn=989.

5-3 تقدِّم هذه المسألة دالة تحوير مماثلة في جوهرها لـ SHA تعمل مع الحروف بدلاً من القيم الثنائية. تسمى بدالة التحوير toy tetragraph hash (tth). تقوم الدالة tth بحساب كود تحوير مكوّن من أربعة حروف لرسالة مكوّنة من سلسلة من الحروف. تقوم الدالة tth في البداية بتقسيم الرسالة إلى كتل كلٌّ منها مكوّن من 16 حرفاً مع إهمال المسافات وحروف الترقيم وحالة الأحرف. إذا كان طول الرسالة لا يقبل القسمة على 16 تضاف حروف حشوة قيمتها null. تبدأ الخوارزمية بمجموع تراكمي قيمته الابتدائية مكوّنة من أربعة أرقام (0, 0, 0, 0) يدخل لدالة الضغط لمعالجة الكتلة الأولى. تتكون دالة الضغط من جولتين. في الجولة الأولى تقوم بتحويل النص للكتلة التالية والمكوّن من 16 حرفاً إلى

مصفوفة مكوّنة من 4 صفوف و4 أعمدة، ثم بتحويل كل حرف في المصفوفة إلى رقم (حرف A يصبح 0 وحرف B يصبح 1 وهكذا). على سبيل المثال إذا كانت الكتلة مكوّن من الحروف التالية: ABCDEFGHIJKLMNOP تكون مصفوفة الحروف ومصفوفة الأرقام المناظرة كالآتي:

A	B	C	D
E	F	G	H
I	J	K	L
M	N	O	P

0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

ثم تقوم بجمع كل عمود mod 26 وإضافة الناتج للمجموع التراكمي mod 26. في هذا المثال يصبح المجموع التراكمي (24, 2, 6, 10). في الجولة الثانية تُستخدم المصفوفة من الجولة الأولى بعد تدوير الصف الأول لليسار خانة واحدة والصف الثاني خانتين وهكذا. في هذ الحالة تصبح المصفوفة:

B	C	D	A
G	H	E	F
L	I	J	K
P	O	N	M

1	2	3	0
6	7	4	5
11	8	9	10
15	14	13	12

الآن قم بإضافة كل عمود ثم أخذ باقي القسمة على 26 (أي حساب mod 26) وإضافة الناتج للمجموع التراكمي، يصبح الناتج (5, 7, 9, 11) والذي يأخذ كمُدخل للجولة الأولى لدالة الضغط لكتلة النص التالية. بعد معالجة الكتلة الأخيرة، قم بتحويل المجموع التراكمي النهائي لحروف. على سبيل المثال إذا كانت الرسالة ABCDEFGHIJKLMNOP فإن كود التحوير يكون FHJL.

a. ارسم أشكال مماثلة للأشكال 3-4 و3-5 لتبين منطق الأسلوب tth ومنطق دالة الضغط.

b. احسب كود التحوير للرسالة "I leave twenty million dollars to my friendly cousin Bill"

c. لتوضيح ضعف أسلوب tth، أوجد كتلة مكوّنة من 48 حرفاً لها نفس كود التحوير كالذي تم حسابه بالجزء b.

6-3 من الممكن استخدام دالة تحوير لبناء أسلوب لتشفير الكتل له تركيبة شبيهة بتلك الخاصة بـ DES. لكن نظراً لكون دالة التحوير دالة ذات اتجاه واحد (أي لا يمكن عكسها) وأسلوب التشفير أسلوباً انعكاسياً (حتى يتسنى حساب النص الأصلي من النص المُشفَّر)، بيِّن كيفية عمل ذلك.

7-3 قبل اكتشاف أيٍّ من أساليب التشفير بالمفاتيح العامة مثل RSA، تم إثبات إمكانية وجود التشفير بالمفاتيح العامة من الناحية النظرية. افترض الدوال:

$$f_1(x_1) = z_1;\ f_2(x_2, y_2) = z_2;\ f_3(x_3, y_3) = z_3$$

حيث جميع قيم x_i، y_i، z_i أعداد صحيحة بين 1 وN. يمكن تمثيل الدالة f_1 بمتجه M1 طوله N حيث تمثل قيمة العنصر k قيمة الدالة $f_1(k)$. بالمثل يمكن تمثيل الدوال f_2 وf_3 بالمصفوفات M2 وM3 من الرتبة $N \times N$. والمقصود هو تمثيل عملية التشفير وإزالة التشفير بعمليات بحث في جداول. وعملياً تكون هذه الجداول ذات أحجام كبيرة (لقيم N الكبيرة) لكن من حيث المبدأ يمكن تكوينها. يعمل هذا الأسلوب كما يلي:

- كوِّن M1 بترتيب عشوائي للأعداد من 1 إلى N (أي يظهر كل عدد صحيح من 1 إلى N مرة واحدة بالمتجه M1).
- كوِّن M2 بحيث يتضمن كل صف ترتيباً عشوائياً للأعداد من 1 إلى N.
- أخيراً املأ M3 بحيث تحقق الشرط الآتي:

$$f_3(f_2(f_1(k), p), k) = p$$

لجميع قيم k وp بحيث $1 \le k, p \le N$. يمكن التعبير عن ذلك بشكلٍ آخر كما يلي:

1. تأخذ M1 القيمة k كمُدخل وتعطي القيمة x كمُخرج.
2. تأخذ M2 القيم x وp كمُدخلات وتعطي القيمة z كمُخرج.
3. تأخذ M3 القيم z وk كمُدخلات وتعطي القيمة p كمُخرج.

بعد تكوين الجداول الثلاث يتم إعلانها للجميع.

a. يجب أن يكون واضحاً إمكانية تكوين M3 لتحقيق الشرط السابق. على سبيل المثال، املأ M3 للحالة البسيطة التالية:

M1=

5
4
3
2
1

M2=

5	2	3	4	1
4	2	5	1	3
1	3	2	4	5
3	1	4	2	5
2	5	3	4	1

M3=

اصطلاح: العنصر i بالمتجه M1 يناظر قيمة $k=i$. الصف i بالمصفوفة M2 يناظر قيمة $x=i$ والعمود j بالمصفوفة M2 يناظر قيمة $p=j$. الصف i بالمصفوفة M3 يناظر قيمة $z=i$ والعمود j بالمصفوفة M3 يناظر قيمة $k=j$.

b. صف استخدام مجموعة الجداول تلك لإجراء تشفير وإزالة التشفير بين مُستخدمين.

c. ناقش سبب كون ذلك الأسلوب آمناً.

3-8 قم بعمل تشفير وإزالة التشفير باستخدام أسلوب RSA كما هو مبين بالشكل 3-9 للحالات الآتية:

a. $p=3$; $q=11$; $e=7$; $M=5$
b. $p=5$; $q=11$; $e=3$; $M=9$
c. $p=7$; $q=11$; $e=17$; $M=8$
d. $p=11$; $q=13$; $e=11$; $M=7$
e. $p=17$; $q=31$; $e=7$; $M=2$

ملحوظة: قد يبدو إزالة التشفير صعباً لكن ببعض البراعة يمكن تبسيطه.

3-9 افترض أنك تمكنت من الحصول على المفتاح العام في نظام تشفير بالمفاتيح العامة يستخدم RSA، فما نص الرسالة M إذا كان المفتاح العام $e=5$ و$n=35$؟

3-10 ما قيمة المفتاح الخاص في نظام RSA إذا كان المفتاح العام $e=312$، $n=3599$؟

3-11 افترض أن لدينا مجموعة من الكتل المُشفَّرة بخوارزمية RSA، لكن ليس لدينا المفتاح الخاص. افترض أن $n=pq$، وأن e تمثل المفتاح العام. افترض أيضاً أن

شخصاً ما أخبرنا أنه يعرف أن أحد كتل النص الأصلي له عامل مشترك مع القيمة n؛ هل يمكن أن تساعدنا تلك المعلومة بأي شكلٍ من الأشكال؟

12-3 بيِّن كيف يمكن وصف RSA باستخدام المصفوفات M1، M2، M3 المذكورة بالمسألة 3-4.

13-3 هل يكافئ الأسلوب المبين فيما يلي خوارزمية RSA؟ وضح سبب الموافقة أو الرفض.

a. اختر عدداً فردياً E.

b. اختر عددين أوليين P، Q بحيث يكون ناتج قسمة $(P\text{-}1)(Q\text{-}1)$-1 على E يساوي عدداً زوجياً.

c. اضرب P، Q للحصول على N.

d. احسب $D = [(P\text{-}1)(Q\text{-}1)(E\text{-}1)+1]/E$.

14-3 افترض استخدام RSA مع مفتاح معروف لبناء دالة تحوير أحادية الاتجاه. ثم قم بمعالجة رسالة مكوَّنة من سلسلة من الكتل كما يلي:

- قم بتشفير الكتلة الأولى.
- قم بعملية XOR للنتيجة السابقة مع الكتلة التالية ثم بعملية التشفير مرة أخرى.
- وهكذا.

وضح أن هذا الأسلوب ليس آمناً وذلك بحل المسألة الآتية:

خذ رسالة مكوَّنة من كتلتين B1، B2 وكود التحوير لها من

$$RSAH(B1, B2) = RSA(RSA(B1) \oplus B2)$$

خذ كتلة اعتباطية C1 ثم اختر C2 بحيث $RSAH(C1, C2) = RSAH(B1, B2)$.

ومن ثمَّ لا تحقق دالة التحوير تلك شرط المقاومة الضعيفة للتصادم.

15-3 افترض أن بوب (Bob) يستخدم نظام التشفير RSA بقيمة كبيرة جدًّا للمعامل n بحيث لا يمكن تحليلها إلى عواملها في مدة زمنية معقولة. افترض أن أليس (Alice) أرسلت رسالة إلى بوب وذلك بتمثيل كل حرف بعدد صحيح من 0 إلى 25

(A تُمثَّل بـ 0، B تُمثَّل بـ 1، ...، Z تُمثَّل بـ 25)؛ ثم تقوم بتشفير كل عدد بشكلٍ مستقل بخوارزمية RSA ذات قيم كبيرة لـ e، n. هل هذا الأسلوب آمن؟ إذا كانت الإجابة بلا، صف أكثر طرق الهجوم كفاءةً ضد هذا الأسلوب.

16-3 افترض أسلوب ديفي - هيلمان يستخدم قيمة أولية مشتركة q=1 وجذراً بدائياً α=2.

a. إذا كان المفتاح العام للمُستخدم A هو Y_A=9؛ فما قيمة المفتاح الخاص به X_A؟

b. إذا كان المفتاح العام للمُستخدم B هو Y_B=3؛ فما قيمة المفتاح السري المشترك K؟

الباب الثاني

تطبيقات أمن الشبكة

قمنا في هذا الباب بتناول طرق التشفير المختلفة واستخداماتها لتوفير السرية والتوثيق و تبادل المفاتيح والوظائف المتعلقة بها.

خريطة الطريق للباب الثاني

الفصل الرابع: تطبيقات التوثيق

يتعرض الفصل الرابع لاثنين من أهم المواصفات المعيارية للتوثيق والأكثر انتشاراً حالياً. الأولى تتمثل في بروتوكول كيربيروس والمعتمد على التشفير التقليدي والمستخدم في كثير من النظم. والثانية تتمثل في خوارزمية X.509 لتعريف صيغة قياسية للشهادات مما يمكن المستخدمين من الحصول على شهادات للمفاتيح العامة للتحقق منها.

الفصل الخامس: أمن البريد الإلكتروني

يعدّ البريد الإلكتروني التطبيق الموزّع الأكثر استخداماً ويزداد الطلب لتوفير خدمات التوثيق والسرية كمكوّن أساسي لهذا التطبيق. نتناول في الفصل الخامس طريقتين يحتمل أن يسودا لأمن البريد الإلكتروني في المستقبل القريب؛ وهما PGP وS/MIME.

الفصل السادس: أمن بروتوكول الإنترنت

يعدّ الأمن في مستوى بروتوكول الإنترنت (IP) مهماً جداً في تصميم نظم الأمن بين الشبكات. ولذا يتناول الفصل السادس أمن بروتوكول الإنترنت والذي طوّر للاستخدام في الشبكات الحالية والمستقبلية.

الفصل السابع: أمن شبكة الويب

يركز الفصل السابع على أمن الويب نظراً للنمو المتزايد لاستخدامات الويب وخصوصاً في التجارة الإلكترونية.

الفصل الثامن: أمن إدارة الشبكات

يركز الفصل الثامن على الإمكانات الأمنية الأوَّليَّة المتاحة في الإصدار الأول لبروتوكول إدارة الشبكات الأكثر انتشاراً والمعروف ببروتوكول إدارة الشبكات البسيط ((Simple Network Management Protocol (SNMP)، ثم يناقش السمات الأمنية الأكثر تطوراً المتوفّرة في الإصدار الثالث.

4

الفصل الرابع

تطبيقات التوثيق

محتويات الفصل:

4-1 نظام كيربيروس (Kerberos)

4-2 خدمة X.509 للتوثيق

4-3 البنية التحتية للمفاتيح العامة

4-4 توصيات للمطالعة

4-5 مصادر للمعلومات على الويب

4-6 مصطلحات رئيسة

4-7 أسئلة للمراجعة ومسائل

الملحق A-4: طرق التشفير في نظام كيربيروس

"لا يمكننا الدخول في تحالف مع الأمراء المجاورين حتى نتعرف على مخططاتهم"
– من كتاب فنّ الحرب، لصن تزو.

النقاط الرئيسة

- يُعدُّ كيربيروس (Kerberos) خدمة توثيق مصُمَّمة للاستخدام في البيئات الموزَّعة.
- يَستخدم كيربيروس خدمة التوثيق عن طريق طرف ثالث مؤتمن لإنشاء اتصالات موثقة بين برامج الزبائن والأنظمة الخادمة.
- يُعرِّف X.509 صيغة شهادات المفاتيح العامة. وتُستخدَم تلك الصيغة على نطاق واسع في مجموعة متنوعة من التطبيقات.
- تُعرَّف البنية التحتية للمفاتيح العامة (PKI) بأنها مجموعة الأجهزة والبرامج والأشخاص والسياسات والإجراءات اللازمة لإنشاء شهادات رقمية مبنيّة على التشفير غير المتماثل (التشفير بالمفاتيح العامة) وإدارتها وتخزينها وتوزيعها وإلغائها.
- في العادة تستخدم تطبيقات PKI شهادات X.509.

يتناول هذا الفصل بعض عمليات التوثيق التي طُوِّرت لدعم توثيق طبقة التطبيقات وتوقيعاتها الرقمية. سنبدأ بتناول واحدة من أقدم الخدمات وأوسعها انتشاراً تُعرَف بكيربيروس (Kerberos). بعدها سنقوم بفحص خدمة توثيق الأدلة X.509. تكمن أهمية هذا المعيار في كونه جزءاً من خدمة الدليل التي يدعمها، وهو أيضاً عنصر أساسي يُستخدم في معايير أخرى (مثل S/MIME) والتي سنناقشها في الفصل الخامس. وأخيراً يتناول هذا الفصل فحص مفهوم البنية التحتية للمفاتيح العامة (PKI).

1-4 نظام كيربيروس

نظام كيربيروس (Kerberos **أو** Cerberus)[1] هو خدمة توثيق طُوِّرت كجزء من مشروع أثينا (Project Athena) في معهد MIT بالولايات المتحدة. ويعالج هذا البروتوكول المشكلة الآتية: بافتراض بيئة مُوزَّعة ومفتوحة يريد فيها مستخدمو محطات العمل الفرعية الاستفادة من الخدمات المتاحة على الخادمات الموزَّعة خلال الشبكة، المطلوب أن تكون الأنظمة الخادمة قادرة على قصر الوصول على المستخدمين المخوّلين فقط، وأن تكون قادرة على التحقق من صحة الطلبات المقدمة للخدمات المختلفة. في مثل هذه البيئة، لا يمكن الثقة بمحطات العمل الفرعية لتعريف هوية مستخدميها لخدمات الشبكة بشكلٍ صحيح. وعلى وجه الخصوص توجد التهديدات الثلاثة الآتية:

- قد يتمكّن مستخدم من الدخول إلى محطة عمل فرعية معينة – بشكلٍ غير مُصرَّح به – ثم يدّعي أنه مستخدم آخر لمحطة العمل الفرعية تلك.
- يمكن تغيير عنوان الشبكة لمحطة عمل فرعية حتى تظهر طلبات الخدمة من تلك المحطة كما لو كانت صادرة من محطة عمل أخرى.
- يمكن أن يتنصت مستخدم على المراسلات ويستخدم هجوم إعادة الإرسال للوصول لخدمة ما أو لعرقلة عمل الخادم.

في أيٍّ من تلك الحالات، قد يتمكن مستخدمٌ ما من الحصول على خدمات وبيانات ليس مخوّلاً له الحصول عليها. وللدفاع ضد ذلك، يقوم كيربيروس – بدلاً من بناء بروتوكولات توثيق تفصيلية في كل خادم – بتوفير خدمة توثيق مركزية لإجراء عمليات التحقّق من الهوية (إثبات هوية المستخدمين للخادمات وهوية الخادمات للمستخدمين). وبخلاف أكثر أساليب التوثيق الأخرى الموصوفة في هذا الكتاب،

---

[1] طبقاً لقاموس موضوعات ورموز في الفن والمنشور عام 1979 "هو كلب في الأساطير اليونانية متعدد الرؤوس (عادةً ثلاثة رؤوس) وقد يكون له ذيل ثعبان يقوم بحراسة بوابة العالم الآخر"، ويطلق عليه أيضاً كلب الجحيم. يشبه نظام كيربيروس الحديث هذه الأسطورة في أنه كان من المفترض أن يكون له ثلاثة مكوِّنات لحراسة بوابة الشبكة: التوثيق، والمراقبة، والمحاسبة. لكن لم يتم تحقيق المكوِّنين الأخيرين.

يعتمد كيربيروس حصرياً على التشفير المتماثل دون أي استخدام للتشفير بالمفاتيح العامة.

هناك إصداران من كيربيروس شائعا الاستخدام؛ الإصدار 4 [MILL88, STEI88] والذي لا تزال بعض تطبيقاته موجودة والإصدار 5 [KOHL94] والذي يُصحِّح بعض أوجه القصور في أمن الإصدار 4، وتم صدوره كمعيار إنترنت مقترح (RFC 1510).[2] سنبدأ في هذا الجزء بمناقشة وجيزة للعوامل المُحفِّزة على استخدام كيربيروس. لكن بعد ذلك، ونظراً لتعقيدات كيربيروس، من الأفضل البدء بوصف لبروتوكول التوثيق المُستخدم في الإصدار 4. سيُمكننا هذا من رؤية جوهر استراتيجية كيربيروس بدون الانشغال ببعض التفاصيل المطلوبة لمعالجة تهديدات الأمن الدقيقة. وأخيراً سنقوم بفحص الإصدار 5.

4-1-1 العوامل المُحفِّزة

إذا خُصِّصت لمجموعة من المستخدمين أجهزة حاسبات شخصية غير متصلة بأي شبكة، فعندئذ يمكن حماية ملفات كل مستخدم وموارده وذلك بالحماية المادية للحاسب الشخصي المُخصَّص له. لكن عندما يتم خدمة هؤلاء المستخدمين من قِبَل نظام مركزي يعمل بطريقة مشاركة الوقت، فإن نظام التشغيل يناطُ به توفير الأمن للمستخدمين. فيمكن لنظام التشغيل فرض سياسات للتحكم في الوصول بناءً على هوية المستخدم واستخدام إجراءات دخول محددة لتعريف مستخدمي النظام.

اليوم لا ينتشر استخدام أيٍّ من تلك السيناريوهات (سواء استخدام حاسبات مخصصة أو استخدام نظام مركزي بمشاركة الوقت)، وإنما يكثر استخدام بنية موزّعة مؤلَّفة من محطات عمل فرعية مخصصة للمستخدمين (الزبائن) وخادمات موزَّعة أو مركزية. في مثل هذه البيئة يمكن تصور ثلاث طرق للأمن:

---

[2] النسخ من 1 إلى 3 كانت نسخاً للتطوير الداخلي، لكن النسخة 4 كانت الإصدار الرئيس لـ Kerberos.

1. الاعتماد على كل محطة عمل فرعية للتأكد من هوية مستخدمها (أو مستخدميها) والاعتماد على كل خادم لفرض سياسة أمن طبقاً لهوية المستخدم (user ID).
2. الطلب من أنظمة الزبائن أن تثبت (تُوثّق) هويتها للخادمات، في حين نثق بتلك الأنظمة فيما يتعلق بهوية مستخدميها.
3. الطلب من المستخدم إثبات هويته لكل خدمة يستدعيها، والطلب أيضاً من الخادمات أن تثبت هويتها للزبائن.

في بيئة صغيرة ومغلقة تملك كل الأنظمة فيها وتقوم بتشغيلها هيئة واحدة، قد تكون الطريقة الأولى وحدها أو ربما الثانية كافية.[3] لكن في بيئة أكثر انفتاحاً ولها اتصالات بأجهزة أخرى من خلال شبكة، نحتاج إلى الطريقة الثالثة لحماية معلومات وموارد المستخدمين المُخزَّنة بالخادم. يدعم كيربيروس الطريقة الثالثة، ويفترض بنية زبون/خادم موزَّعة، كما يستخدم واحداً أو أكثر من الخادمات الخاصة لتوفير خدمة التوثيق.

قام أول تقرير تم نشره عن كيربيروس [STEI88] بعرض المتطلبات الآتية:

- الأمن: يجب ألا يتمكن متنصّت على الشبكة من الحصول على المعلومات اللازمة لانتحال هوية مستخدم آخر. وبوجه أعم، يجب أن يكون كيربيروس قويًّا بما فيه الكفاية حتى لا يجده خَصْم محتمل الوصلة الضعيفة للنفاذ للشبكة.
- الاعتمادية: بالنسبة لكافة الخدمات التي تعتمد على كيربيروس للتحكم في الوصول، فإن عدم توفر خدمة كيربيروس يعني عدم توفر الخدمات المدعومة. ولذا يجب أن يتمتع كيربيروس باعتمادية عالية، كما يجب أن يستخدم بنية خادمات موزَّعة يكون كل نظام فيها داعماً احتياطياً للآخر.

[3] حتي تلك البيئات المغلقة معرضة لخطر الهجوم من قِبَل موظف ساخط.

- الشفافية: في صورتها المثالية؛ تعني ألا يشعر المستخدم بعمليات التوثيق الجارية إلا عند قيامه بإدخال كلمة السر.
- قابلية التوسع: يجب أن يكون النظام قادراً على دعم أعداد كبيرة من الزبائن والخادمات، ويمكن ذلك باستخدام بنية موزَّعة مؤلَّفة من وحدات.

لدعم هذه المتطلبات، يتضمن مخطط كيربيروس العام خدمة التوثيق عن طريق طرف ثالث مؤتمن تستخدم بروتوكولاً مبنياً على مقترح نيدام وشرودر (Needham & Schroeder) [NEED78]. ونعني بكونه مؤتمناً أن الزبائن والخادمات تثق بكيربيروس في إدارة عمليات التوثيق المتبادل بينهم. بافتراض أن بروتوكول كيربيروس مُصمَّمٌ بشكلٍ جيد، فعندئذ تكون خدمة التوثيق آمنة إذا كان خادم كيربيروس نفسه آمناً.[4]

4-1-2 الإصدار الرابع لكيربيروس

يستخدم إصدار كيربيروس الرابع خوارزمية DES في بروتوكولٍ مُحكمٍ نوعاً ما لتوفير خدمة التوثيق. بالنظر إلى البروتوكول ككل، سيكون من الصعب إدراك الحاجة لكثير من العناصر المُتضمَّنة به. لذا سنتبنّى استراتيجية استخدمها بيل براينت في مشروع أثينا [BRYA 88] وسنطورها حتى نحصل على البروتوكول بكامله. سنبدأ أولاً بالنظر في سلسلة حوارات افتراضية، بحيث يضيف كل حوار درجة من التعقيد لمواجهة نقاط الضعف الأمنية التي تُكتشَف في الحوار السابق. بعد فحص البروتوكول، سننظر إلى بعض السمات الأخرى للإصدار الرابع.

❖ حوار توثيق بسيط:

في بيئة شبكة غير محمية، يمكن لأي زبون طلب خدمة من أي خادم. الخطورة الأمنية الواضحة هنا هي "انتحال الهوية". فيمكن لخَصْم ما أن يدّعي أنه

---

[4] لا ينبغي افتراض أمن خادم كيربيروس بشكلٍ تلقائي، وإنما يجب حراسته بعناية (على سبيل المثال وضعه في غرفة محكمة الإغلاق).

زبون آخر ومن ثَمَّ يحصل على امتيازات من الخادم غير مخوَّلة له. لمواجهة هذا التهديد، يجب أن يكون الخادم قادراً على التحقق من هويّة الزبائن طالبي الخدمة. يمكن أن يقوم الخادم بتلك المهمّة لكل تفاعل بين الزبون والخادم. لكن في بيئة مفتوحة سيضع هذا الأمر عبئاً كبيراً على كل خادم. كبديل لذلك يمكن استخدام خادم توثيق (Authentication Server (AS)) على علمٍ بكلمات السر لكل المستخدمين والمحفوظة لديه في قاعدة بيانات مركزية. بالإضافة لذلك يشترك AS مع كل خادم في مفتاحٍ سريٍ فريدٍ بينهما. توزع هذه المفاتيح بشكلٍ مادي أو بأي أسلوب آمن آخر. لنأخذ في عين الاعتبار الحوار الافتراضي الآتي:[5]

$$\begin{aligned}
&\textbf{(1)}\ \mathrm{C} \rightarrow \mathrm{AS}: && ID_C \| P_C \| ID_V \\
&\textbf{(2)}\ \mathrm{AS} \rightarrow \mathrm{C}: && Ticket \\
&\textbf{(3)}\ \mathrm{C} \rightarrow \mathrm{V}: && ID_C \| Ticket \\
&Ticket = \mathrm{E}(K_v, [ID_C \| AD_C \| ID_V])
\end{aligned}$$

حيث:

C: الزبون.

AS: خادم التوثيق.

V: خادم.

ID_V: معرِّف (هوية) الخادم V.

ID_C: معرِّف (هوية) مستخدم على نظام الزبون C.

P_C: كلمة السر لمستخدم على نظام الزبون C.

AD_C: عنوان الشبكة لـ C.

K_v: مفتاح تشفير سري مشترك بين AS وV.

في هذا السيناريو، يسجل المستخدم دخولاً على محطة عمل ويطلب الوصول إلى الخادم V. تطلب وحدة الزبون C بمحطة العمل تلك كلمة السر من المستخدم ثم ترسل رسالة إلى AS تتضمّن معرِّف (هوية) المستخدم ومعرِّف (هوية) الخادم وكلمة

[5] يشير الجزء على يسار النقطتين إلى المُرسِل والمُستلِم، ويشير الجزء على يمين النقطتين إلى محتويات الرسالة. أما الرمز || فيعني الوصل.

السر للمستخدم. يقوم AS بفحص قاعدة البيانات لديه ليرى ما إذا كانت كلمة السر التي أدخلها المستخدم صحيحة، وإذا ما كان هذا المستخدم مسموحاً له بالوصول للخادم V. في حالة اجتياز كلا الاختبارين بنجاح، يكون AS قد تحقق من هوية المستخدم وعليه أن يقنع الخادم V بأن ذلك المستخدم موثوق به. ولتحقيق ذلك، يُنشئ AS تذكرة (ticket) تتضمن معرِّف المستخدم وعنوان الشبكة ومعرِّف الخادم. تُشفَّر تلك التذكرة باستخدام المفتاح السري المشترك بين AS والخادم V، ثم يُعاد إرسالها إلى C. ونظراً لكون التذكرة مُشفَّرة فلا يمكن تعديلها لا من قِبَل C ولا من قِبَل أي خَصْم.

يمكن الآن أن يتقدم C بهذه التذكرة إلى V لطلب الخدمة، وذلك بإرسال رسالة إلى V تحتوي هويته بالإضافة إلى التذكرة. يزيل V تشفير التذكرة ويتحقق من أن هوية المستخدم في التذكرة هي نفسها هوية المستخدم غير المُشفَّرة في الرسالة. عند تطابق الهويتين، يعتبر الخادم أن هذا المستخدم موثوق به ويمنحه الخدمة المطلوبة.

يُعدُّ كل مكوِّن من مكوِّنات الرسالة في الخطوة (3) مهماً. تُشفَّر التذكرة لمنع تعديلها أو تزييفها. تُضمَّن هوية الخادم ID_V في التذكرة لكي يتحقق الخادم من أن إزالة تشفير التذكرة قد تم بشكلٍ صحيح. تحتوي التذكرة أيضاً على ID_C للإشارة إلى أن هذه التذكرة أُصدِرت نيابةً عن C. وأخيراً يساعد AD_C على مواجهة التهديد الآتي: يمكن أن يلتقط خَصْم التذكرة المُرسَلة في الرسالة (2)، ثم يستخدم الاسم ID_C ويرسل رسالة بالصيغة (3) من محطة عمل أخرى. في هذه الحالة يكون الخادم قد تلقى تذكرةً صحيحةً تتطابق مع هوية المستخدم ومن ثَمَّ يسمح له – خطأً – بالوصول إلى الخدمة. لمنع مثل هذا الهجوم، يُضمِّن AS في التذكرة عنوان الشبكة الذي جاء منه الطلب الأصلي. ومن ثَمَّ تكون التذكرة صحيحة فقط إذا كانت مُرسَلةً من نفس محطة العمل التي طلبت التذكرة في البداية.

❖ حوار توثيق أكثر أمناً:

بالرغم من أن السيناريو السابق يحل بعض مشاكل التوثيق في بيئة شبكة مفتوحة، تبقى هناك بعض المشكلات التي يبرز منها مشكلتان بشكلٍ خاص. أولاً: نريد تقليل عدد مرات إدخال كلمة السر من المستخدم. بافترض أن كل تذكرة يمكن استخدامها لمرّة واحدة، إذا سجل المستخدم C دخولاً على محطة عمل فرعية في الصباح ورغب في الاطلاع على بريده الإلكتروني، فعليه أن يُدخِل كلمة السر للحصول على تذكرة تُرسَل لخادم البريد. كما أن عليه أن يقوم بإدخال كلمة السر في كل مرة يريد فيها الاطلاع على بريده الإلكتروني أثناء اليوم. يمكن تحسين ذلك الأمر باستخدام تذاكر قابلة لإعادة الاستخدام. وذلك بأن تقوم محطة العمل بتخزين تذكرة خادم البريد التي تتلقاها ثم تستخدمها نيابةً عن المستخدم للاتصال بخادم البريد في كل مرة يرغب المستخدم فيها الاتصال بالخادم أثناء نفس الجلسة.

لكن تبقى مشكلة في هذا الأسلوب وهي أن المستخدم يحتاج تذكرة جديدة لكل خدمة مختلفة. فمثلاً إذا رغب المستخدم الدخول لخادم الطباعة أو خادم البريد أو خادم الملفات أو غير ذلك، فإنه يحتاج تذكرة جديدة في أول مرة لكل وصول مما يتطلب إعادة إدخال كلمة السر. هناك مشكلة ثانية هي أن السيناريو السابق يتضمّن إرسال كلمة السر غير مُشفَّرة (الرسالة (1)). ولذا يمكن أن يلتقط متنصّت كلمة السر وبسهولة يستخدم أي خدمة متاحة للضحيّة.

لحل تلك المشكلات الإضافية، سنقدّم أسلوباً آخر لتجنب استخدام كلمات السر غير المُشفَّرة، لكن سنحتاج لخادم جديد يُعرَف بخادم منح التذاكر (Ticket-Granting Server (TGS)). السيناريو الجديد والذي ما زال افتراضياً يكون كما يأتي:

مرة واحدة لكل جلسة اتّصال للمستخدم:

(1) C → AS: $ID_C \| ID_{tgs}$
(2) AS → C: $E(K_c, Ticket_{tgs})$

مرة واحدة لكل نوع من الخدمة:

(3) C → TGS: $ID_C \| ID_V \| Ticket_{tgs}$
(4) TGS → C: $Ticket_v$

مرة واحدة لكل جلسة خدمة:

(5) C → V: $ID_C \| Ticket_v$

وهكذا يكون:

$$Ticket_{tgs} = E(K_{tgs}, [ID_C \| AD_C \| ID_{tgs} \| TS_1 \| Lifetime_1])$$
$$Ticket_v = E(K_v, [ID_C \| AD_C \| ID_v \| TS_2 \| Lifetime_2])$$

تمنح الخدمة الجديدة (TGS) تذاكر للمستخدمين الذين تم التحقق من هويتهم عن طريق الخادم AS. وهكذا يطلب المستخدم في البداية من AS تذكرة للحصول على الخدمة من خادم منح التذاكر ($Ticket_{tgs}$) سنطلق عليها من الآن ولاحقاً تذكرة منح التذاكر. تقوم وحدة الزبون في محطة العمل للمستخدم بتخزين تلك التذكرة. وفي كل مرّة يحتاج المستخدم الوصول إلى خدمة جديدة، تُقدِّم وحدة الزبون تلك التذكرة إلى خادم TGS للتوثيق. ومن ثم يقوم خادم TGS بمنح المستخدم تذكرة للخدمة المطلوبة. تُخزِّن وحدة الزبون كل تذكرة تحصل عليها لمنح خدمة معينة وتستخدمها في كل مرة لتوثيق المستخدم للخادم الموفِّر لتلك الخدمة. دعنا ننظر لتفاصيل هذه الطريقة:

1. تطلب وحدة الزبون تذكرة منح التذاكر نيابةً عن المستخدم بإرسال هوية المستخدم وكلمة السر له وهوية خادم TGS إلى AS مشيرةً إلى الحاجة لاستخدام خدمة TGS.
2. يردّ AS بتذكرة مُشفَّرة بمفتاح مشتق من كلمة سر المستخدم. عندما يصل هذا الردّ إلى وحدة الزبون، تطلب من المستخدم إدخال كلمة السر، وتُولِّد المفتاح الذي يُستخدَم في إزالة تشفير الرسالة المتلقاة. إذا كانت كلمة السرّ صحيحة، يمكن عندها استخلاص التذكرة بنجاح.

ولأنه من المفترض أن المستخدم الحقيقي فقط هو الذي يعرف كلمة السر، فهو فقط الذي يمكنه استخلاص التذكرة. وهكذا نكون قد استخدمنا كلمة السر للحصول على شهادة من كيربيروس دون الحاجة إلى إرسال كلمة السر بدون

تشفير. تتضمّن التذكرة نفسها عنوان الشبكة وهوية المستخدم وهوية خادم TGS. يمثل هذا السيناريو الأول. تكمن الفكرة في أن الزبون يمكن أن يستخدم هذه التذكرة لطلب عدة تذاكر لمنح خدمات متعددة. لذا يجب أن تكون تذكرة منح التذاكر قابلةً لإعادة الاستخدام. لذا لا ينبغي أن يتمكن الخَصْم من التقاط التذكرة واستخدامها. خذ بعين الاعتبار السيناريو الآتي:

يلتقط خَصْم تذكرة الدخول وينتظر حتى يُسجِّل المستخدم الخروج من محطة العمل. عندئذ يمكن للخَصْم تسجيل الدخول لمحطة العمل الفرعية تلك أو تهيئة محطة العمل الفرعية الخاصة به ليكون لها نفس عنوان الشبكة لمحطة العمل الفرعية الضحيّة. هكذا يتمكن الخَصْم من إعادة استخدام التذكرة للاحتيال على خادم TGS. لمواجهة تلك المشكلة يتم تضمين خاتم الوقت بالتذكرة للإشارة إلى تاريخ إصدار التذكرة ووقتها، وتضمين فترة الصلاحية للإشارة إلى المدة التي تكون فيها التذكرة سارية المفعول (على سبيل المثال ثماني ساعات).

وهكذا يصبح لدى الزبون تذكرة قابلة لإعادة الاستخدام ولا يحتاج لمضايقة المستخدم بتكرار إدخال كلمة السر مع كل طلب خدمة جديد. أخيراً، لاحظ أن تذكرة منح التذاكر تكون مُشفَّرةً بمفتاح سري معروفٌ فقط لخادم AS وخادم TGS مما يؤدي إلى منع إمكانية تعديل التذكرة. ويعاد تشفير التذكرة مرة ثانية بمفتاح يعتمد على كلمة السر الخاصة بالمستخدم مما يضمن إمكانية استرجاع التذكرة من قِبَل المستخدم الحقيقي فقط، ومن ثَمَّ توفير التوثيق.

بامتلاك الزبون تذكرة منح التذاكر، يمكنه أن يتصل بأي خادم من خلال الخطوات 3 و4 الآتية:

3. يطلب الزبون تذكرة منح خدمة نيابةً عن المستخدم. ولهذا الغرض، يرسل الزبون رسالة إلى خادم TGS تتضمن هوية المستخدم وهوية الخدمة المطلوبة وتذكرة منح التذاكر.

4. يقوم خادم TGS بإزالة تشفير التذكرة التي استلمها والتحقق من نجاح إزالة التشفير بحصوله على هويته من الرسالة. ثم يقوم بالتأكد من عدم انتهاء فترة الصلاحية، ويقارن هوية المستخدم وعنوان الشبكة بالمعلومات الواردة للتحقق من هوية المستخدم. إذا كان المستخدم مُرخَّصاً له بالوصول إلى الخادم V، يصدر خادم TGS تذكرة تسمح لذلك المستخدم بالوصول للخدمة المطلوبة.

لتذكرة منح الخدمة نفس صيغة تذكرة منح التذاكر. واقعياً ولكون TGS خادماً، نتوقّع أن تكون العناصر المطلوبة لتوثيق الزبون لدى خادم TGS هي نفسها المطلوبة لتوثيق الزبون لدى خادم أي تطبيق. وكذلك، تحتوي تلك التذكرة على خاتم الوقت وفترة الصلاحية. فإذا أراد المستخدم الوصول إلى نفس الخدمة في وقتٍ لاحقٍ، يمكن للزبون ببساطة استخدام تذكرة منح الخدمة التي حصل عليها سابقاً ولا يحتاج لأن يضايق المستخدم بإعادة إدخال كلمة السر. لاحظ أن التذكرة مُشفَّرة بمفتاح سري (K_v) معروف فقط لخادم TGS وخادم التطبيق، مما يمنع إجراء أي تعديل عليها.

أخيراً، بامتلاك تذكرة منح خدمة معيّنة، يتمكّن الزبون من الاستفادة من تلك الخدمة بالخطوة 5:

5. يطلب الزبون الوصول للخدمة نيابةً عن المستخدم. ولهذا الغرض، يُرسِل الزبون رسالة إلى الخادم تتضمن هوية المستخدم وتذكرة منح الخدمة. يتحقّق الخادم من الهوية باستخدام محتويات التذكرة.

يحقق هذا السيناريو الجديد المتطلبين التاليين: الاستفسار عن كلمة السر مرة واحدة فقط لكل جلسة للمستخدم، وحماية كلمة السر.

❖ الإصدار 4 لحوار التوثيق:

رغم أن السيناريو السابق يُحسِّن الأمن مقارنةً بالمحاولة الأولى، تبقى هناك مشكلتان إضافيتان. لب المشكلة الأولى هو فترة الصلاحية الخاصة بتذكرة منح

التذاكر. إذا كانت تلك الفترة قصيرة جداً (مثلاً عدة دقائق)، فسيُسأل المستخدم مراراً وتكراراً عن كلمة السر. أما إذا كانت فترة الصلاحية طويلة (مثلاً عدة ساعات)، فسيتاح للخَصْم فرصة أكبر لهجوم إعادة الإرسال (replay attack). يمكن أن يتنصت الخَصْم على الشبكة، ويلتقط نسخة من تذكرة منح التذاكر، ثم ينتظر أن يسجل المستخدم الشرعي الخروج، ويقوم بانتحال عنوان الشبكة للمستخدم الشرعي وإرسال رسالة بالخطوة (3) إلى خادم TGS. وبذلك سيتمكَّن الخَصْم من الوصول للموارد والملفات المتاحة للمستخدم الشرعي لعدد غير محدود من المرات.

بنفس الطريقة، إذا التقط الخَصْم تذكرة منح خدمة واستخدمها قبل أن تنتهي صلاحيتها، فسيتمكن من الحصول على تلك الخدمة. ولذا نحتاج إلى متطلب إضافي أن تكون خدمة الشبكة (خادم TGS أو خادم التطبيق) قادرة على إثبات أن الشخص المُستخدِم للتذكرة هو نفسه الشخص الذي صدرت له تلك التذكرة.

المشكلة الثانية هي أنه قد يكون هناك متطلب بأن تُوثِّق الخادمات نفسها للمستخدمين. بدون مثل هذا التوثيّق، يمكن أن يُخرِّب الخَصْم التهيئة بحيث يتم تحويل أي رسائل موجهة للخادم إلى موقع آخر. بهذا يكون الخادم الخطأ بوضعٍ يمكنه التصرف كالخادم الحقيقي والحصول على أي معلومات من المستخدم ثم يقوم بعدها بحجب الخدمة الحقيقية عن المستخدم. سنقوم بفحص هذه المشاكل تباعاً وسنشير إلى الجدول 4-1 الذي يبيّن بروتوكول كيربيروس الفعلي.

أولاً: خذ بعين الاعتبار مشكلة التقاط تذاكر منح التذاكر والحاجة إلى التأكد من أن مُقدِّم التذكرة (ticket presenter) هو نفسه الزبون الذي أُصدِّرت له التذكرة. يكمن الخطر في ذلك في أن الخَصْم قد يسرق التذكرة ويستخدمها قبل أن تنتهي مدة صلاحيتها. لتفادي هذه المشكلة، يقوم AS بتزويد الزبون وخادم TGS بمعلومة سرية بطريقة آمنة. يمكن أن يثبت الزبون هويته لخادم TGS بكشف تلك المعلومة السرية (مرة أخرى بطريقة آمنة). يمكن إنجاز ذلك بطريقة فعالة عن طريق

استخدام مفتاح التشفير كالمعلومة السرية؛ يُعرَف هذا في كيربيروس باسم مفتاح الجلسة.

يبيّن الجدول 4-1(a) طريقة لتوزيع مفتاح الجلسة. كما سبق، يُرسِل الزبون رسالة إلى AS يطلب فيها الاتصال بخادم TGS. يرد AS برسالة مُشفَّرة بمفتاح مشتق من كلمة السر للمستخدم (K_c) تتضمن التذكرة. كما تتضمن الرسالة المُشفَّرة أيضاً نسخة من مفتاح الجلسة $K_{c,tgs}$ بين C وTGS. ونظراً لوجود مفتاح الجلسة هذا بداخل الرسالة المُشفَّرة بالمفتاح K_c، فيمكن أن يقرأها زبون المستخدم فقط. كما يتم أيضاً تضمين نفس مفتاح الجلسة في التذكرة والتي يمكن قراءتها فقط من قِبَل TGS. ومن ثَمَّ يكون قد تم تسليم مفتاح الجلسة بشكلٍ آمن لكلٍّ من C وTGS.

الجدول 4-1: ملخص للرسائل المتبادلة في الإصدار 4 لـ كيربيروس.

(1) C → AS $ID_c \| ID_{tgs} \| TS_1$
(2) AS → C $E(K_c, [K_{c,tgs} \| ID_{tgs} \| TS_2 \| Lifetime_2 \| Ticket_{tgs}])$
$Ticket_{tgs} = E(K_{tgs}, [K_{c,tgs} \| ID_C \| AD_C \| ID_{tgs} \| TS_2 \| Lifetime_2])$

(a) تبادل خدمة توثيق للحصول على تذكرة منح تذكرة

(3) C → TGS $ID_v \| Ticket_{tgs} \| Authenticator_c$
(4) TGS → C $E(K_{c,tgs}, [K_{c,v} \| ID_v \| TS_4 \| Ticket_v])$
$Ticket_{tgs} = E(K_{tgs}, [K_{c,tgs} \| ID_C \| AD_C \| ID_{tgs} \| TS_2 \| Lifetime_2])$
$Ticket_v = E(K_v, [K_{c,v} \| ID_C \| AD_C \| ID_v \| TS_4 \| Lifetime_4])$
$Authenticator_c = E(K_{c,tgs}, [ID_C \| AD_C \| TS_3])$

(b) تبادل خدمة منح تذكرة للحصول على تذكرة منح خدمة

(5) C → V $Ticket_v \| Authenticator_c$
(6) V → C $E(K_{c,v}, [TS_5 + 1])$ (for mutual authentication)
$Ticket_v = E(K_v, [K_{c,v} \| ID_C \| AD_C \| ID_v \| TS_4 \| Lifetime_4])$
$Authenticator_c = E(K_{c,v}, [ID_C \| AD_C \| TS_5])$

(c) تبادل توثيق بين الخادم والزبون للحصول على خدمة

لاحظ أنه قد تم إضافة عدة معلومات إضافية للمرحلة الأولى للحوار. تتضمّن الرسالة (1) خاتم الوقت لكي يتحقق AS من صلاحيتها، وتتضمّن الرسالة (2) عدة عناصر من التذكرة في صيغة مُتاحة لـ C. يُمكِّن ذلك C من التأكد من أن تلك التذكرة لـ TGS ومن معرفة وقت انتهاء صلاحيتها.

بمعرفة التذكرة ومفتاح الجلسة، يكون C جاهزاً لمخاطبة TGS. كما في السابق، يُرسِل C رسالة إلى TGS تتضمّن التذكرة وهوية الخدمة المطلوبة (الرسالة (3) في الجدول 4-1(b)). كما يُرسِل C أيضاً مُوثِّقاً (authenticator) يتضمّن هوية وعنوان مستخدم C وخاتم الوقت. على خلاف التذكرة – التي يمكن إعادة استخدامها – يُفترض في المُوثِّق أن يُستخدم مرة واحدة فقط وتكون فترة صلاحيته قصيرةً جداً. يمكن أن يزيل TGS تشفير التذكرة بالمفتاح الذي يشترك فيه مع AS. تشير هذه التذكرة إلى أن مستخدم C قد زُوِّد بمفتاح الجلسة. عملياً تنص التذكرة على أن: "أي شخص يستخدم $K_{c,tgs}$ يجب أن يكون C". يستخدم TGS مفتاح الجلسة ليزيل تشفير المُوثِّق. عندها يقوم TGS بفحص الاسم والعنوان الموجودين بالمُوثِّق ومقارنتهما بماورد بالتذكرة وعنوان الشبكة في الرسالة الواردة. إذا حدث تطابق، يتأكد TGS من أن مُرسِل التذكرة هو حقاً مالك التذكرة الحقيقي. عملياً، يقول المُوثِّق: "عند اللحظة TS_3، ها أنا استخدم $K_{c,tgs}$". لاحظ أن التذكرة لا تثبت هوية أي شخص، لكنها وسيلة لتوزيع المفاتيح بشكلٍ آمن. أما المُوثِّق فهو من يحقق هوية الزبون. ولأن المُوثِّق يمكن أن يُستخدم فقط مرة واحدة وعمره قصير، يبطل خطر الخَصْم الذي يسرق التذكرة والمُوثِّق لاستخدامهما لاحقاً.

يتخذ رد TGS في الرسالة (4) نفس صيغة الرسالة (2). تُشفَّر الرسالة بمفتاح الجلسة المشترك بين TGS وC. تتضمّن الرسالة مفتاح جلسة مُشترك بين C والخادم V، وهوية الخادم V، وخاتم الوقت للتذكرة. ويتم تضمين نفس مفتاح الجلسة بالتذكرة نفسها.

الآن يكون لدى C تذكرة منح خدمة من الخادم V قابلة لإعادة الاستخدام. عندما يُقدِّم C هذه التذكرة (كما هو مبين بالرسالة (5))، يقوم أيضاً بإرسال

مُوثّق. يمكن للخادم أن يزيل تشفير التذكرة، ويستعيد مفتاح الجلسة، ويزيل تشفير المُوثّق. إذا لزم إجراء توثيق متبادل، يمكن أن يرد الخادم كما هو مبين بالرسالة (6) في الجدول 4-1. يُرجِع الخادم قيمة خاتم الوقت من المُوثّق زائد واحد بعد تشفيرها بمفتاح الجلسة. يمكن أن يزيل C تشفير تلك الرسالة لاستعادة قيمة خاتم الوقت بعد الزيادة. ولأن الرسالة مُشفَّرة بمفتاح الجلسة، يتأكد C من أنها قد أُنشِئت فقط بواسطة V. تؤكد محتويات الرسالة لـ C أنها ليست ردّاً قديماً أُعيد إرساله. وأخيراً في ختام هذه العملية، يكون الزبون والخادم قد اشتركا في مفتاح سري يمكن أن يُستخدَم لتشفير الرسائل في المُستقبَل بينهما أو لتبادل مفتاح جلسة عشوائي جديد لذلك الغرض. يُلخَّص الجدول 4-2 سبب وجود كل مكوِّن من مكوِّنات كيربيروس. ويعطي الشكل 4-1 لمحة مبسطة عن طريقة عمله.

❖ عوالم كيربيروس وأنظمته المتعددة:

توفير بيئة خدمة متكاملة لنظام كيربيروس (مكون من خادم كيربيروس، وعدد من الزبائن، وعدد من خادمات التطبيقات) يتطلب ما يأتي:

1. يجب أن تحتوي قاعدة بيانات خادم كيربيروس على هوية وكلمة سر كل مُستخدِم مُشترِك (أي أن كل المستخدمين مسجلون لدى خادم كيربيروس)
2. يجب أن يشترك خادم كيربيروس مع كل خادم في مفتاحٍ سرّيٍ (أي أن كل الخادمات مسجلة لدى خادم كيربيروس).

مثل هذه البيئة تُعرَف باسم عالَم كيربيروس (Kerberos realm). ويمكن توضيح مفهوم هذا العالم كما يأتي: يتكون عالَم كيربيروس من مجموعة العقد المُدارة والتي تشترك في نفس قاعدة بيانات كيربيروس الموجودة في نظام حاسب كيربيروس الرئيس، والذي يجب أن يُحتفظ به في غرفة آمنة مادياً. قد توجد نسخ للقراءة فقط من قاعدة بيانات كيربيروس على أنظمة حاسب كيربيروس أخرى. أما عند الحاجة لإجراء تغييرات في قاعدة البيانات فلا بد من إجرائها على نظام الحاسب الرئيس. يتطلب تغيير أو إدخال محتويات في قاعدة بيانات كيربيروس كلمة السر الرئيسة لكيربيروس. يتعلّق بهذا مفهوم "ممثل كيربيروس" (Kerberos

principal/actor)، والذي يشير إلى خدمة أو مستخدم معروف لنظام كيربيروس. يُميَّز كل ممثل كيربيروس باسمه والذي يشتمل على ثلاثة أجزاء: اسم الخدمة أو المستخدم، واسم الحالة، واسم العالم (عالَم كيربيروس).

في العادة تُشكِّل شبكات الزبائن والخادمات تحت الأنظمة الإدارية المختلفة عوالم مختلفة. أي أنه بشكلٍ عام ليس أمراً عملياً (أو لا يتوافق مع السياسة الإدارية) أن يكون المستخدمون والخادمات في مجال إداري معين مسجلين مع خادم كيربيروس في مكان آخر. لكن قد يحتاج مستخدمون في عالم ما إلى الوصول إلى خادمات في عوالم أخرى، وقد ترغب بعض الخادمات في تقديم خدمة لمستخدمين في عوالم أخرى بشرط إمكان التحقق من هوية هؤلاء المستخدمين.

الجدول 4-2: الأسباب المنطقية لمكوِّنات الإصدار 4 لبروتوكول كيربيروس.

(a) تبادل خدمة التوثيق

الرسالة (1)	الزبون يطلب تذكرة منح تذاكر
ID_C	تخبر AS عن هوية المستخدم لهذا الزبون
ID_{tgs}	تخبر AS أن المستخدم يطلب وصولاً لـ TGS
TS_1	تمكِّن AS من التحقق من أن ساعة الزبون متزامنة مع ساعة AS
الرسالة (2)	AS يُرجع تذكرة منح تذاكر
K_c	التشفير يعتمد على كلمة السر الخاصة بالمستخدم مما يُمكّن AS والزبون من التحقق من كلمة السر وحماية محتويات الرسالة (2)
$K_{c,\ tgs}$	نسخة من مفتاح الجلسة في متناول الزبون وتم إنشاؤه من قِبَل AS ليسمح بالتبادل الآمن بين الزبون وخادم TGS بدون أن يتطلب ذلك وجود مفتاح ثابت مشترك بينهما
ID_{tgs}	تؤكد أن هذه التذكرة هي لـ TGS
TS_2	تخبر الزبون عن وقت إصدار هذه التذكرة
$Lifetime_2$	تخبر الزبون عن فترة الصلاحية لهذه التذكرة
$Ticket_{tgs}$	التذكرة التي سيستخدمها الزبون للوصول لـ TGS

(b) تبادل خدمة منح التذاكر

الرسالة (3)	الزبون يطلب تذكرة منح خدمة
ID_V	تخبر TGS أن المستخدم يطلب الوصول للخادم V
$Ticket_{tgs}$	تؤكد لـ TGS أن هذا المستخدم تم توثيقه من قِبَل AS
$Authenticator_c$	ينشأ من قِبَل الزبون ليثبت صحة التذكرة
الرسالة (4)	TGS يُرجع تذكرة منح خدمة
$K_{c,\ tgs}$	مفتاح مشترك فقط بين C وTGS لحماية محتويات الرسالة (4)
$K_{c,\ v}$	نسخة من مفتاح الجلسة في متناول الزبون وتم إنشاؤه من قِبَل TGS ليسمح بالتبادل الآمن بين الزبون والخادم بدون أن يتطلب ذلك وجود مفتاح ثابت مشترك بينهما
ID_V	تؤكد أن هذه التذكرة هي للخادم V
TS_4	تخبر الزبون عن وقت إصدار هذه التذكرة
$Ticket_V$	التذكرة التي سيستخدمها الزبون للوصول لـ V
$Ticket_{tgs}$	قابلة لإعادة الاستخدام حتى لا يحتاج المستخدم أن يعيد إدخال كلمة السر
K_{tgs}	التذكرة مُشفَّرة بمفتاح معروف فقط لـ AS وTGS لمنع التلاعب
$K_{c,\ tgs}$	نسخة من مفتاح الجلسة في متناول TGS ويستخدم في إزالة تشفير المُوثِّق ومن ثَمَّ توثيق التذكرة
ID_C	تشير إلى مالك هذه التذكرة الحقيقي
AD_C	تمنع استخدام التذكرة من محطة عمل غير تلك التي طلبت التذكرة في البداية
ID_{tgs}	تؤكد للخادم أنه قام بإزالة التشفير بشكلٍ صحيح
TS_2	تخبر TGS عن وقت إصدار هذه التذكرة
$Lifetime_2$	تمنع إعادة الإرسال بعد انتهاء مدة التذكرة
$Authenticator_c$	تؤكد لـ TGS أن مُقدِّم التذكرة هو نفسه الزبون الذي صدرت له التذكرة؛ لها فترة عمر قصيرة جداً لمنع إعادة الإرسال
$K_{c,\ tgs}$	يُشفر المُوثِّق بمفتاح معروف فقط للزبون وTGS لمنع التلاعب
ID_C	يجب أن تطابق الهوية ID الموجودة بالتذكرة لتوثيق التذكرة
AD_C	يجب أن تطابق العنوان بالتذكرة لتوثيق التذكرة
TS_3	تخبر TGS عن وقت إنشاء المُوثِّق

(c) تبادل توثيق بين الخادم والزبون

الرسالة (5)	الزبون يطلب خدمة
$Ticket_V$	تؤكد للخادم أن هذا المستخدم تم توثيقه من قِبَل AS
$Authenticator_c$	ينشأ من قِبَل الزبون ليثبت صحة التذكرة
الرسالة (6)	تحقق اختياري للخادم من قِبَل الزبون
$K_{c,v}$	تؤكد لـ C أن الرسالة من V
TS_5+1	تؤكد لـ C أن هذا ليس إعادة إرسال لرد قديم
$Ticket_v$	قابلة لإعادة الاستخدام حتى لا يحتاج الزبون أن يعيد طلب تذكرة جديدة من TGS لكل وصول لنفس الخادم
K_v	التذكرة مُشفَّرة بمفتاح معروف فقط لـ TGS والخادم لمنع التلاعب
$K_{c,v}$	نسخة من مفتاح الجلسة في متناول الزبون وتستخدم في إزالة تشفير المُوثَّق ومن ثَمَّ توثيق التذكرة
ID_C	تشير إلى مالك هذه التذكرة الحقيقي
AD_C	تمنع استخدام التذكرة من محطة عمل غير تلك التي طلبت التذكرة في البداية
ID_V	تؤكد للخادم أنه قام بإزالة التشفير بشكلٍ صحيح
TS_4	تخبر الخادم عن وقت إصدار هذه التذكرة
$Lifetime_4$	تمنع إعادة الإرسال بعد انتهاء مدة التذكرة
$Authenticator_c$	تؤكد للخادم أن مُقدِّم التذكرة هو نفسه الزبون الذي صدرت له التذكرة؛ لها فترة عمر قصيرة جداً لمنع إعادة الإرسال
$K_{c,v}$	يُشفِّر المُوثَّق بمفتاح معروف فقط للزبون والخادم لمنع التلاعب
ID_c	يجب أن تطابق الهوية ID الموجودة بالتذكرة لتوثيق التذكرة
AD_c	يجب أن تطابق العنوان بالتذكرة لتوثيق التذكرة
TS_5	تخبر الخادم عن وقت إنشاء المُوثَّق

يوفر كيربيروس آلية لدعم التوثيق بين العوالم. وحتى يتمكن عالمان من دعم التوثيق بينهما لا بد من توافر متطلب ثالث:

3. يشترك خادم كيربيروس مع كل عالم منهما في مفتاحٍ سريٍ مع الخادم في العالم الآخر، كما يسجل كلٌ من خادمي كيربيروس لدى الآخر.

تتطلّب هذه الطريقة أن يثق خادم كيربيروس في أحد العالمين بخادم كيربيروس في العالم الآخر لتوثيق مستخدميه. علاوة على ذلك، يجب أيضاً أن ترغب الخادمات المشاركة في العالم الثاني في الثقة بخادم كيربيروس في العالم الأول. بهذه القواعد الأساسية، يمكننا أن نصف الآلية كما يلي (انظر الشكل 4-2): يحتاج المستخدم الذي يرغب في الخدمة الموجودة على خادم في عالم آخر إلى تذكرة لذلك الخادم. يتبع زبون هذا المستخدم الإجراءات الاعتيادية للوصول إلى خادم TGS المحلي، ثم يطلب تذكرة منح تذاكر لخادم TGS البعيد (أي TGS في العالم الآخر). بعدها يمكن للزبون أن يتقدم بطلب إلى TGS البعيد للحصول على تذكرة منح خدمة من الخادم المطلوب في عالم TGS البعيد.

يوضّح الشكل 4-2 تفاصيل تلك التبادلات كما يأتي (قارنها بالجدول 4-1):

(1) $C \rightarrow AS$: $ID_c \| ID_{tgs} \| TS_1$

(2) $AS \rightarrow C$: $E(K_c, [K_{c,tgs} \| ID_{tgs} \| TS_2 \| Lifetime_2 \| Ticket_{tgs}])$

(3) $C \rightarrow TGS$: $ID_{tgsrem} \| Ticket_{tgs} \| Authenticator_c$

(4) $TGS \rightarrow C$: $E(K_{c,tgs}, [K_{c,tgsrem} \| ID_{tgsrem} \| TS_4 \| Ticket_{tgsrem}])$

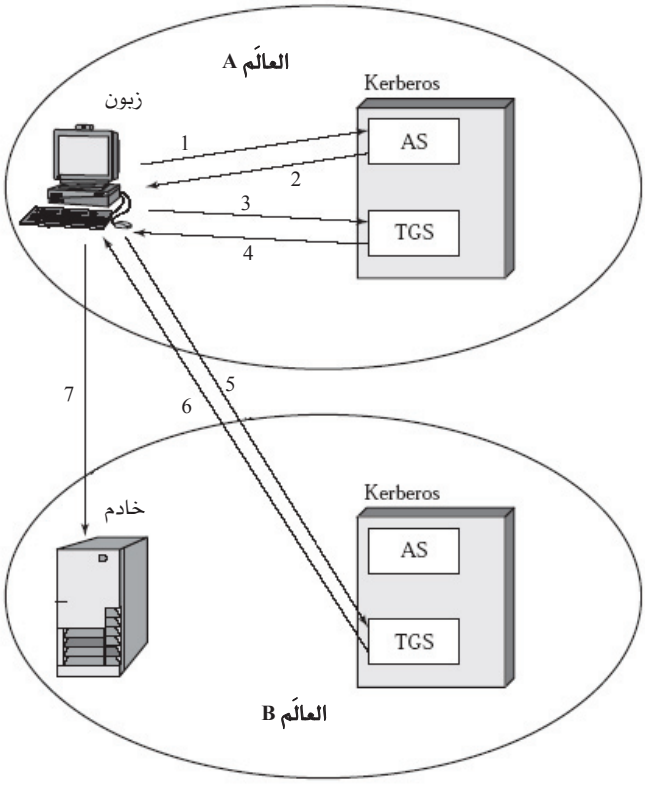
(5) $C \rightarrow TGS_{rem}$: $ID_{vrem} \| Ticket_{tgsrem} \| Authenticator_c$

(6) $TGS_{rem} \rightarrow C$: $E(K_{c,tgsrem}, [K_{c,vrem} \| ID_{vrem} \| TS_6 \| Ticket_{vrem}])$

(7) $C \rightarrow V_{rem}$: $Ticket_{vrem} \| Authenticator_c$

تبين التذكرة المُقدَّمة للخادم البعيد V_{rem} العالم الذي تم فيه توثيق المستخدم في الأصل. يختار الخادم ما إذا كان يلبي الطلب البعيد أو لا.

الشكل 4-1: نظرة عامة على كيربروس.



1. طلب تذكرة لخادم TGS المحلي
2. تذكرة لخادم TGS المحلي
3. طلب تذكرة لخادم TGS البعيد
4. تذكرة لخادم TGS البعيد
5. طلب تذكرة لخادم TGS البعيد
6. تذكرة للخادم البعيد
7. طلب لخدمة بعيدة

الشكل 4-2: طلب خدمة موجودة في عالم آخر.

إحدى المشكلات التي تعاني منها الطريقة السابقة أنها لا تعمل بشكلٍ جيد عند زيادة عدد العوالم. إذا كان هناك N من العوالم، فعندئذ لا بد أن يكون هناك عدد $N(N-1)/2$ من التبادلات الآمنة للمفاتيح حتى يتمكن كل عالم من العمل مع عوالم كيربيروس الأخرى.

4-1-3 إصدار كيربيروس الخامس

مواصفات إصدار كيربيروس الخامس مذكورة في طلب التعلقات RFC 1510، وبه عدد من التحسينات على الإصدار 4 [KOHL94]. في البداية سنلقي نظرة عامة على التغييرات من الإصدار 4 إلى الإصدار 5، و سنتناول بعد ذلك بروتوكول الإصدار 5.

❖ أوجه الاختلاف بين الإصدارين 4 و5:

يهدف الإصدار 5 إلى معالجة أوجه القصور الموجودة بالإصدار 4 في مجالي: أوجه القصور البيئية والتقنية. دعنا نُلخِّص بسرعة التحسينات في كلا المجالين.[6]

طُوِّر إصدار كيربيروس الرابع للاستخدام ضمن بيئة مشروع أثينا. ولذلك، لم تكن هناك حاجة ليكون متعدد الأغراض بشكلٍ كامل، مما أدى إلى وجود أوجه القصور البيئية الآتية:

1. الاعتماد على نظام التشفير: يتطلّب الإصدار 4 استخدام DES في التشفير. ولذا فإن قيود التصدير (export) على DES والشكوك المثارة حول مدى فعاليته كانت تمثل قلقاً. لذا فقد تم وسم الشفرة بمُعرِّف لنوع التشفير المُستخدَم في الإصدار 5؛ مما يسمح باستخدام أي خوارزمية من خوارزميات التشفير. كما تم وسم مفاتيح التشفير بالنوع (type) والطول (length)، مما يسمح باستخدام نفس المفتاح في خوارزميات مختلفة، كما يسمح بتحديد النوعيات المختلفة لخوارزمية ما.

[6] تتبع المناقشة التالية العرض الموجود في [KOHL94].

2. الاعتماد على بروتوكول الإنترنت: يتطلّب الإصدار 4 استخدام عناوين بروتوكول الإنترنت (IP). فهو لا يدعم العناوين الأخرى مثل عنوان شبكة ISO. أما الإصدار 5 فيسمح بوسم عناوين الشبكة بالنوع والطول، ومن ثَمَّ يسمح باستخدام أي نوع من عناوين الشبكة.
3. ترتيب بايتات الرسالة: في الإصدار 4 يرتب مُرسِل الرسالة البايتات حسب اختياره، ثم يبيّن في الرسالة ما إذا كان البايت الأقل وزناً في العنوان الأقل أو في العنوان الأعلى. رغم أن هذه الطريقة تؤدي الغرض المطلوب، إلا أنها لا تتبع الاصطلاحات المتعارف عليها. في الإصدار 5، تُعرَّف كل هياكل الرسالة باستخدام ASN.1 وقواعد التشفير الأساسية (BER)، مما يؤدي إلى ترتيب البايتات ترتيباً غير مبهم.
4. فترة الصلاحية للتذكرة: ي في الإصدار 4 يتم تكويد (ترميز) قيمة فترة الصلاحية في 8 بتات بوحدات من خمس دقائق. ولذا، تكون أقصى قيمة يمكن التعبير عنها لفترة الصلاحية هي $2^8 \times 5 = 1280$ دقيقة (أي أكثر قليلاً من 21 ساعة). قد تكون هذه الفترة غير كافية لبعض التطبيقات (كالمحاكاة التي تستغرق وقتاً طويلاً وتتطلّب اعتمادات كيربيروس صحيحة خلال فترة التنفيذ بكاملها). بدلاً من ذلك تتضمّن التذاكر في الإصدار 5 بداية الوقت ونهايته مما يسمح لها باستخدام أي قيم لفترات الصلاحية.
5. إعادة توجيه التوثيق: لا يسمح الإصدار 4 بإعادة توجيه الاعتمادات (credentials) الممنوحة لزبونٍ ما إلى مضيف آخر ومن ثَمَّ استخدامها من قِبَل زبونٍ آخر. لكن توفير هذه القدرة يُمكِّن الزبون من الوصول لخادم ما وجعله يتصل بخادم آخر نيابةً عنه. فعلى سبيل المثال، قد يصدر زبون طلباً إلى خادم الطباعة ويريده أن يصل بدوره لملف الزبون الموجود على خادم الملفات مستخدماً اعتمادات هذا الزبون لتنفيذ الوصول. يوفر الإصدار 5 هذه الإمكانية.
6. التوثيق بين العوالم: في الإصدار 4 يتطلب التشغيل البيني بين عدد N من العوالم علاقات من كيربيروس إلى كيربيروس برتبة (N^2) كما وصفنا

سابقاً. أما الإصدار 5 فيدعم طريقة تتطلب عدداً أقل من تلك العلاقات كما سنصف بعد قليل.

فضلاً عن تلك المحدودات البيئية، هناك أوجه قصور تقنية في بروتوكول الإصدار 4 ذاته، وقد تم ذكر أغلبها في [BELL90]. يحاول الإصدار 5 معالجة أوجه القصور الآتية:

1. مضاعفة التشفير: لاحظ أنه في الجدول 4-1 (الرسائل (2) و(4)) تُشفَّر التذاكر المُقدَّمة للزبائن مرّتين: مرة بالمفتاح السرّي لخادم الوجهة ومرة أخرى بمفتاح سري معروف للزبون. ويلاحظ أن التشفير الثاني ليس ضرورياً ويُضيّع وقتاً في الحساب.
2. تشفير PCBC: يستخدم التشفير في الإصدار 4 نمطاً غير قياسي بخوارزمية DES يُعرَف بـ PCBC (تسلسل كتل الشفرة المنتشر)[7]. وقد تم إثبات أن هذا النمط عرضة لهجوم يتضمّن تبديل كتل النص المُشفَّر [KOHL89]. استهدف PCBC توفير فحص سلامة البيانات كجزء من عملية التشفير. وفَّر الإصدار 5 آليات محددة للسلامة تسمح باستخدام نمط CBC المعياري في التشفير. على وجه التحديد يتم إلحاق مجموع تدقيقي (checksum) أو كود تحوير (hash) قبل تشفيرها باستخدام CBC.
3. مفاتيح الجلسة: تتضمّن كل تذكرة مفتاح جلسة يُستخدَم من قِبَل الزبون لتشفير المُوثَّق (authenticator) المُرسَل إلى الخدمة المرتبطة بتلك التذكرة. بالإضافة لذلك، قد يُستخدَم مفتاح الجلسة بعد ذلك من قِبَل الزبون والخادم لحماية الرسائل المارة أثناء تلك الجلسة. ومع ذلك، فبسبب استخدام نفس التذكرة مراراً وتكراراً للوصول إلى الخدمة من خادم معيّن، هناك خطر أن يقوم الخَصْم بتسجيل رسائل جلسة قديمة وإعادة إرسالها إلى الزبون أو الخادم. في الإصدار 5، يمكن أن يتفاوض الزبون والخادم حول مفتاح جلسة ثانوي يُستخدَم فقط لهذا الاتصال الوحيد. ولذا

[7] موصوف في الملحق A-4.

يؤدّي كل اتصال جديد من قِبَل الزبون إلى استخدام مفتاح جلسة ثانوي جديد.

4. هجمات كلمة السر: كلا الإصدارين 4 و5 عرضة لهجمات كلمة السر. وتتضمّن الرسالة من AS إلى الزبون مادة مُشفَّرة بمفتاح يعتمد على كلمة السر الخاصة بالزبون.[8] ويمكن أن يلتقط الخَصْم هذه الرسالة ويحاول إزالة التشفير بتجربة كلمات السر المختلفة. فإذا تمكن الخَصْم من حل الشفرة، فسيمكنه اكتشاف كلمة السر الخاصة بالزبون وقد يستخدمها بعد ذلك للحصول على اعتمادات التوثيق من كيربيروس. فهذا هو نفس نوع الهجوم على كلمة السر الموصوف في الفصل التاسع، ولذا يمكن تطبيق نفس أساليب المواجهة. يوفر الإصدار 5 آلية تُعرَف بإجراء ما قبل التوثيق (preauthentication)، والذي يجعل الهجوم على كلمة السر أكثر صعوبة، لكنّه لا يمنعه.

❖ حوار التوثيق بالإصدار 5:

يُلخِّص الجدول 4-3 الحوار الأساسي للإصدار 5، والذي يمكن توضيحه بشكلٍ أفضل بمقارنته بالإصدار 4 (انظر الجدول 4-1).

أولاً: خذ بعين الاعتبار تبادل خدمة التوثيق. تمثل الرسالة (1) رسالة طلب من الزبون لتذكرة منح تذاكر. وكما سبق، تتضمّن تلك الرسالة هوية المستخدم وهوية خادم TGS، مع إضافة العناصر الجديدة الآتية:

- عالم (realm): يشير إلى عالم المستخدم.
- خيارات (options): تُستخدَم لطلب تعيين أعلام (إشارات) (flags) معينة في التذكرة المعادة.

[8] يصف الملحق A-4 تحويل كلمات السر إلى مفاتيح تشفير.

الجدول 4-3: ملخص للرسائل المتبادلة في الإصدار 5 لـ كيربيروس.

(1) C → AS $Options \| ID_c \| Realm_c \| ID_{tgs} \| Times \| Nonce_1$

(2) AS → C $Realm_c \| ID_C \| Ticket_{tgs} \| E(K_c, [K_{c,tgs} \| Times \| Nonce_1 \| Realm_{tgs} \| ID_{tgs}])$

$Ticket_{tgs} = E(K_{tgs}, [Flags \| K_{c,tgs} \| Realm_c \| ID_C \| AD_C \| Times])$

(a) تبادل خدمة توثيق للحصول على تذكرة منح تذكرة

(3) C → TGS $Options \| ID_v \| Times \| \| Nonce_2 \| Ticket_{tgs} \| Authenticator_c$

(4) TGS → C $Realm_c \| ID_C \| Ticket_v \| E(K_{c,tgs}, [K_{c,v} \| Times \| Nonce_2 \| Realm_v \| ID_v])$

$Ticket_{tgs} = E(K_{tgs}, [Flags \| K_{c,tgs} \| Realm_c \| ID_C \| AD_C \| Times])$

$Ticket_v = E(K_v, [Flags \| K_{c,v} \| Realm_c \| ID_C \| AD_C \| Times])$

$Authenticator_c = E(K_{c,tgs}, [ID_C \| Realm_c \| TS_1])$

(b) تبادل خدمة منح تذكرة للحصول على تذكرة منح خدمة

(5) C → V $Options \| Ticket_v \| Authenticator_c$

(6) V → C $E_{K_{c,v}}[TS_2 \| Subkey \| Seq\#]$

$Ticket_v = E(K_v, [Flags \| K_{c,v} \| Realm_c \| ID_C \| AD_C \| Times])$

$Authenticator_c = E(K_{c,v}, [ID_C \| Realm_c \| TS_2 \| Subkey \| Seq\#])$

(c) تبادل توثيق بين الخادم والزبون للحصول على خدمة

- أوقات (times): تُستخدَم من قِبَل الزبون لطلب تعيين إعدادات الوقت الآتية بالتذكرة:
  - from: وقت البداية المطلوب للتذكرة المطلوبة.
  - till: وقت الانتهاء المطلوب للتذكرة المطلوبة.
  - rtime: الوقت المطلوب لتجديد وقت الإنتهاء "till"
- nonce: قيمة عشوائية تُكرَّر في الرسالة (2) لضمان أن يكون الردّ جديداً وليس ناتجاً عن إعادة إرسال من قِبَل خَصْمٍ ما.

تُرجِع الرسالة (2) تذكرة منح تذاكر، ومعلومات تعريف الزبون، وكتلة مُشفَّرة باستخدام مفتاح تشفير يعتمد على كلمة السر للمستخدم. تتضمّن هذه الكتلة مفتاح الجلسة الذي سيُستخدَم بين الزبون وTGS، والأوقات المحدّدة بالرسالة (1)، والـ nonce المحدد بالرسالة (1)، ومعلومات تعريف TGS. تتضمّن التذكرة نفسها مفتاح الجلسة، ومعلومات تعريف الزبون، وقيم الوقت المطلوبة،

والخيارات المطلوبة، وأعلام (إشارات) تعكس حالة هذه التذكرة. توفر تلك الأعلام وظائف مهمة جديدة للإصدار 5. سنرجئ مناقشة تلك الأعلام في الوقت الحالي وسنركّز على البنية العامة لبروتوكول الإصدار 5.

دعنا الآن نقارن خدمة منح التذاكر بالإصدارين 4 و5. تتضمّن الرسالة (3) بالإصدارين مُوثّقاً وتذكرة واسم الخدمة المطلوبة. بالإضافة إلى ذلك يقوم الإصدار 5 أيضاً بتضمين الأوقات المطلوبة وخيارات التذكرة وnonce. ولكل مكوِّن وظيفة مشابهة لنظيره الموجود بالرسالة (1). أما المُوثّق ذاته فهو – بشكلٍ أساسي – تماماً كالموجود بالإصدار 4.

تتبع الرسالة (4) نفس بنية الرسالة (2)، حيث تُرجِع تذكرة بالإضافة للمعلومات المطلوبة من قِبَل الزبون والتي تكون مُشفَّرة بمفتاح الجلسة الحالي المشترك بين الزبون وTGS.

وأخيراً، بالنسبة للتوثيق المتبادل بين الزبون والخادم، فقد ظهرت عدة خصائص جديدة في الإصدار 5. ففي الرسالة (5)، قد يحدد الزبون بأحد الخيارات الحاجة للتوثيق المتبادل. ويتضمّن المُوثّق عدة حقول جديدة كما يأتي:

- مفتاح ثانوي: مفتاح تشفير مختار من قِبَل الزبون لاستخدامه لحماية جلسة التطبيق المحددة تلك. إذا حُذِف هذا الحقل، فسيتم استخدام مفتاح الجلسة من التذكرة لهذا الغرض.
- رقم التسلسل: حقل اختياري يحدّد رقم بداية التسلسل ليستخدمه الخادم للرسائل التي يرسلها إلى الزبون أثناء هذه الجلسة. قد يتم ترقيم الرسائل بالتسلسل لاكتشاف إعادة الإرسال.

إذا كان التوثيق المتبادل مطلوباً، يرد الخادم بالرسالة (6) والتي تتضمّن خاتم الوقت من المُوثّق. لاحظ أنه في الإصدار 4 كانت قيمة خاتم الوقت تُزاد واحداً. وهذا الأمر ليس ضرورياً في الإصدار 5، لأن طبيعة صيغة الرسائل تكون بحيث لا يمكن أن يقوم خَصْم ما بإنشاء الرسالة (6) بدون معرفة مفاتيح التشفير الملائمة.

يكون للمفتاح الثانوي – في حالة وجوده – بالرسالة (6) الهيمنة على المفتاح الثانوي – في حالة وجوده – بالرسالة (5). كما سيحدّد حقل رقم التسلسل الاختياري رقم بداية التسلسل للاستخدام من قِبَل الزبون.

الجدول 4-4: أعلام الإصدار 5 لـ كيربيروس.

INITIAL	هذه التذكرة أُصدرت باستخدام بروتوكول خادم التوثيق AS ولم تُصدر بناءً على تذكرة منح تذاكر
PRE-AUTHENT	أثناء مرحلة التوثيق الأولى، تم التحقق من الزبون من قِبَل KDC قبل إصدار التذكرة
HW-AUTHENT	تطلّب البروتوكول المُستخدم في مرحلة التوثيق الأولى استخدام المكونات المادية المتوقع امتلاكها فقط من قِبَل الزبون المُسمّى
RENEWABLE	تخبر TGS أن هذه التذكرة يمكن استخدامها للحصول على تذكرة بديلة تنتهي في وقت لاحق
MAY-POSTDATE	تخبر TGS أنه يمكن إصدار تذكرة منتهية الصلاحية بناءً على تذكرة منح تذاكر هذه
POSTDATED	تشير إلى أن هذه التذكرة انتهت صلاحيتها، يمكن للخادم الطرفي أن يفحص حقل زمن التوثيق (authtime) لمعرفة متى تم التوثيق الأصلي
INVALID	هذه التذكرة غير صالحة ويجب تصحيحها عن طريق KDC قبل استخدامها
PROXIABLE	تخبر TGS أنه يمكن إصدار تذكرة منح خدمة جديدة بعنوان شبكة مختلف بناءً على التذكرة المقدمة
PROXY	تشير إلى أن هذه التذكرة مفوَّض
FORWARDABLE	تخبر TGS أنه يمكن إصدار تذكرة منح تذاكر جديدة بعنوان شبكة مختلف بناءً على تذكرة منح تذاكر هذه
FORWARDED	تشير إلى أن هذه التذكرة إما أنه تم إعادة توجيهها أو تم إصدارها بناءً على توثيق يشمل تذكرة منح تذاكر تم إعادة توجيهها

❖ أعلام (إشارات) التذكرة:

في الإصدار 5 تدعم الأعلام المُتضمَّنة في التذكرة وظائف أكثر مقارنة بتلك الموجودة في الإصدار 4. يلخّص الجدول 4-4 الأعلام التي يمكن تضمينها في تذكرة ما:

يُشير العَلَم INITIAL (البداية) إلى أن هذه التذكرة تم إصدارها من قِبَل AS، وليس من قِبَل TGS. عندما يطلب زبون تذكرة منح خدمة من TGS، يقدّم تذكرة منح التذاكر التي حصل عليها من AS. في الإصدار 4، كان هذا هو الطريق الوحيد للحصول على تذكرة منح خدمة. أما الإصدار 5 فيوفر إمكانية إضافية للزبون تسمح له بالحصول على تذكرة منح خدمة مباشرةً من AS. ويكون ذلك مفيداً في حال رغبة خادمٍ ما (كخادم تغيير كلمة السر) في معرفة كون كلمة السر للزبون قد تم اختبارها مؤخراً.

يُشير العَلَم PRE-AUTHENT (التوثيق المسبق) إلى أن AS قام بتوثيق الزبون قبل إصدار التذكرة حال تسلّمه أول طلب (أي الرسالة (1)). ولم تحدد صيغة معينة لهذا التوثيق المسبق. على سبيل المثال، في تنفيذ MIT للإصدار 5 يكون الوضع الافتراضي هو التوثيق المسبق المُشفَّر بخاتم الوقت. وعندما يريد مستخدم ما الحصول على تذكرة، يجب عليه أن يرسل إلى AS كتلة التوثيق المسبق متضمنةً رقماً عشوائياً ورقم الإصدار وخاتم الوقت ومُشفَّرةً بمفتاح الزبون الذي يعتمد على كلمة السر. ويقوم AS بإزالة تشفير الكتلة ولا يرسل تذكرة منح تذاكر ما لم يكن خاتم الوقت بكتلة التوثيق المسبق في حدود الوقت المسموح به (بعد أخذ انحراف الساعة وتأخيرات الشبكة في الحسبان). احتمال آخر هو استخدام بطاقة ذكية (smart card) لتوليد كلمات سرّ متغيرة بشكلٍ مستمر تُضمَّن في الرسائل الموثَّقة مسبقاً. يمكن أن تعتمد كلمات السر التي تم توليدها بواسطة البطاقة على كلمة سرّ معينة للمستخدم لكن يتم تحويلها بالبطاقة حتى تكون النتيجة استخدام كلمات سر اعتباطية. يمنع ذلك حدوث هجوم يعتمد على تخمين سهل

لكلمات السرّ. عند استخدام بطاقة ذكية أو أداة مماثلة، ويشار لذلك بعَلَم HW-AUTHENT.

عندما تكون فترة الصلاحية للتذكرة طويلة، فإن فترة احتمال سرقتها واستخدامها من قِبَل خَصْم ما تكون طويلة. أما إذا استُخدِمت فترة عمر قصيرة – للتقليل من هذا التهديد – فسيكون هناك عبء إضافي للحصول على تذاكر جديدة. في حالة تذكرة منح تذاكر، يجب على الزبون تخزين المفتاح السري للمستخدم (والذي يُشكِّل خطراً بشكلٍ جلي جداً)، أو طلب كلمة السرّ من المستخدم مراراً وتكراراً. طريقة أخرى تُوازن بين كلا الأمرين هي استخدام التذاكر القابلة للتجديد (renewable). أما التذكرة التي يكون بها العَلَم RENEWABLE محدداً فتتضمّن وقتين لانتهاء الصلاحية: وقت لانتهاء هذه التذكرة ما لم تُجدَّد، ووقت يمثل أطول وقت مسموح به لانتهاء الصلاحية تماماً لهذا الزبون. يمكن أن يجدد الزبون التذكرة بتقديمها إلى TGS مع طلب وقت انتهاء جديد. إذا كان الوقت الجديد ضمن القيمة المسموح بها (أي أقل من قيمة انتهاء الصلاحية الثانية)، يمكن أن يصدر TGS تذكرة جديدة بوقت جلسة جديد ووقت انتهاء لاحق. إن فائدة هذه الآلية هي تمكين TGS من رفض تجديد تذكرة تم الإبلاغ عن سرقتها.

قد يطلب الزبون أن يعطي AS تذكرة منح تذاكر يكون فيها العَلَم MAY-POSTDATE محدداً. ويمكن أن يستخدم الزبون هذه التذكرة لطلب تذكرة مُؤشَّرة كـ POSTDATED (أي انتهى وقتها) وINVALID (أي غير صحيحة) من TGS. بعد ذلك، قد يُقدِّم الزبون تلك التذكرة للتصديق. يمكن أن يكون هذا الأسلوب مفيداً لتشغيل دفعة شغل (batch job) طويلة على خادم يتطلب تذكرة بشكلٍ دوري. فيمكن أن يحصل الزبون علي عدد من التذاكر لهذه الجلسة بقيم وقت مختلفة جملةً واحدةً. تكون كل تلك التذاكر – ما عدا الأولى – غير صحيحة في البداية. وعندما يصل التنفيذ لنقطة زمنية معينة تتطلب تذكرة جديدة، يمكن أن يحصل الزبون على تصديق للتذكرة الملائمة. وبهذه الطريقة، لا يحتاج

الزبون إلى استخدام تذكرة منح التذاكر الخاصة به مراراً وتكراراً للحصول على تذكرة منح خدمة.

في الإصدار 5 من المحتمل أن يعمل خادم كوكيل نيابةً عن زبون، أي يتبنّى اعتمادات وامتيازات الزبون لطلب خدمة من خادم آخر. إذا رغب زبون في استخدام هذه الآلية، يطلب تذكرة منح تذاكر يكون بها العَلَم PROXIABLE محدداً. عندما تُقدَّم هذه التذكرة إلى TGS، يُسمَح له بإصدار تذكرة منح خدمة بعنوان شبكة مختلف؛ هذه التذكرة الأخيرة سيكون بها عَلَم PROXY معيَّناً. وقد يقبلها التطبيق الذي يستلمها، أو يطلب تحققاً إضافياً لتوفير أثر للمراجعة (audit trail). ويمكن للقارئ الاطلاع على مناقشة حول بعض الاستخدامات الممكنة لخاصية الوكيل في [NEUM93b].

يعدّ مفهوم الوكيل حالة خاصة ومحدودة لإجراء التمرير الأكثر قوّة. إذا كان عَلَم FORWARDABLE لتذكرة ما محدداً، فيمكن أن يُصدِر TGS تذكرة منح تذاكر إلى الطالب بعنوان شبكة مختلف وعَلَم FORWARDED محدداً. ويمكن تقديم هذه التذكرة إلى خادم TGS البعيد. تسمح هذه الإمكانية للزبون بالدخول إلى خادم على عالم آخر بدون أن يتطلب ذلك أن يحتفظ كل عالم من عوالم كيربيروس بمفتاح سري بخادمات كيربيروس في كلٍّ من العوالم الأخرى. على سبيل المثال، يمكن تنظيم العوالم بشكلٍ هرمي. ومن ثَمَّ يتحرك الزبون لأعلى الشجرة حتى يصل إلى عقدة مشتركة وبعدها ينزل ليصل إلى العالم المستهدف. قد تتضمّن كل خطوة يتحركها الزبون تمرير تذكرة منح التذاكر إلى TGS التالي على المسار.

2-4 خدمة X.509 للتوثيق

يمثل معيار الاتحاد الدولي للاتصالات السلكية واللاسلكية X.509 جزءاً من سلسلة معايير X.500 التي تُعرِّف خدمة الدليل. والدليل هو في الواقع خادم أو مجموعة خادمات موزَّعة تحتفظ بقاعدة بيانات عن المستخدمين. تتضمّن المعلومات

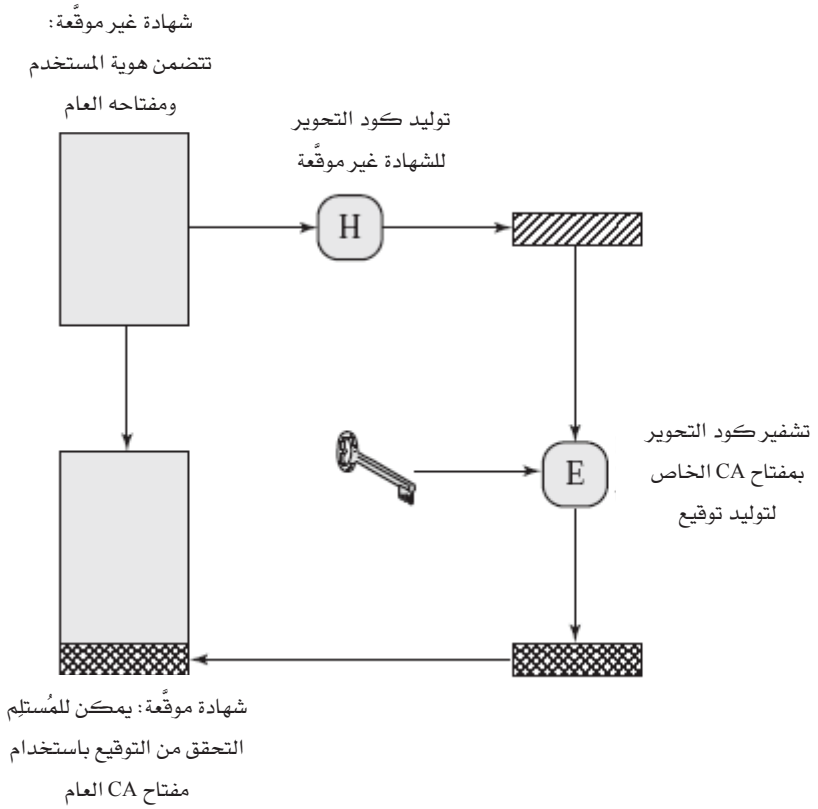
تحويل اسم المستخدم إلى عنوان شبكة، بالإضافة إلى خواص ومعلومات أخرى عن المستخدمين.

يُعرِّف X.509 إطاراً لتوفير خدمات التوثيق لمستخدمي دليل X.500. قد يُستخدَم الدليل كمستودع لشهادات المفاتيح العامة من النوع الذي تم مناقشته في الفصل الثالث. تحتوي كل شهادة على المفتاح العام للمستخدم وتكون موَقَّعةً بالمفتاح الخاص لهيئة شهادات مؤتمنة. بالإضافة لذلك يُعرِّف X.509 بروتوكولات توثيق بديلة تعتمد على استخدام شهادات المفاتيح العامة.

يُعدّ X.509 معياراً مهماً لأن بنية الشهادة وبروتوكولات التوثيق المُعرَّفة في X.509 تُستخدَم في سياقات متنوعة. فعلى سبيل المثال، تُستخدَم صيغة شهادة X.509 في بروتوكول S/MIME (راجع الفصل الخامس)، وأمن IP (راجع الفصل السادس)، وبروتوكول SSL/TLS وبروتوكول SET (راجع الفصل السابع).

كان أول إصدار لـ X.509 عام 1988. ثم بعد ذلك تم تعديله ليتناول بعضاً من المخاوف الأمنية المذكورة في [IANS90] و[MITC90]؛ وتم إصدار معيار مُعدَّل في عام 1993. وقد ظهر الإصدار الثالث منه في عام 1995 ثم عُدِّل في عام 2000.

يعتمد X.509 على استخدام التشفير بالمفاتيح العامة والتوقيعات الرقمية. ولا يملي المعيار استخدام خوارزمية معينة لكنه يوصي بـ RSA. ويفترض أن أسلوب التوقيع الرقمي يتطلب استخدام دالة تحوير (hash function). كما لا يُملي المعيار خوارزمية معينة لدالة التحوير. وتَضمَّن معيار عام 1988 وصفاً لخوارزمية موصى بها لدالة التحوير؛ لكن منذ ذلك الحين تم إثبات أن هذه الخوارزمية غير آمنة ومن ثَمَّ أُسقطَت من معيار عام 1993. ويبيّن الشكل 4-3 عملية توليد شهادة للمفتاح العام.



الشكل 4-3: استخدام شهادة المفتاح العام.

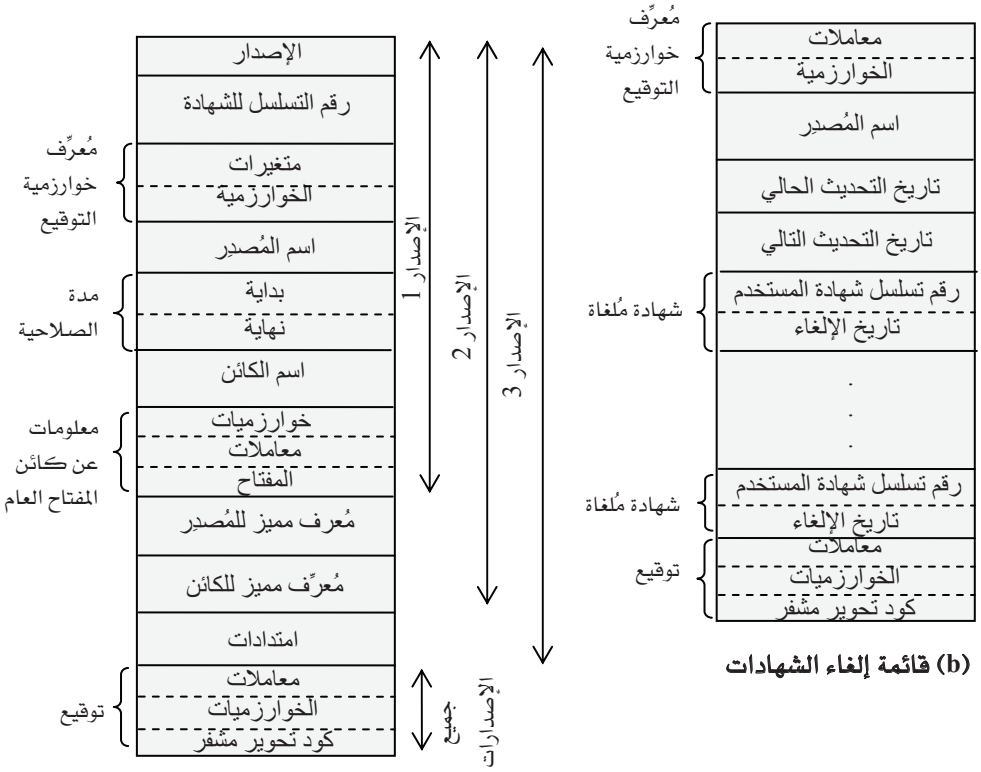
4-2-1 الشهادات

إنّ أساس X.509 هو استخدام شهادة المفتاح العام الخاصة بكل مستخدم. ويُفترَض أن تكون شهادات المستخدمين تلك قد تم إنشاؤها من قِبَل هيئة شهادات (CA) مؤتمنة، وتم وضعها في الدليل من قِبَل CA أو من قِبَل المستخدم. ويقتصر دور خادم الدليل فقط على توفير موقع يسهل للمستخدمين الوصول إليه للحصول على الشهادات، وليست له أي مسؤولية عن إنشاء المفاتيح العامة أو عن عملية إثبات صحة الشهادات.

يبيّن الشكل 4-4 (a) الصيغة العامة للشهادة، والتي تتضمّن العناصر الآتية:

- رقم الإصدار: للتمييز بين الإصدارات المتعاقبة لصيغة الشهادة؛ وتكون القيمة الافتراضية لرقم الإصدار هي 1. وفي حالة وجود مُعرِّف فريد لمُصدِر الشهادة أو مُعرِّف فريد للكائن، يجب أن يكون رقم الإصدار 2. أما في حالة وجود امتداد (extension) أو أكثر، يجب أن يكون رقم الإصدار 3.
- رقم التسلسل: عدد صحيح فريد ضمن هيئة CA التي تُصدِر الشهادة يرتبط بشكلٍ واضح بهذه الشهادة.
- مُعرِّف خوارزمية التوقيع: الخوارزمية المُستخدَمة لتوقيع الشهادة، مع أي مُعامِلات (بارامترات) ذات صلة. وبسبب تكرار تلك المعلومات في حقل التوقيع في نهاية الشهادة، فإن فائدة هذا الحقل لا تُذكر (إن وُجدت).
- اسم مُصدِر الشهادة: اسم X.509 لـ CA التي تنشئ الشهادة وتوقعها.
- فترة الصلاحية: وتتألف من تاريخين: تاريخ بداية صلاحية الشهادة وتاريخ نهايتها.
- اسم الكائن: اسم المُستخدِم الذي تُشير إليه هذه الشهادة. أي أن هذه الشهادة توثق المفتاح العام للشخص الذي يحمل المفتاح الخاص المناظر.
- معلومات المفتاح العام للكائن: المفتاح العام للكائن، ومُعرِّف الخوارزمية التي ستستخدم المفتاح، وأي مُعامِلات أخرى ذات صلة.
- مُعرِّف فريد للمُصدِر: وهو حقل اختياري مؤلَّف من سلسلة بتات تُستخدَم لتمييز CA التي أصدرت الشهادة في حالة إعادة استخدام اسم X.509 مرة ثانية لكيانات مختلفة.
- مُعرِّف فريد للكائن: وهو حقل اختياري مؤلَّف من سلسلة بتات تُستخدَم لتمييز "الكائن" في حالة إعادة استخدام اسم X.509 مرة ثانية لكيانات مختلفة.

- الامتدادات: مجموعة من حقل أو أكثر من حقول الامتدادات. أضيفت الامتدادات في الإصدار 3 وستناقش لاحقاً في هذا الجزء.
- التوقيع: يغطي كل الحقول الأخرى في الشهادة؛ ويحتوي على كود التحوير للحقول الأخرى مُشفَّراً بمفتاح CA الخاص. ويتضمّن هذا الحقل مُعرِّف خوارزمية التوقيع.



(a) شهادة X.509

(b) قائمة إلغاء الشهادات

الشكل 4-4: صيغ X.509.

ولقد تم إضافة حقول المُعرِّفات الفريدة تلك في الإصدار 2 لمعالجة إعادة الاستخدام المحتمل لاسم الكائن أو المُصدِر بمرور الوقت. ونادراً ما تستخدم هذه الحقول.

يَستخدِم المعيار الاصطلاح الآتي لتعريف شهادة:

$$CA \ll A \gg = CA \{V, SN, AI, CA, T_A, A, Ap\}$$

حيث:

Y<<X>>: شهادة المستخدم X الصادرة من هيئة الشهادات Y.

Y{I}: كود التحوير الملحق.

تُوقِّع CA الشهادة بمفتاحها الخاص. إذا كان المفتاح العام المناظر معروفاً للمستخدم، فسيستطيع هذا المستخدم التحقق من أن الشهادة المُوقَّعة من قِبَل CA صحيحة. هذه هي الطريقة المعتادة للتوقيع الرقمي وموضحة بالشكل 3-2 (b).

❖ الحصول على شهادة المُستخدِم:

تتسم شهادة المُستخدِم المولَّدة من قِبَل CA بالخصائص الآتية:

- أي مُستخدِم يمكنه الوصول للمفتاح العام لـ CA يمكنه التحقق من المفتاح العام لمستخدم سبق توثيقه.
- لا يمكن لأى طرف آخر عدا هيئة الشهادات أن يعدِّل الشهادة بدون أن يُكتشَف.

نظراً لعدم قابلية الشهادات للتزوير، يمكن وضعها في دليل دون الحاجة إلى بذل جهود خاصة من قِبَل الدليل لحمايتها.

عند اشتراك كل المستخدمين بنفس هيئة الشهادات (CA)، يكون هناك ثقة مشتركة بتلك الهيئة. ويمكن أن توضع كل شهادات المستخدمين في الدليل ليتمكن كل المستخدمين من الوصول إليها. كما يمكن أيضاً أن يُرسِل المستخدم

شهادته مباشرةً إلى المستخدمين الآخرين. وفي أيٍّ من الحالتين، عندما يتوفر لـ B شهادة A، يكون B واثقاً من أن الرسائل التي يشفّرها بمفتاح A العام ستكون آمنة من التنصت، وتكون الرسائل الموقَّعة بمفتاح A الخاص غير قابلة للتزوير.

في حالة وجود تجمع كبير من المستخدمين، فإن اشتراك كل المستخدمين بنفس CA قد لا يكون عملياً. ولأن CA هي التي توقّع الشهادات، يجب أن يكون عند كل مستخدم مشترك نسخة من مفتاح CA العام للتحقق من التوقيعات. يجب أن يُعطَى هذا المفتاح العام لكل مُستخدِم بطريقة آمنة جداً (من حيث السلامة والتوثيق) حتى تتوفر الثقة للمُستخدِم في الشهادات المرتبطة. وهكذا مع كثير من المستخدمين، قد يكون عملياً أكثر أن يوجد عدد من هيئات الشهادات، كلٌّ منها يعطى مفتاحه العام إلى مجموعة من المستخدمين بشكلٍ آمن.

دعنا نفترض الآن أن A حصل على شهادة من هيئة الشهادات X_1 وأن B حصل أيضاً على شهادة من هيئة الشهادات X_2. فإذا كان A لا يعرف المفتاح العام لـ X_2 بشكلٍ آمن، ستكون شهادة B الصادرة من قِبَل X_2 عديمة الفائدة لـ A. فرغم أن A يمكنه قراءة شهادة B، إلا إنه لا يستطيع التحقق من التوقيع. ومع ذلك، إذا تمكنت الهيئتان من تبادل مفتاحيهما العامين بشكلٍ آمن، فسيتمكَّن A من الحصول على مفتاح B العام بالإجراء الآتي:

1. يحصل A من الدليل على شهادة X_2 مُوقَّعة من قبل X_1. ونظراً لأن A يعرف بشكلٍ آمن المفتاح العام لـ X_1، فسيمكنه الحصول على المفتاح العام لـ X_2 من شهادته، وسيمكنه التحقّق منها عن طريق توقيع X_1 عليها.
2. يعود A مرة أخرى إلى الدليل ويحصل على شهادة B موقَّعة من قِبَل X_2. ونظراً لأن A لديه الآن نسخة موثَّقة من مفتاح X_2 العام، فسيمكنه التحقق من التوقيع، وسيمكنه الحصول بشكلٍ آمن على مفتاح B العام.

بهذا يكون A قد استخدم سلسلة من الشهادات للحصول على مفتاح B العام. في اصطلاحات X.509، يتم التعبير عن هذه السلسلة كالآتي:

$$X_1 \ll X_2 \gg X_2 \ll B \gg$$

بنفس الأسلوب، يمكن أن يحصل B على مفتاح A العام بالسلسلة العكسية:

$$X_2 \ll X_1 \gg X_1 \ll A \gg$$

هذه الطريقة ليست بالضّرورة مقصورة على سلسلة من شهادتين. فبشكلٍ عام يمكن تتبع مسار طويل من الهيئات لإنتاج سلسلة. يتم التعبير عن سلسلة مكونة من N من العناصر كما يأتي:

$$X_1 \ll X_2 \gg X_2 \ll X_3 \gg \cdots X_N \ll B \gg$$

في هذه الحالة، كل زوج من الهيئات في السلسلة لا بد وأن يكون قد أنشأ شهادات لبعضهما بعضاً.

تحتاج كل هذه الشهادات للهيئات لأن تظهر في الدليل، وينبغي على المستخدم معرفة كيفية ربطهم حتى يمكنه اتباع مسار إلى شهادة المفتاح العام لمستخدم آخر. وتقترح X.509 أن تكون الهيئات مُرتَّبة بشكلٍ هرمي لكي تكون الحركة مباشرة.

يوضّح الشكل 4-5 (مأخوذ من X.509) مثالاً لمثل هذا الترتيب الهرمي. تشير الدوائر الموصلة إلى العلاقة الهرمية بين هيئات إصدار الشهادات؛ بينما تمثل المستطيلات المرتبطة بها شهادات موجودة بالدليل لكل مُدخَل CA. يتضمّن مُدخَل الدليل لكل هيئة نوعين من الشهادات:

- شهادات تمرير: شهادات X المولَّدة من قِبَل هيئات أخرى.
- شهادات عكسية: الشهادات المولَّدة من قِبَل X والتي تمثل شهادات الهيئات الأخرى.

في هذا المثال، يحصل المستخدم A على الشهادات الآتية من الدليل لتأسيس مسار التصديق إلى B:

$$X \ll W \gg W \ll V \gg V \ll Y \gg Y \ll Z \gg Z \ll B \gg$$

عندما يحصل A على هذه الشهادات، يمكنه أن يفتح مسار التصديق بالتسلسل لاستعادة نسخة مؤتمنة من مفتاح B العام. باستخدام هذا المفتاح العام، يرسل A رسائل مُشفَّرة إلى B. إذا رغب A في استلام رسائل مُشفَّرة من B، أو في توقيع الرسائل المرسلة إلى B، فسيطلب B مفتاح A العام، والذي يمكن الحصول عليه من مسار التصديق الآتي:

$$Z \ll Y \gg Y \ll V \gg V \ll W \gg W \ll X \gg X \ll A \gg$$

يمكن أن يحصل B على هذه المجموعة من الشهادات من الدليل، أو يمكن أن يزوّده بها A كجزء من رسالته الأولى له.

❖ إلغاء الشهادات:

كما بالشكل 4-4 فإن كل شهادة تتضمّن فترة صلاحية، تماماً كبطاقات الائتمان. عادةً ما يتم إصدار شهادة جديدة فقط قبل انتهاء صلاحية الشهادة القديمة. بالإضافة لذلك، قد يكون من المرغوب فيه أحياناً إلغاء شهادة قبل أن تنتهي صلاحيتها، لأحد الأسباب الآتية:

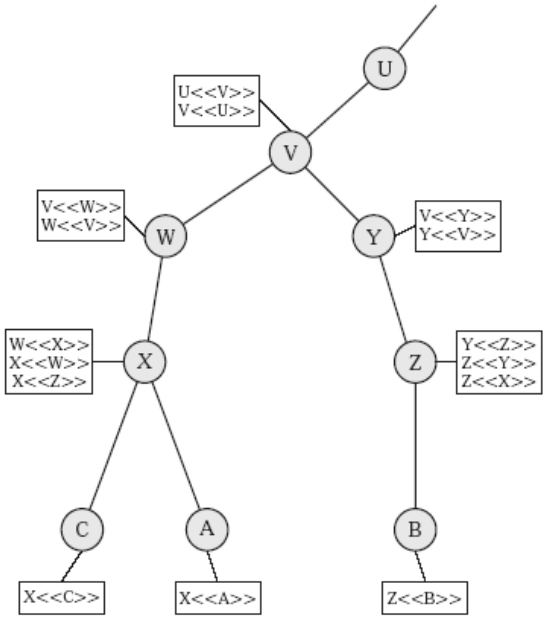
1. افتراض أن مفتاح المستخدم الخاص قد تعرض لخطر (مثلاً أصبح معروفاً للمهاجم).
2. لم يعد المستخدم مُوَثَّقاً من قِبَل تلك الهيئة CA.
3. افتراض أن شهادة CA قد تعرضت لخطر.

يجب أن تحتفظ كل هيئة من هيئات الشهادات بقائمة تشمل كل الشهادات من إصدارها التي تم إلغاؤها قبل انتهاء فترة صلاحيتها، بما في ذلك الشهادات التي أصدرت إلى مستخدمين وإلى هيئات شهادات أخرى. كما يجب أيضاً أن تُرسَل هذه القوائم للدليل. توقَّع كل قائمة من قوائم الشهادات المُلغاة (CRL) االتي تُرسَل للدليل من قِبَل المُصدِر وتتضمّن (انظر الشكل 4-4 (b)) اسم المُصدِر، وتاريخ إنشاء القائمة، والتاريخ الذي ستصدر عنده CRL التالية، ومُدخَل لكل شهادة مُلغاة.

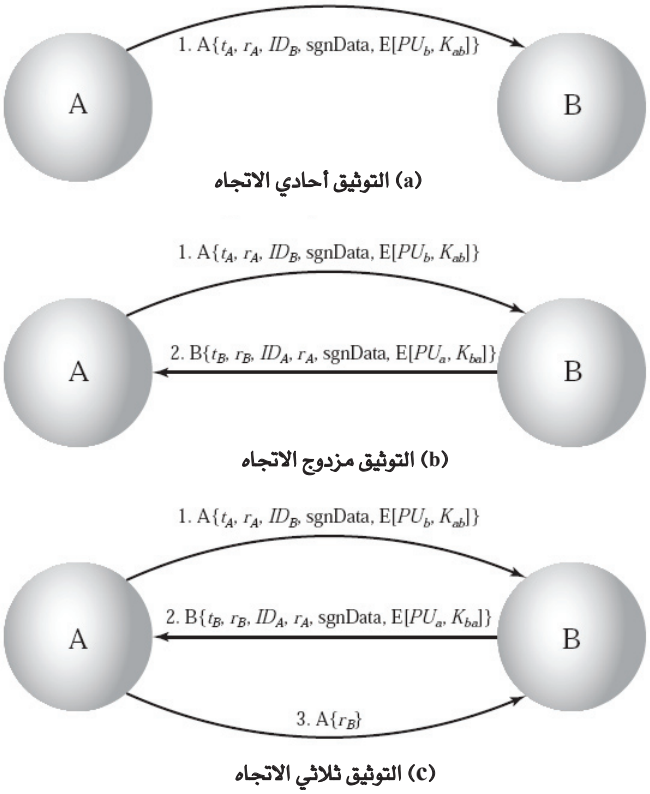
يشمل كل مُدخَل بالقائمة رقم التسلسل للشهادة المُلغاة وتاريخ إلغائها. ونظراً لأن أرقام التسلسل فريدة ضمن هيئة الشهادات، يكون رقم التسلسل كافياً لتمييز الشهادة. عندما يستلم مُستخدِمٌ شهادة في رسالة، يجب أن يقرّر ما إذا كانت تلك الشهادة مُلغاة أو لا. يمكن أن يفحص المستخدم الدليل كل مرة يستلم شهادة. لتفادى التأخيرات (والكلفة المحتملة) المصاحبة للبحث في الدليل، يمكن أن يحتفظ المستخدم بذاكرة وسيطة محليّة للشهادات وقوائم الشهادات المُلغاة.

4-2-2 إجراءات التوثيق

يتضمّن X.509 أيضاً ثلاثة بدائل لإجراءات التوثيق بهدف استخدامها عبر تطبيقات مختلفة. وتستخدم كل هذه الإجراءات توقيعات المفتاح العام. كما تفترض أن كل طرف يعرف المفتاح العام للآخر، عن طريق الحصول على شهادات الطرف الآخر من الدليل أو بتضمين الشهادة في الرسالة الأولى من كل طرف. ويوضّح الشكل 4-6 الإجراءات الثلاثة.



الشكل 4-5: مثال افتراضي للترتيب الهرمي في X.509.



الشكل 4-6: إجراءات توثيق قوية لـ X.509.

❖ التوثيق أحادي الاتجاه:

يتضمّن التوثيق أحادي الاتجاه انتقال معلومات من أحد المُستخدِمَين A إلى الآخر B؛ كما يُرسِّخ الآتي:

1. هوية A والتحقق من أن الرسالة قد أنشئت من قِبَل A.
2. التحقق من أن الرسالة موجهة لـ B.

3. إثبات سلامة وأصالة الرسالة (أي عدم تغيير محتواها وعدم إرسالها عدة مرات).

لاحظ أن هوية الكيان البادئ فقط هي التي يتم توثيقها في هذه العملية، وليست هوية كيان المُجيب. كحدٍّ أدنى، تتضمّن الرسالة خاتم وقت t_A وnonce r_A وهوية B، وتكون مُوقَّعة بمفتاح A الخاص. يتكون خاتم الوقت من وقت توليد الرسالة (وهو اختياري) ووقت انتهاء صلاحيتها. يساعد ذلك في منع تسليم الرسائل متأخرة. ويستخدم الـ nonce لاكتشاف هجمات إعادة الإرسال. لكن يجب أن تكون قيمة الـ nonce فريدة خلال وقت الرسالة. ومن ثَمَّ يمكن أن يُخزِّن B قيمة الـ nonce حتى تنتهي صلاحيته، بعد ذلك يقوم برفض أي رسائل جديدة بها نفس الـ nonce.

للقيام بالتوثيق فقط، تُستخدَم الرسالة ببساطة لتقديم أوراق الاعتماد إلى B. قد تتضمّن الرسالة معلومات أخرى يراد إيصالها (sgnData) والتي تُضمَّن أيضاً في مجال التوقيع مما يضمن أصالة الرسالة وسلامتها. وقد تُستخدَم الرسالة أيضاً لنقل مفتاح جلسة إلى B مُشفَّراً بمفتاح B العام.

❖ التوثيق مزدوج الاتجاه:

بالإضافة إلى العناصر الثلاثة التي ذُكِرت للتو (في التوثيق أحادي الاتجاه) يُرسِّخ التوثيق مزدوج الاتجاه العناصر الآتية أيضاً:

1. هوية B وأن رسالة الرد مصدرها B.
2. أن الرسالة موجهة إلى A.
3. سلامة رسالة الرد وأصالتها.

وهكذا يسمح التوثيق مزدوج الاتجاه لكل طرف من طرفي الاتصال بالتحقق من هوية الطرف الآخر.

تتضمّن رسالة الرد nonce من A يُستخدم في توثيقها. كما تتضمّن أيضاً خاتم الوقت (timestamp) وnonce من قِبَل B. وكما سبق قد تتضمّن الرسالة معلومات إضافية موقَّعة ومفتاح جلسة مُشفَّراً بمفتاح A العام.

❖ **التوثيق ثلاثي الاتجاه:**

بالإضافة للرسالتين السابقتين بالتوثيق مزدوج الاتجاه، يشمل التوثيق ثلاثي الاتجاه رسالة نهائية من A إلى B تتضمَّن نسخة مُوقَّعة من الـ nonce المشار له بـ r_B. يهدف هذا التصميم إلى عدم الحاجة لفحص أختام الوقت (timestamps)؛ وذلك لأن قيمة كل nonce يعاد إرسالها من قبل الطرف الآخر، ومن ثَمَّ يمكن لكل طرف أن يفحص قيمة الـ nonce المعاد لاكتشاف هجمات إعادة الإرسال. نحتاج لهذه الأسلوب للتوثيق عند عدم توفر ساعات متزامنة يمكن الاعتماد عليها (كما هو الحال في التوثيق مزدوج الاتجاه).

4-2-3 إصدار X.509 الثالث

لم تتضمن صيغة الإصدار الثاني لـ X.509 كل الجوانب التي أظهرت الخبرة الحديثة في التصميم والتنفيذ الحاجة إليها. يستعرض [FORD95] قائمة المتطلبات الآتية غير المتوفرة في الإصدار الثاني:

1. حقل الموضوع (Subject field) غير مناسب للتعبير عن هوية مالك المفتاح لمستخدم المفتاح العام. فقد تكون الأسماء في X.509 قصيرة نسبياً وتفتقر لتفاصيل التعريف الواضحة التي قد يحتاجها المستخدم.
2. أيضاً يُعدُّ حقل الموضوع غير كافٍ لكثير من التطبيقات، التي عادةً ما تتعرف على الكيانات المختلفة من عنوان البريد الإلكتروني أو عنوان URL أو أي وسيلة تعريف أخرى مرتبطة بالإنترنت.
3. هناك حاجة للإشارة إلى معلومات سياسة الأمن مما يُمكِّن تطبيقاً ما أو دالةً معينة للأمن (مثل IPSec) من ربط شهادة X.509 بسياسة معينة.

4. هناك حاجة لخفض الضرر الذي يمكن أن ينتج من وجود هيئة شهادات خبيثة أو بها خلل ما، وذلك بوضع القيود على مدى قابلية تطبيق شهادة معيّنة.
5. من المهم أن تتوفر القدرة على تمييز المفاتيح المختلفة المُستخدَمة من قِبَل نفس المالك في أوقات مختلفة. تدعم هذه الميزّة إدارة دورة حياة المفاتيح، وبالتحديد القدرة على تجديد أزواج المفاتيح للمستخدمين وهيئات الشهادات بشكلٍ منتظم أو في الظروف الاستثنائية.

بدلاً من مواصلة إضافة الحقول إلى صيغة ثابتة، أحس مطوّرو المعايير القياسية بالحاجة إلى أسلوب أكثر مرونة. ومن ثَمَّ تضمّن الإصدار الثالث عدداً من الامتدادات الاختيارية التي يمكن إضافتها لصيغة الإصدار الثاني. يتضمَّن كل امتداد: مُعرِّف امتداد (extension identifier)، ومؤشر للدرجة الحرجة (criticality indicator)، وقيمة للامتداد.

يبيّن مؤشر الدرجة الحرجة مدى إمكانية إهمال الامتداد بأمان. إذا كانت قيمة المؤشر TRUE ولم يتعرف تطبيق معين على الامتداد، فعليه أن يعامل الشهادة على أنها غير صحيحة.

تنتمي امتدادات الشهادة لثلاثة أصناف رئيسة: معلومات السياسة والمفتاح، وخواص المُصدِر والكائن، وقيود مسار التصديق.

❖ معلومات السياسة والمفتاح:

تحمل هذه الامتدادات معلومات إضافية حول مفاتيح المُصدِر والكائن، بالإضافة إلى مؤشرات سياسة الشهادة. وشهادة السياسة هي مجموعة مسمّاة من القواعد التي تشير إلى إمكانية استخدام الشهادة ضمن تجمُّع معيّن أو لنوعية معيّنة من التطبيقات بمتطلبات أمن مشتركة. على سبيل المثال، قد تكون السياسة قابلة للتطبيق لتوثيق مُعامَلات يتم فيها تبادل البيانات إلكترونياً (EDI) لتسويق السلع ضمن مدى معين من الأسعار.

تتضمّن هذه الفئة الحقول الآتية:

- مُعرِّف مفتاح الهيئة: يُحدد المفتاح العام الذي سيُستخدَم للتحقق من التوقيع على هذه الشهادة أو CRL. وهو يساعد على تمييز عدة مفاتيح لنفس الهيئة. أحد استخدامات هذا الحقل هو التعامل مع عمليات تحديث أزواج المفاتيح لـ CA.
- مُعرِّف مفتاح كائن: يُحدد المفتاح العام المراد التحقق منه. وهذا الحقل مفيد في تحديث أزواج المفاتيح للكائن. وقد يكون أيضاً للكائن عدة أزواج مفاتيح وبالتالي عدة شهادات مناظرة للأغراض المختلفة (مثلاً للتوقيع الرقمي وللاتفاق على مفتاح التشفير).
- استخدامات المفتاح: يشير إلى القيود المفروضة على الأغراض التي يمكن أن يُستخدَم فيها المفتاح العام الموثَّق أو المفروضة على السياسات التي يمكن أن يُستخدَم معها ذلك المفتاح. وقد يشير هذا الحقل إلى واحد أو أكثر من الأمور الآتية: التوقيع الرقمي، وعدم التنصل، وتشفير المفتاح، وتشفير البيانات، واتفاقية المفاتيح، والتحقق من توقيع CA على الشهادات، والتحقق من توقيع CA على CRLs.
- فترة استخدام المفتاح الخاص: تشير إلى الفترة التي يمكن أن يُستخدَم فيها المفتاح الخاص المناظر للمفتاح العام. وفي العادة يُستخدَم المفتاح الخاص لفترة مختلفة عن فترة صلاحية المفتاح العام. فعلى سبيل المثال، في حالة مفاتيح التوقيع الرقمي يُستخدَم المفتاح الخاص لفترة أقصر من فترة المفتاح العام المُستخدَم في التحقق.
- سياسات الشهادات: قد تُستخدَم الشهادات في بيئات يُطبَّق فيها أكثر من سياسة. يعرض هذا الامتداد قائمة بالسياسات التي تدعمها الشهادة بالإضافة إلى معلومات اختيارية عن مجال تطبيق الشهادة.
- تحويل السياسة: يُستخدَم فقط في الشهادات التي تُصدر لهيئات الشهادات من قِبَل هيئة شهادات أخرى. يسمح ذلك للهيئة التي تُصدر الشهادات

بالإشارة إلى إمكانية اعتبار سياسة أو أكثر من سياساتها مكافئة لسياسة أخرى مستخدمة في مجال هيئة الشهادات الخاصة بالكائن.

❖ سمات مُصدِر وكائن الشهادة:

تدعم هذه الامتدادات الأسماء البديلة بصيغ مختلفة لكائن الشهادة أو جهة إصدارها، ويمكن أن تتضمن معلومات إضافية عن كائن الشهادة لزيادة ثقة مستخدم الشهادة في أن كائن الشهادة هو شخص أو كيان معيّن. من أمثلة تلك المعلومات: العنوان البريدي للشخص، ورتبته في الشركة، وصورته.

تشمل حقول الامتداد في هذه الفئة ما يأتي:

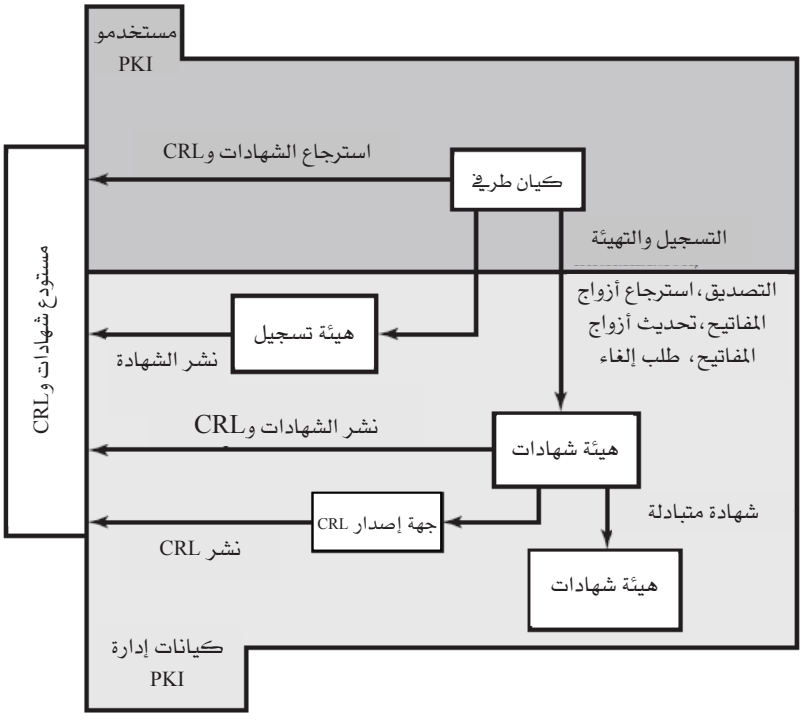
- الاسم البديل للكائن: يتضمن اسماً بديلاً أو أكثر بعدة صيغ مختلفة. هذا الحقل مهم لدعم بعض التطبيقات التي قد تستخدم صيغاً خاصة بها للأسماء (كالبريد الإلكتروني، وتبادل البيانات الإلكتروني (EDI)، وبروتوكول IPSec).
- الاسم البديل للمُصدِر: يتضمن اسماً بديلاً أو أكثر بعدة صيغ مختلفة.
- سمات دليل الكائن: يتضمن أي قيم مرغوب فيها لسمات كائن هذه الشهادة بدليل X.500.

❖ قيود مسار التصديق:

تسمح هذه الامتدادات بتضمين القيود الموجودة بالمواصفات في الشهادات التي تُصدرها هيئات شهادات لهيئات أخرى للشهادات. وقد تُحد تلك القيود من أنواع الشهادات التي يمكن أن تُصدَر من قِبَل هيئة الشهادات الخاصة بكائنٍ ما أو التي قد تستخدم بعد ذلك ضمن سلسلة توثيق.

تشمل حقول الامتداد في هذه الفئة ما يأتي:

- قيود أساسية: يشير إلى ما إذا كان الكائن يمكن أن يتصرف كـ CA. وإذا كان الأمر كذلك، فقد يتم تحديد قيود على طول مسار التصديق.
- قيود على الاسم: يشير إلى فضاء الاسم الذى ينتمي له اسم كل كائن في الشهادات المتتالية على مسار التصديق.
- قيود على السياسة: يحدد القيود التي قد تتطلّب تعريفاً صريحاً لسياسة الشهادات أو تمنع تحويل السياسة (policy mapping) على بقيّة مسار التصديق.



الشكل 4-7: نموذج بنية PKIX.

3-4 البنية التحتية للمفاتيح العامة

يُعرِّف مسرد مصطلحات أمن الإنترنت (RFC 2822) البنية التحتية للمفاتيح العامة (PKI) على أنها مجموعة الأجهزة والبرامج والأشخاص والسياسات والإجراءات اللازمة لإنشاء وإدارة وتخزين وتوزيع وإلغاء الشهادات الرقمية بشكلٍ يعتمد على التشفير غير المتماثل (التشفير بالمفاتيح العامة).

إنّ الهدف الرئيس من تطوير PKI هو توفير إمكانية للحصول على المفاتيح العامة بشكلٍ آمن ومريح وكفء. وكانت المجموعة الخاصة بالبنية التحتية لمفاتيح X.509 العامة والمنبثقة من فريق مهام هندسة الإنترنت (IETF) والمعروفة بـ PKIX القوة المحركة وراء إنشاء نموذج رسمي (وعام) يعتمد على X.509 ويناسب استخدام بنية الشهادات على الإنترنت. سنصف في هذا الجزء نموذج PKIX.

يبيّن الشكل 4-7 العلاقة البينية بين عناصر PKIX الرئيسة والتي تشمل:

- كيان نهاية: مصطلحٌ عام يُستخدَم للدلالة على المستخدمين الطرفيين أو الأجهزة (كالخادمات والموجهات) أو أي كيان آخر يُمكن تعريفه بحقل الكائن في شهادة المفتاح العام. في العادة تكون الكيانات الطرفية مستهلِكة للخدمات المتعلقة بـ PKI أو تكون موفِّرة لها أو قد تكون مستهلِكة وموفِّرة لها في ذات الوقت.
- هيئة الشهادات (CA): وتقوم بإصدار الشهادات وعادةً ما تُصدر قوائم الشهادات المُلغاة (CRL). وقد تدعم أيضاً تشكيلة من الوظائف الإدارية، رغم أنه في أغلب الأحيان يتم تفويض واحدة أو أكثر من هيئات التسجيل للقيام بتلك الوظائف.
- هيئة التسجيل (RA): عنصر اختياري قد يقوم بعدد من الوظائف الإدارية المخولَّة له من قِبَل CA. وفي أغلب الأحيان تُقرَن RA بعملية تسجيل الكيان الطرفي، لكن يمكن أن تُساعد في نواحٍ أخرى أيضاً.

- مُصدِر CRL: عنصر اختياري يمكن أن يُفوَّض من قِبَل CA لنشر قوائم الشهادات المُلغاة.
- مستودع: مصطلح عام يُستخدَم للدلالة على طريقة تخزين الشهادات وقوائم الشهادات المُلغاة حتى يُمكن استرجاعها من قِبَل الكيانات الطرفية.

1-3-4 وظائف PKIX الإدارية

يحدد PKIX عدداً من الوظائف الإدارية التي قد يتطلب الأمر دعمها من قِبَل بروتوكولات الإدارة كما هو مبيَّن بالشكل 4-7. وتشمل تلك الوظائف ما يأتي:

- التسجيل: وفيها يُعرِّف المُستخدِم نفسه في البداية لـ CA (بشكلٍ مباشر أو من خلال RA) قبل أن تُصدر له هيئة CA شهادة أو أكثر. يعدّ التسجيل بداية لعملية الإدراج ضمن PKI، ويتضمّن عادةً إجراءً ما للتوثيق المتبادل بشكلٍ متصل (online) أو منفصل (offline). وفي العادة يتم تزويد الكيان الطرفي بمفتاح سري مشترك أو أكثر للاستخدام في عمليات التوثيق اللاحقة.
- التهيئة: قبل أن يعمل نظام زبون بشكلٍ آمن، من الضروري تركيب مواد المفاتيح (key materials) ذات العلاقة المناسبة بالمفاتيح المخزَّنة في مكان آخر ضمن البنية التحتية. فعلى سبيل المثال، يحتاج الزبون إلى تزويده بالمفتاح العام والمعلومات الأخرى الموثَّقة من هيئة الشهادات المؤتمنة بشكلٍ آمن، وذلك لاستخدامها في التحقق من مسارات التصديق.
- التصديق: هذه هي العملية التي تُصدر فيها CA شهادة للمفتاح العام للمستخدم، وتُرسِلها إلى نظام زبون المستخدم و/أو تُرسِلها إلى المستودع.

- استعادة زوج المفاتيح: يُمكن أن استخدام زوج المفاتيح لدعم إنشاء التوقيع الرقمي والتحقق منه وأيضاً في التشفير وإزالة التشفير. وعندما يُستخدم زوج من المفاتيح في التشفير وإزالة التشفير، من المهم توفر آلية لاستعادة مفاتيح إزالة التشفير اللازمة عندما يتعذر الوصول الطبيعي لمواد المفاتيح، وإلا سيكون من المستحيل استعادة البيانات المشفَّرة. وقد يتعذر الوصول للمفاتيح بسبب نسيان كلمات السر، أو الرقم التعريفي للمستخدم (PIN)، أو حدوث تلف في مشغل الأقراص، أو غير ذلك. ويسمح إجراء استعادة زوج المفاتيح للكيانات الطرفية باستعادة زوج مفاتيح التشفير وإزالة التشفير الخاصة بهم من هيئة مرخص لها للنسخ الاحتياطي للمفاتيح (والتي عادة ما تكون CA التي أصدرت شهادة الكيان الطرفي).
- تحديث زوج المفاتيح: تحتاج كل أزواج المفاتيح إلى تحديث بشكلٍ منتظم (أي استبدالها بأزواج مفاتيح جديدة) وبالتالي لإعادة إصدار شهادات جديدة. يلزم التحديث عندما تنقضي فترة الصلاحية للشهادة أو كنتيجة لإلغاء الشهادة.
- طلب إلغاء: قد يُخبِر شخصٌ مخوّلٌ له CA بوجود حالة شاذّة تتطلّب إلغاء الشهادة. تشمل أسباب الإلغاء تسرب المفاتيح الخاصة، أو حدوث تغيير في الانتساب أو الاسم.
- التصديق المتقاطع: تقوم هيئتان للشهادات بتبادل معلومات تُستخدَم في تأسيس شهادة تقاطع؛ وهي شهادة تُصدر من CA إلى CA أخرى وتتضمن مفتاح توقيع CA المستخدم في إصدار الشهادات.

2-3-4 بروتوكولات إدارة PKIX

قامت مجموعة عمل PKIX بتعريف بروتوكولين بديلين للاستخدام بين كيانات PKIX يدعمان وظائف الإدارة التي استعرضناها سابقاً. يُعرِّف RFC 2510 بروتوكول إدارة الشهادة (Certificate Management Protocol (CMP))

والذي تُميَّز فيه كل وظيفة من وظائف الإدارة بشكلٍ واضح من خلال تبادلات محددة للبروتوكول. صُمِّم CMP ليكون نظاماً مرناً قادراً على دعم تشكيلة من النماذج التقنية والتشغيلية والتجارية.

أما RFC 2797 فيُعرِّف رسائل إدارة الشهادات على CMS (Certificate Management over CMS (CMC))؛ وCMS (Cryptographic Message Syntax) هو صيغة لرسائل التشفير مُعرَّفة في RFC 2630. بُنى CMC على عمل سابق ويهدف إلى زيادة فعالية التنفيذات الموجودة حالياً. ورغم دعم كل وظائف PKIX، لا يوجد تبادلات معينة للبروتوكول تناظر جميع الوظائف.

4-4 توصيات للمطالعة

لمراجعة مفاهيم كيربيروس بشكلٍ مبسّط يمكن الاطلاع على [BRYA88]. كما توجد معالجة جيدة لنظام كيربيروس في [KOHL94]. أما [TUNG99] فيصف كيربيروس من وجهة نظر المستخدم. كذلك يتضمن [PERL99] مراجعة للنماذج المؤتمنة المختلفة التي يمكن استخدامها في PKI. ويسلط [GUTM02] الضوء على صعوبات استخدام PKI، كما يقدم توصيات لبنية تحتية فعالة.

[BRYA88] Bryant, W. *Designing an Authentication System: A Dialogue in Four Scenes*. Project Athena document, February 1988.Available at http://web.mit.edu/kerberos/www/dialogue.html.

[GUTM02] Gutmann, P. "PKI: It's Not Dead, Just Resting." *Computer*, August 2002.

[KOHL94] Kohl, J.; Neuman, B.; and Ts'o, T. "The Evolution of the Kerberos Authentication Service." in Brazier, F., and Johansen, D. *Distributed Open Systems*. Los Alamitos, CA: IEEE Computer Society Press, 1994.Available at http://web.mit.edu/kerberos/www/papers. html.

[PERL99] Perlman, R. "An Overview of PKI Trust Models." *IEEE Network*, November/December 1999.

[TUNG99] Tung, B. *Kerberos: A Network Authentication System*. Reading, MA: Addison-Wesley, 1999.

5-4 مصادر للمعلومات على الويب

- موقع كيربيروس بمعهد MIT: يتضمَّن معلومات عن كيربيروس تشمل أسئلة متكررة وأوراقاً بحثية ومستندات وعناوين لمواقع منتجات تجارية.
- موقع كيربيروس بمعهد علوم المعلومات بجامعة كاليفورنيا الجنوبية: مصدر آخر جيد للمعلومات عن كيربيروس.
- مجموعة عمل كيربيروس: مجموعة عمل (جزء من IETF) لتطوير المعايير المبنية على كيربيروس.
- مجموعة عمل البنية التحتية للمفاتيح العامة: مجموعة عمل (جزء من IETF) لتطوير المعايير المبنية على X.509v3.
- موقع Verisign: إحدى الشركات الرائدة لمنتجات X.509، ويتضمَّن الموقع أوراقاً بحثية (white papers) ومواد أخرى.
- برنامج المعهد القومي للمعايير والتقنية (NIST) للبنية التحتية للمفاتيح العامة: مصدر جيد آخر للمعلومات.

6-4 مصطلحات رئيسة

authentication	توثيق، تحقق من الهوية، استيثاق
authentication server	خادم التوثيق
Kerberos system	نظام كيربيروس
Kerberos realm	عالَم كيربيروس
lifetime	فترة الصلاحية
nonce	عدد يُستخدَم لمرة واحدة
Propagating cipher block chaining (PCBC) mode	نمط سَلسَلة الكتل للشفرة المنتشرة
public-key certificate	شهادة المفتاح العام
realm	عالم
sequence number	رقم تسلسل
subkey	مفتاح فرعي
ticket	تذكرة
Ticket-Granting Server (TGS)	خادم منح التذاكر
X.509 certificate	شهادة X.509

7-4 أسئلة للمراجعة ومسائل

1-7-4 أسئلة للمراجعة

1-4 ما المشكلة التي صُمِّم كيربيروس لمعالجتها؟

2-4 ما التهديدات الثلاثة المرتبطة بالتحقق من هوية المستخدم على شبكة أو الإنترنت؟

3-4 اذكر ثلاث طرق لتأمين التحقق من هوية المستخدم في بيئة موزَّعة.

4-4 ما المتطلبات الأربعة المُعرَّفة بنظام كيربيروس؟

5-4 ما العناصر التي تُشكِّل بيئة خدمة كاملة لنظام كيربيروس ؟

6-4 في سياق كيربيروس ماذا يعني المصطلح realm؟

7-4 ما الفروق الأساسية بين إصدار كيربيروس الخامس والرابع؟

8-4 ما الهدف من معيار X.509؟

9-4 ماذا تعني سلسلة من الشهادات؟

10-4 كيف يتم إلغاء شهادة X.509؟

2-7-4 مسائل

1-4 بيِّن أن خطأً عشوائياً في كتلة ما من النص المُشفَّر ينتقل لجميع الكتل التالية من النص غير المُشفَّر في نمط PCBC.

2-4 افترض أنه تم تبديل موقعي الكتلتين C_i وC_{i+1} في نمط PCBC أثناء الانتقال. بيِّن أن هذا سيؤثر على الكتليتن الناتجتين من إزالة التشفير P_i وP_{i+1} فقط ولا يؤثر على الكتل التالية لهما.

3-4 يوجد خلل أمني بالإجراء الأصلي لتوثيق X.509 ثلاثي الاتجاه والموضح بالشكل 6-4 (c). جوهر البروتوكول كالآتي:

$$A \rightarrow B:\ A\{t_A, r_A, ID_B\}$$
$$B \rightarrow A:\ B\{t_B, r_B, ID_A, r_A\}$$
$$A \rightarrow B:\ A\{r_B\}$$

تذكر الوثيقة الخاصة بـ X.509 أن فحص خاتم الوقت t_A وt_B يعدّ أمراً اختيارياً للتوثيق ثلاثي الاتجاه. لكن بالنظر للمثال الآتي:

افترض أن A وB استخدما البروتوكول السابق في مناسبة سابقة معينة، وأن الخَصْم C اعترض طريق الرسائل الثلاث السابقة. إضافةً لذلك افترض أن قيم خاتم الوقت لم تُستخدَم وأن جميعها وضعت صفراً. وأخيراً افترض أن C أراد أن يتقمص شخصية A ويتصل بـ B. في البداية سيُرسِل C الرسالة الأولى التي التقطها إلى B:

$$C \rightarrow B\colon\ A\{0, r_A, ID_B\}$$

يرد B معتقداً أن A هو المُرسِل (لكن في الحقيقة هو يتحدث مع C):

$$B \rightarrow C\colon\ B\{0, r'_B, ID_A, r_A\}$$

في نفس الوقت يكون C بطريقةٍ ما قد جعل A يبدأ بالتوثيق مع C. ونتيجة ذلك سيُرسِل A إلى C ما يأتي:

$$A \rightarrow C\colon\ A\{0, r'_A, ID_C\}$$

يرد C على A مستخدماً نفس الـ nonce الذي وصل لـ C من B:

$$C \rightarrow A\colon\ C\{0, r'_B, ID_A, r'_A\}$$

يرد A بـ:

$$A \rightarrow C\colon\ A\{r'_B\}$$

هذا هو بالضبط ما يريده C ليقنع B أنه A، ولذا سيكرِّر الآن C الرسالة القادمة بإرسالها إلى B:

$$C \rightarrow B\colon\ A\{r'_B\}$$

وعليه سيعتقد B أنه يتحدث مع A في حين أنه في الحقيقة يتحدث مع C.

اقترح حلاً بسيطاً لتلك المشكلة والناتجة من عدم استخدام خاتم الوقت.

4-4 يعرض إصدار عام 1988 لـ X.509 الخصائص التي يجب أن تحققها مفاتيح RSA لكي تكون آمنة باستغلال ما هو معروف حالياً عن صعوبة تحليل الأعداد الكبيرة. تُختَتَم المناقشة بقيد على الأس المعلن وعملية باقي قسمة n: "يجب ضمان أن $e > \log_2(n)$ لمنع الهجوم من خلال حساب الجذر e ثم باقي قسمة n للحصول على النص غير المُشفَّر"

رغم أن هذا القيد صحيح إلا أن السبب المذكور لتطلّبه غير صحيح. فما هو الخطأ في السبب المُعطى وما هو السبب الحقيقي؟

الملحق A-4

طرق التشفير في نظام كيربيروس

يشتمل كيربيروس على مكتبة للتشفير تدعم مختلف العمليات المرتبطة بالتشفير. كانت هذه العمليات ضمن مواصفات إصدار كيربيروس الخامس كما يشيع استخدامها في المنتجات التجارية. في فبراير/شباط عام 2005 أصدرت IETF طلبات التعليقات RFC 3961 وRFC 3962 لتوسيع أساليب التشفير. في هذا الملحق سنصف أساليب كيربيروس بـ RFC 1510.

تحويل كلمات السرّ إلى مفاتيح

ينحصر استخدام كلمات السرّ في كيربيروس على الحروف التي يمكن تمثيلها بشفرة الآسكي بطول 7 بتات. هذه الكلمات بطول ما يتم تحويلها إلى مفاتيح تشفير تُخزَّن في قاعدة بيانات كيربيروس. يوضّح الشكل 4-8 ذلك الإجراء.

في البداية تحوَّل سلسلة الحروف s إلى سلسلة بتات b بحيث يُخزَّن أول حرف في أول 7 بتات، ويُخزَّن الحرف الثاني في البتات السبع التالية، وهكذا. يمكن التعبير عن ذلك رياضياً كما يأتي:

$$b[0] = \text{bit } 0 \text{ of } s[0]$$

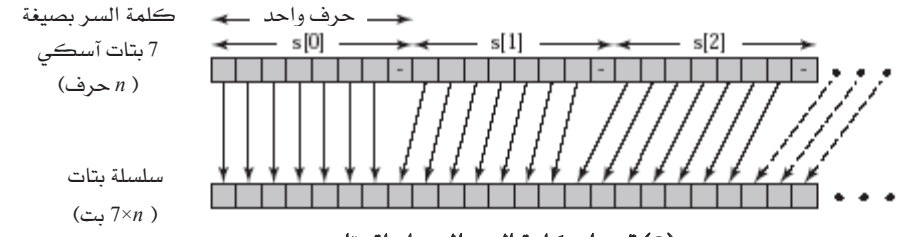
$$\dots$$

$$b[6] = \text{bit } 6 \text{ of } s[0]$$

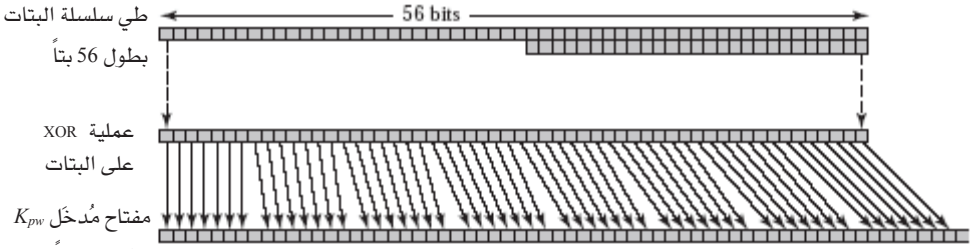
$$b[7] = \text{bit } 0 \text{ of } s[1]$$

$$\dots$$

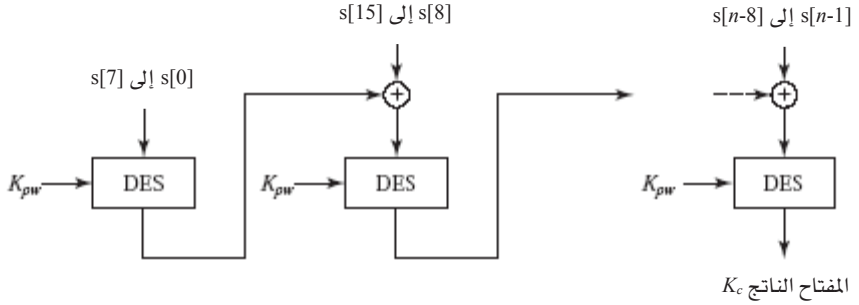
$$b[7i+m] = \text{bit } m \text{ of } s[i],\ 0 \le m \le 6$$



(a) تحويل كلمة السر إلى سلسلة بتات



(b) تحويل سلسلة بتات إلى مفتاح دخل



(c) توليد مجموع تدقيقي DES CBC لكلمة السر

الشكل 4-8: توليد مفاتيح التشفير من كلمات السرّ.

بعد ذلك يتم ضغط البتات لتكوين سلسلة طولها 56 بتاً فقط، وذلك بطي البتات على صفوف بطول 56 بتاً ثم حساب XOR على الأعمدة. على سبيل المثال إذا كان طول سلسلة البتات 59 بتاً، فعندئذ:

$$b[55] = b[55] \oplus b[56]$$

$$b[54] = b[54] \oplus b[57]$$
$$b[53] = b[53] \oplus b[58]$$

يُنشئ ذلك مفتاح DES بطول 56 بتاً. ولتحقيق الطول المطلوب (64 بتاً) للمفتاح K_{pw} تقسَّم سلسلة البتات الناتجة إلى كتل بطول 7 بتات وتُحوَّل كلٌّ منها إلى 8 بتات.

في النهاية تُشفَّر كلمة السرّ باستخدام نمط سَلسَلة كتل الشفرة (CBC) لخوارزمية DES مع المفتاح K_{pw}. تُعرف الـ 64 بتاً الأخيرة الناتجة من هذه العملية بالمجموع التدقيقي لـ CBC وهي تمثل المفتاح الناتج المرتبط بكلمة السرّ تلك. يمكن اعتبار الخوارزمية كلها كدالة تحوير تحوّل كلمة سرّ اعتباطية إلى كود تحوير بطول 64 بتاً.

نمط سلسلة كتل التشفير المنتشرة (PCBC)

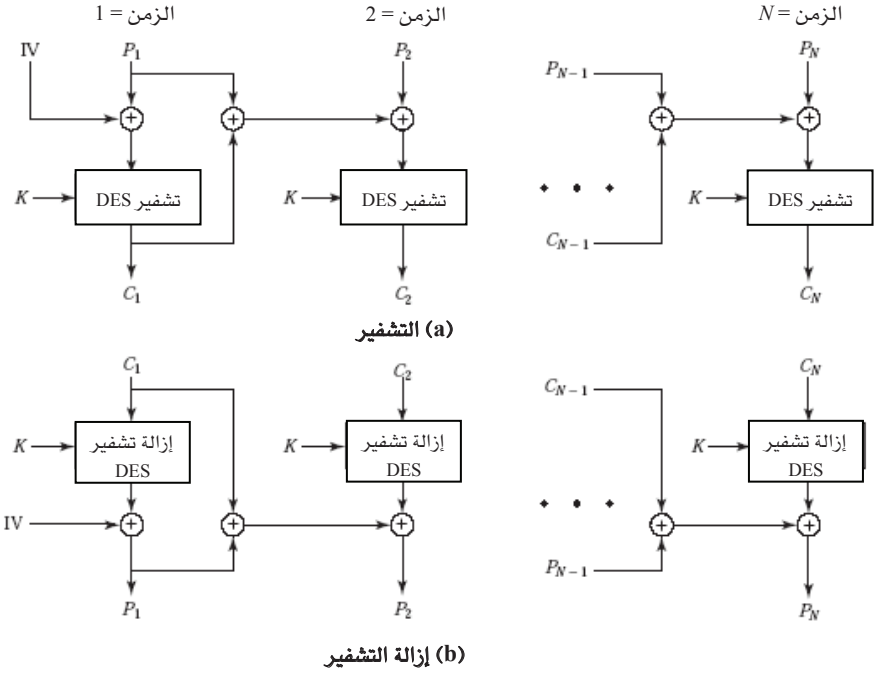
تذكَّر من الفصل الثاني أنه في نمط CBC لخوارزمية DES يتكون المُدخل في كل مرحلة من XOR للكتلة الحالية من النص غير المُشفَّر مع الكتلة السابقة من النص المُشفَّر، وأن نفس المفتاح يُستخدم في كل مرحلة (انظر الشكل 2-9). يتميز هذا النمط على نمط ECB (والذي يتم فيه تشفير كل كتلة بشكلٍ مستقل) بما يأتي: في CBC تُنتج نفس كتلة النص غير المُشفَّر في حالة تكررها كتلاً مُشفَّرةً مختلفةً.

من خصائص CBC أنه في حالة حدوث خطأ أثناء انتقال الكتلة المُشفَّرة فإنه سينتقل ليؤثر في جميع الكتل التالية من الرسالة بعد إزالة التشفير. ومن ثَمَّ يكون التشفير وسلامة البيانات قد تحققا في عملية واحدة (انظر المسألة 4-2 للاطلاع على استثناء لهذه الحالة). يوضّح الشكل 4-9 نمط PCBC. في هذه الطريقة يتكون المُدخل لخوارزمية التشفير من XOR لكتلة النص غير المُشفَّر الحالية وكتلة النص المُشفَّر السابقة وكتلة النص غير المُشفَّر السابقة:

$$C_n = \mathrm{E}(K, [C_{n-1} \oplus P_{n-1} \oplus P_n])$$

عند إزالة التشفير تُمرّر كل كتلة من النص المُشفَّر خلال خوارزمية إزالة التشفير. وبعد ذلك نجري عملية XOR للناتج مع كتلة النص المُشفَّر السابقة وكتلة النص غير المُشفَّر السابقة:

$$\mathrm{D}(K, C_n) = \mathrm{D}\big(K, \mathrm{E}(K, [C_{n-1} \oplus P_{n-1} \oplus P_n])\big)$$
$$\mathrm{D}(K, C_n) = C_{n-1} \oplus P_{n-1} \oplus P_n$$
$$C_{n-1} \oplus P_{n-1} \oplus \mathrm{D}(K, C_n) = P_n$$



الشكل 4-9: نمط سلسلة كتل التشفير المنتشرة (PCBC).

5

الفصل الخامس

أمن البريد الإلكتروني

محتويات الفصل:

5-1 بروتوكول الخصوصية الجيدة جداً (PGP)

5-1-1 الترقيم والاختصارات

5-1-2 وصف طريقة التشغيل

5-1-3 مفاتيح التشفير وحلقات المفاتيح

5-1-4 إدارة المفاتيح العامة

5-2 امتدادات بريد الإنترنت متعددة الأغراض الآمنة (S/MIME)

5-2-1 وثيقة طلب التعليقات RFC 822

5-2-2 امتدادات بريد الإنترنت متعددة الأغراض

5-2-3 وظائف S/MIME

5-2-4 رسائل S/MIME

5-2-5 معالجة الشهادات في S/MIME

5-2-6 خدمات الأمن المحسّنة

5-3 مصادر للمعلومات عى الويب

5-4 مصطلحات رئيسة

5-5 أسئلة للمراجعة ومسائل

الملحق A-5 ضغط البيانات باستخدام خوارزمية ZIP

الملحق B-5 التحويل للأساس 64 (Radix-64)

الملحق C-5 توليد الأعداد العشوائية في PGP

"على الرغم من رفض مساعد الأدميرال بوينديكستر واللفتنانت كولونيل نورث المثول أمام الهيئة، فإن الهيئة استطاعت ملء معظم ذلك الفراغ بفضل مصادر المعلومات الأخرى التي كانت لديها. قدَّم مكتب التحقيقات الفيدرالي (FBI) وثائق مأخوذة من ملفات مستشار الأمن القومي وموظفي مجلس الأمن القومي ذوي الصلة، بما في ذلك رسائل من نظام PROF بين مساعد الأدميرال بوينديكستر واللفتنانت كولونيل نورث. كانت تلك الرسائل محادثات عن طريق الحاسب كُتبت وقت وقوع الأحداث، وافترض كاتبوها أنها محمية من الكشف، ومن ثم فقد كانت توفر توثيقاً مباشراً لأحداث تلك الفترة."

ـــ من تقرير لجنة تاور للرئيس الأمريكي ريجان بخصوص قضية إيران كونترا، 1987.

"ليبارك الله الرجل الذي أنجزها ... وأدعو الله أنه لم يمت ... فقد كان بوسعه أن يكون من أصحاب الملايين ... لو أنه باعها لمكتب التحقيقات الفيدرالي ... لكنه كان متحمساً للحرية؛ فعرضها مجاناً للجمهور. الآن كل مواطن لديه PGP"

ـــ من أغنية "P.G.P"، ليزلي فيش.

النقاط الرئيسة

- يتألف بروتوكول الخصوصية الجيدة جداً ((Pretty Good Privacy (PGP من حزمة من البرمجيات المفتوحة المصدر (Open Source) والمتوفرة مجاناً لتوفير الأمن للبريد الإلكتروني. يوفر PGP التوثيق من خلال استخدام التوقيع الرقمي، والسرية من خلال استخدام التشفير المتماثل لكتل البيانات، وضغط البيانات باستخدام خوارزمية ZIP، والتوافق مع البريد الإلكتروني باستخدام نظام التكويد بالأساس 64 (radix-64)، كما يوفر وسائل للتقطيع (segmentation) وإعادة التجميع (reassembly) للتعامل مع رسائل البريد الإلكتروني الطويلة.
- يتضمن بروتوكول PGP أدواتٍ لتطوير نموذج للثقة وإدارة شهادات المفاتيح العامة.

- امتدادات بريد الإنترنت متعددة الأغراض الآمنة (S/MIME): هي طريقة قياسية لتوفير أمن البريد الإلكتروني وتتضمن نفس وظائف PGP.

في كل بيئات الأنظمة الموزعة تقريباً، يُعدُّ البريد الإلكتروني أكثر تطبيقات الشبكة استخداماً. كما يُعدُّ التطبيق الموزَّع الوحيد المُستخدَم على نطاق واسع في مختلف بنَى الأنظمة ومنصّات التشغيل؛ حيث يستطيع المستخدمون التراسل مع المستخدمين الآخرين المرتبطين بشبكة الإنترنت بشكلٍ مباشر أو غير مباشر بغض النظر عن نظام التشغيل أو بروتوكول الاتصالات المُستخدَم على المضيف.

مع النمو الهائل للاعتماد على البريد الإلكتروني في كل الأغراض التي يمكن تصورها، تزايد الطلب على خدمات التوثيق والسرية. يبرز أسلوبان في هذا المجال يتمتعان باستخدام واسع وسنتناولهما بالتفصيل في هذا الفصل: بروتوكول الخصوصية الجيدة جداً (PGP)، وامتدادات بريد الإنترنت متعددة الأغراض الآمنة (S/MIME).

1-5 بروتوكول الخصوصية الجيدة جداً (PGP)

يُعدُّ بروتوكول PGP ظاهرةً رائعة. فهو إلى حدٍّ كبيرٍ حصيلة جهود بذلها شخص بمفرده هو فيل زيمرمان (Phil Zimmermann). يوفر PGP خدمةً للسرية والتوثيق يمكن استخدامها مع تطبيقات البريد الإلكتروني وتخزين الملفات. باختصار، قام زيمرمان بالآتي:

- اختار نخبة من أفضل خوارزميات التشفير المتاحة واستخدمها كلبنات لبناء PGP.
- قام بتجميع تلك الخوارزميات بشكلٍ متكامل على هيئة تطبيق عام مستقل عن نظام التشغيل والمعالِج ويقوم على أساس مجموعة صغيرة من الأوامر سهلة الاستخدام.

- قام بتوفير حزمة البرمجيات وكل الوثائق اللازمة لها بما في ذلك برنامج المصدر والمتاح مجاناً عبر شبكة الإنترنت ولوحات الإعلانات الإلكترونية (bulletin boards) والشبكات التجارية كشبكة أميركا أون لاين (AOL).
- أبرم اتفاقاً مع شركة Viacrypt (الآن شركة Network Associates) لتوفير نسخة تجارية منخفضة الكلفة ومتوائمة تماماً مع PGP.

انتشر استخدام بروتوكول PGP بشكلٍ هائل، ويرجع ذلك إلى عدة أسباب يمكن الاستناد إليها لتفسير هذا النمو المضطرد:

1. PGP متاح مجاناً في جميع أنحاء العالم في إصداراتٍ تعمل على مجموعةٍ متنوعةٍ من المنصات تشمل نظم التشغيل ويندوز ويونيكس وماكنتوش وغيرها. بالإضافة إلى ذلك، فإن النسخة التجارية المتوفرة تلبي احتياجات المستخدمين الذين يرغبون في منتج يأتي مع دعمٍ من المورّد.
2. PGP مبنيٌ على خوارزميات خضعت لعمليات تمحيصٍ ومراجعةٍ شاملة من الجمهور وتُعدُّ آمنةً للغاية. بالتحديد تتضمن الحزمة خوارزميات RSA وDSS وديفي ـ هيلمان (Diffie-Hellman) للتشفير بالمفاتيح العامة، وCAST-128 وIDEA و3DES للتشفير المتماثل، وSHA-1 للتشفير بدوال التحوير (الهاش).
3. يغطي PGP مجموعةً واسعةً من مجالات الاستخدام، ابتداءً من الشركات التي ترغب في اختيار نظام قياسي واستخدامه لتشفير الملفات والرسائل وانتهاءً بالأفراد الذين يرغبون في التواصل مع الآخرين في جميع أنحاء العالم بشكلٍ آمنٍ على شبكة الإنترنت وغيرها من الشبكات.
4. لم يقم بتطوير PGP والسيطرة عليه أي منظمةٍ حكوميةٍ أو هيئة معايير، مما أضفى عليه جاذبية خاصة عند الذين يميلون بشكلٍ فطري لعدم الثقة في المؤسسات الكبيرة.

5. يأخذ PGP الآن طريقه ليصبح معيار إنترنت (RFC 3156). ومع ذلك، فلا تزال تحيط به هالة بوصفه كياناً "من خارج المؤسسة".

سنبدأ بإلقاء نظرةٍ عامةٍ على طريقة تشغيل بروتوكول PGP. وبعد ذلك سنشرح كيفية إنشاء مفاتيح التشفير وتخزينها، ثم نتناول المسألة الحيوية المتعلقة بإدارة المفاتيح العامة.

1-1-5 الترقيم والاختصارات

معظم الترقيمات والاختصارات الواردة في هذا الفصل قد استُخدمت من قبل، لكنك ستجد أيضاً بعض المصطلحات الجديدة. ولعله من الأفضل تلخيص تلك المصطلحات في البداية. فيما يلي قائمةٌ بالرموز المُستخدَمة:

K_s = مفتاح الجلسة المُستخدَم في نظام التشفير المتماثل

PR_a = المفتاح الخاص للمستخدم A، ويُستخدم في نظام التشفير بالمفاتيح العامة

PU_a = المفتاح العام للمستخدم A، ويُستخدم في نظام التشفير بالمفاتيح العامة

EP = التشفير بالمفاتيح العامة

DP = إزالة التشفير بالمفاتيح العامة

EC = التشفير المتماثل

DC = إزالة التشفير المتماثل

H = دالة التحوير (hash function)

|| = الوصل (concatenation)

Z = ضغط البيانات باستخدام خوارزمية ZIP

R64 = التحويل إلى صيغة آسكي بالأساس 64 (Radix-64 ASCII)

غالباً ما تستخدم وثائق PGP مصطلح "مفتاح سري" (secret key) للإشارة إلى مفتاح يقترن بمفتاح عام في نظام للتشفير بالمفاتيح العامة. كما ذكرنا آنفاً، قد

يؤدي ذلك الاستخدام إلى الخلط مع المفتاح السري المُستخدَم في نظام التشفير المتماثل. ولذا، فسوف نستخدم هنا مصطلح "مفتاح خاص" (private key) كبديل.

5-1-2 وصف طريقة التشغيل

يتضمن التشغيل الفعلي لبروتوكول PGP (إذا ما استبعدنا إدارة المفاتيح) خمس خدمات هي: التوثيق، والسرية، والضغط، والتوافق مع البريد الإلكتروني، والتقطيع (انظر الجدول 5-1). سوف نتناول كلاً من هذه الخدمات بالترتيب.

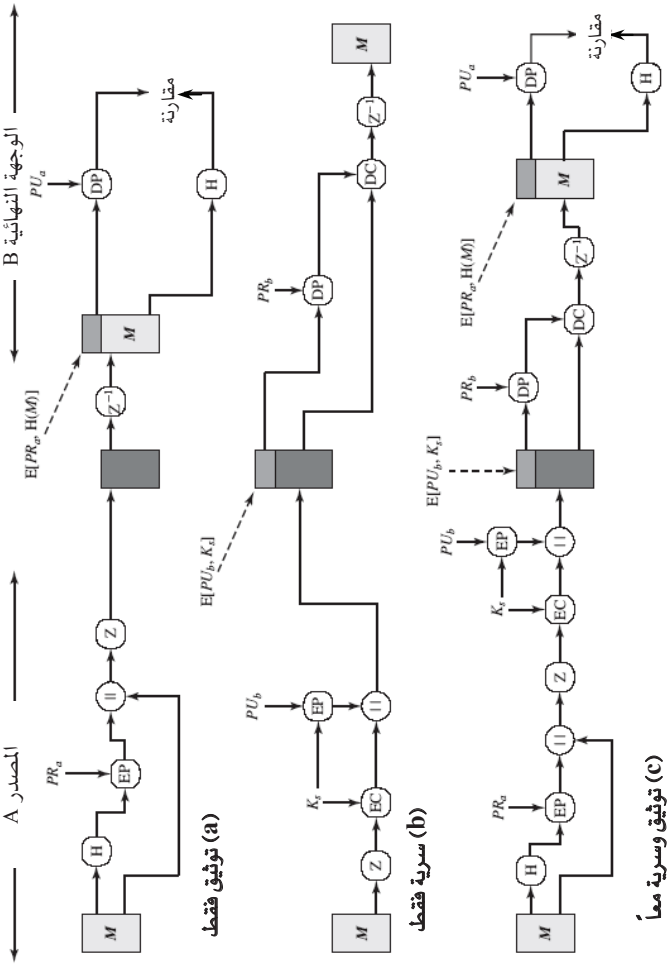
❖ التوثيق:

يوضّح الشكل 5-1 (a) خدمة التوقيع الرقمي في PGP. هذا هو نظام التوقيع الرقمي الذي ناقشناه في الفصل الثالث وبيّناه في الشكل 3-2 (b)، ويتم تنفيذه بالتسلسل الآتي:

1. ينشئ المُرسِل رسالة.
2. تُستخدَم خوارزمية SHA-1 لتوليد كود تحوير (hash code)، بطول 160 بتاً، للرسالة.
3. يتم تشفير كود التحوير بخوارزمية RSA باستخدام المفتاح الخاص للمُرسِل، وتُستهَل الرسالة بنتيجة التشفير.
4. يستخدم المُستلِم خوارزمية RSA مع المفتاح العام للمُرسِل لإزالة تشفير كود التحوير ومن ثم استعادة الكود.
5. يولِّد المُستلِم كود تحوير جديد للرسالة ويقارنه بكود التحوير الذي تم استعادته بإزالة التشفير. إذا حدث تطابق بين الكودين يتم قبول الرسالة على أنها موثَّقة المصدر.

الجدول 5-1: ملخص بخدمات بروتوكول PGP.

الخدمة	الخوارزميات المُستخدَمة	الوصف
التوقيع الرقمي	DSS/SHA أو RSA/SHA	يتم توليد كود تحوير باستخدام SHA-1. تُشفّر خلاصة الرسالة تلك بواسطة DSS أو RSA بالمفتاح الخاص للمُرسِل وتُلحَق بالرسالة.
تشفير الرسالة	CAST أو IDEA أو DES الثلاثية (بثلاثة مفاتيح) مع ديفي - هيلمان أو RSA	تُشفّر الرسالة بواسطة CAST-128 أو IDEA أو 3DES بمفتاح جلسة لمرة واحدة يقوم بتوليده المُرسِل. يتم تشفير مفتاح الجلسة باستخدام ديفي – هيلمان أو RSA بالمفتاح العام للمُستلِم وإلحاق المفتاح المُشفّر بالرسالة.
ضغط البيانات	ZIP	يمكن ضغط بيانات الرسالة بواسطة ZIP لأغراض الإرسال أو التخزين.
التوافق مع البريد الإلكتروني	التحويل إلى الأساس 64	لتحقيق الشفافية مع تطبيقات البريد الإلكتروني، يمكن تحويل الرسالة المُشفَّرة إلى صيغة آسكي باستخدام التحويل إلى الأساس 64.
التقطيع	- -	للتعامل مع القيد المتعلق بالحد الأقصى لطول رسالة البريد الإلكتروني، يقوم PGP بعمليتي التقطيع وإعادة التجميع.



الشكل 5-1: وظائف التشفير في PGP.

يوفر الجمع بين SHA-1 وRSA أسلوباً فعّالاً للتوقيع الرقمي. فنظراً لقوة خوارزمية RSA، يكون مُستلِم الرسالة مطمئناً إلى أن صاحب المفتاح الخاص الموائم (للمفتاح العام) هو فقط الذي يمكن أن يكون قد أنشأ التوقيع. وبسبب قوة خوارزمية SHA-1، يكون المُستلِم مطمئناً أيضاً إلى أنه لا يمكن أن يكون شخص آخر قد أنشأ رسالة جديدة توائم كود التحوير المُرسَل مع الرسالة الأصلية، ومن ثم لا يمكن أن يكون قد أنشأ التوقيع الرقمي الخاص بتلك الرسالة. يمكن أيضاً توليد التوقيعات باستخدام SHA-1 مع DSS بدلاً من RSA.

رغم وجود التوقيعات عادةً ملحقةً بالرسالة أو الملف الذي توقّعه، فإن هذا ليس هو الحال دائماً، فبروتوكول PGP يدعم أيضاً التوقيعات المنفصلة حيث يتم تخزين التوقيعات وإرسالها بمعزل عن الرسالة التي توقّعها، ويكون ذلك مفيداً في عدد من السيناريوهات. وقد يرغب المُستخدِم مثلاً في الاحتفاظ بسجل مستقل بالتوقيعات الخاصة بجميع الرسائل المُرسلَة أو الواردة. كما يمكن أن يكشف توقيع منفصل لبرنامج تنفيذي عن مدى إصابته لاحقاً بفيروس. وأخيراً، تُستخدم التوقيعات المنفصلة عندما يتعين توقيع الوثيقة من أكثر من طرف، كما في حالة العقود القانونية. يُعدُّ توقيع كل شخص توقيعاً مستقلاً، ومن ثم يُطبَّق على الوثيقة فقط. فيما عدا ذلك ستكون التوقيعات متراكبة (nested)، حيث يوقع الطرف الأول على الرسالة ويوقع الطرف الثاني على الوثيقة وعلى توقيع الطرف الأول، وهكذا.

❖ السرية:

خدمة أخرى من الخدمات التي يوفرها PGP، حيث يتم تشفير الرسائل التي سيتم إرسالها أو تخزينها محلياً في ملفات. في كلتا الحالتين يمكن استخدام خوارزمية CAST-128 للتشفير المتماثل. يمكن أيضاً استخدام IDEA أو 3DES بديلاً عن ذلك. يُستخدَم نمط التغذية المرتدَّة للشفرات ((cipher feedback (CFB بطول 64 بتاً.

كما هو الحال دائماً، لا بد من معالجة مشكلة توزيع المفاتيح. في PGP يُستخدَم كل مفتاح للتشفير المتماثل مرة واحدة فقط، حيث يتم توليد مفتاح جديد لكل رسالة كعدد عشوائي من 128 بتاً. وعليه، فرغم أن هذا المفتاح يُشار إليه في وثائق PGP كمفتاح جلسة، فإنه في الواقع مفتاحٌ يُستخدَم لمرة واحدة فقط. ونظراً لأنه لا يُستخدَم إلا مرة واحدة، فإن مفتاح الجلسة هذا يجب ربطه بالرسالة وإرساله معها. لحماية المفتاح يتم تشفيره بالمفتاح العام لمُستلِم الرسالة. ويوضّح الشكل 5-1 (b) تسلسل الخطوات التي يمكن تفصيلها على النحو الآتي:

1. يُولِّد المُرسِل رسالةً وعدداً عشوائياً بطول 128 بتاً لاستخدامه كمفتاح جلسة لتلك الرسالة فقط.
2. يتم تشفير الرسالة بخوارزمية CAST-128 (أو IDEA أو 3DES) باستخدام مفتاح الجلسة.
3. يُشفَّر مفتاح الجلسة بخوارزمية RSA باستخدام المفتاح العام للمُستلِم، وتُستهَل الرسالة بنتيجة التشفير.
4. يستخدم المُستلِم خوارزمية RSA ومفتاحه الخاص لإزالة تشفير مفتاح الجلسة، ومن ثم يحصل على مفتاح الجلسة.
5. يَستخدِم المُستلِم مفتاح الجلسة الناتج من الخطوة 4 لإزالة تشفير الرسالة.

يوفر PGP خياراً آخر يُعرَف بخوارزمية ديفي ـ هيلمان (Diffie-Hellman) بديلاً لاستخدام خوارزمية RSA لتشفير المفتاح.

إنَّ خوارزمية ديفي ـ هيلمان هي خوارزمية لتبادل المفاتيح مثلما تقدم في الفصل الثالث. في الواقع، يَستخدِم PGP نوعيةً من خوارزمية ديفي - هيلمان تُعرَف باسم الجَمَل (ElGamal) توفر كلاً من التشفير وإزالة التشفير (انظر التمرين 10-6).

يمكن إبداء عدة ملحوظات هنا. أولاً: للحد من زمن التشفير يتم الجمع ما بين التشفير المتماثل والتشفير بالمفاتيح العامة كحلٍ أفضل من مجرد استخدام

خوارزمية RSA أو الجَمَل لتشفير الرسالة مباشرةً. فخوارزمية CAST-128 وغيرها من الخوارزميات المتماثلة أسرع بكثير من RSA أو الجَمَل. ثانياً: يحل استخدام خوارزمية المفاتيح العامة مشكلة توزيع مفاتيح الجلسة، وذلك لأن المُستلِم فقط هو الوحيد الذي يكون بمقدوره استعادة مفتاح الجلسة المرتبط بالرسالة. لاحظ أيضاً أننا لسنا بحاجة إلى بروتوكول لتبادل مفاتيح الجلسة من النوع الذي ناقشناه في الفصل الثالث، لأننا لسنا بصدد بدء جلسة. وإنما تُعدُّ كل رسالة حدثاً مستقلاً قائماً بذاته لمرةٍ واحدةٍ له مفتاح الجلسة الخاص به. وعلاوةً على ذلك، نظراً لطابع "خزِّن وأرسِل" الذي يميز البريد الإلكتروني، فإن استخدام المصافحة لضمان أن كلا الجانبين لديه نفس مفتاح الجلسة ليس أمراً عملياً. وأخيراً، يعزز استخدام مفاتيح متماثلة لمرةٍ واحدةٍ أسلوباً قوياً أساساً هو أسلوب التشفير المتماثل. فكل مفتاحٍ يُستخدَم لتشفير كميةٍ ضئيلةٍ فقط من الرسالة، ولا توجد علاقةٌ بين المفاتيح المختلفة. وهكذا، فبالقدر الذي تكون به خوارزمية التشفير بالمفاتيح العامة آمنةً يكون النظام كلُّه آمناً. لهذه الغاية، يوفر PGP مجموعةً من الخيارات لأطوال المفتاح ابتداءً من 768 بتاً وحتى 3072 بتاً (في حين يقتصر طول مفتاح DSS للتوقيعات على 1024 بتاً).

❖ السرية والتوثيق:

يمكن استخدام هاتين الخدمتين لنفس الرسالة كما يتضح من الشكل 5-1 (c). أولاً: يتم إنشاء توقيع للرسالة غير المُشفَّرة وتُستهَل به الرسالة. ثم يتم تشفير الرسالة بعد إضافة التوقيع إليها باستخدام CAST-128 (أو IDEA أو 3DES)، ويُشفَّر مفتاح الجلسة باستخدام RSA (أو الجَمَل). يُعدُّ هذا التسلسل أفضل من التسلسل العكسي (أي تشفير الرسالة ثم توليد توقيع للرسالة المُشفَّرة). من المُسلَّم به عموماً أنه من الأكثر مناسبة تخزين التوقيع مع نسخة غير مُشفَّرة من الرسالة. وعلاوةً على ذلك، عند إجراء التحقق عن طريق طرف ثالث، إذا تمت إضافة التوقيع أولاً، فلن يكون الطرف الثالث بحاجة إلى أخذ المفتاح المتماثل بعين الاعتبار عند التحقق من التوقيع. باختصار، عند استخدام كلتا الخدمتين معاً، يوقع المُرسِل

الرسالة بمفتاحه الخاص قبل التشفير، ثم يُشفّر الرسالة بمفتاح الجلسة، وبعد ذلك يُشفر مفتاح الجلسة بالمفتاح العام للمُستلِم.

❖ ضغط البيانات:

في الوضع المعتاد، يضغط PGP بيانات الرسالة بعد إضافة التوقيع ولكن قبل التشفير. من فوائد ذلك التوفير في الحيز المطلوب لإرسال رسالة البريد الإلكتروني أو لتخزينها كملف. إن موضع خوارزمية الضغط التي نرمز لها في الشكل 5-1 بالرمز Z للضغط وZ^{-1} لإزالة الضغط، أمرٌ له أهميته:

1. يتم توليد التوقيع قبل الضغط لسببين:

a. من الأفضل توقيع رسالة غير مضغوطة بحيث نحتاج لتخزين الرسالة غير المضغوطة فقط مع التوقيع لأغراض التحقق في المستقبل. إذا وقَّع الشخص وثيقة مضغوطة فسيكون من الضروري تخزين نسخة مضغوطة من الرسالة للتحقق منها في وقت لاحق أو إعادة ضغط الرسالة عند الحاجة لإجراء عملية التحقق.

b. حتى لو كنّا على استعداد لإعادة ضغط الرسالة بشكلٍ ديناميكي لأغراض التحقق، فإن خوارزمية الضغط التي يستخدمها PGP تسبب بعض الصعوبات. فالخوارزمية ليست محددة (undeterministic)؛ والصيغ المختلفة لتنفيذها تحقق موازنات مختلفة بين سرعة التشغيل ونسبة ضغط البيانات، ولذلك تُنتَج أشكالاً مختلفة من الرسالة المضغوطة. ومع ذلك فكل خوارزميات الضغط المختلفة صالحة للعمل مع بعضها بعضاً، لأن كل صيغة للخوارزمية يمكنها إزالة الضغط الناتج من أي صيغة أخرى. إن تطبيق دالة التحوير والتوقيع بعد ضغط الرسالة من شأنه أن يقيد جميع تطبيقات PGP بحيث تستخدم نفس الإصدار من خوارزمية الضغط.

2. يتم تشفير الرسالة بعد ضغط بياناتها من أجل تعزيز أمن التشفير. فالرسالة المضغوطة تمتاز بتكرارية (redundancy) أقل من تكرارية الرسالة غير المُشفَّرة، مما يجعل عملية تحليل شفرتها أكثر صعوبة.

خوارزمية ضغط البيانات المُستخدَمة هي ZIP، وسنورد وصفاً لها في الملحق A-5.

❖ التوافق مع البريد الإلكتروني (Email Compatibility):

عند استخدام PGP، يتم تشفير جزءٍ على الأقل من كتلة البيانات التي سيتم إرسالها. إذا كنا نستخدم خدمة التوقيع فقط، فسيتم تشفير خلاصة الرسالة (باستخدام المفتاح الخاص للمُرسِل). وإذا كنا نستخدم خدمة السرية، فسيتم تشفير الرسالة بكاملها بالإضافة إلى التوقيع (إن وُجد) (باستخدام مفتاح متماثل يُستخدم لمرةٍ واحدة). وعليه، فإن جزءاً من كتلة البيانات الناتجة أو كلها سيتألَّف من سلسلة من وحداتٍ كلٌّ منها 8 بتات بقيم اعتباطية. غير أن كثيراً من أنظمة البريد الإلكتروني تسمح فقط باستخدام كتلٍ تتألف من نصٍّ بصيغة آسكي (ASCII) (حيث يُمثَّل كلُّ حرفٍ بـ 7 بتات). للتعامل مع هذا القيد، يوفر PGP خدمة لتحويل سلسلة البايتات الخام إلى سلسلة حروف آسكي.

الأسلوب المُستخدَم لهذا الغرض هو التحويل للأساس 64 (radix-64). وفيه يتم استبدال كلّ مجموعة من ثلاث بايتات من البيانات الثنائية بأربعة حروف آسكي مناظرة. يُلحِق هذا الأسلوب أيضاً كود فحص الفائض الدوري (Cyclic Redundancy Check (CRC) للكشف عن أخطاء الإرسال. انظر الملحق B-5 للاطلاع على وصفٍ أكثر تفصيلاً لهذا الأسلوب.

يؤدي استخدام التحويل للأساس 64 إلى تمديد طول الرسالة بنسبة 33%. ولحسن الحظ، فإن أجزاء مفتاح الجلسة والتوقيع تُعدُّ أجزاءً مختصرةً نسبياً من الرسالة، والرسالة غير المُشفَّرة تكون قد ضُغطت. في الواقع، ينبغي أن يكون الضغط أكثر من كافٍ لتعويض التمديد الحاصل للرسالة نتيجة التحويل للأساس

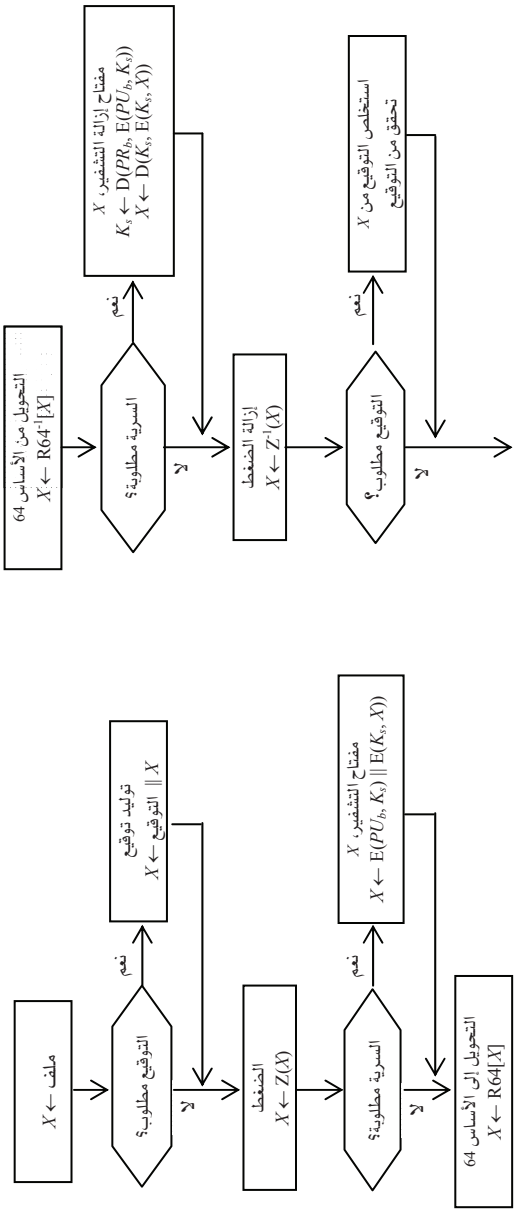
64. على سبيل المثال، يذكر [HELD96] أن متوسط نسبة الضغط باستخدام خوارزمية ZIP يبلغ حوالى 2.0. إذا أهملنا أجزاء التوقيع والمفتاح لصغرها نسبياً، فإن التأثير الكلي لضغط ملفٍ طوله X وإزالة ضغطه هو جعل طوله

$$1.33 \times 0.5 \times X = 0.665 \times X$$

ومن ثَمَّ لا يزال هناك انضغاط صافٍ بحوالي الثلث.

من الجدير بالذكر أن خوارزمية التحويل للأساس 64 تحوّل سلسلة البيانات الداخلة بشكلٍ أعمى إلى صيغة الأساس 64 بغض النظر عن المضمون، حتى لو حدث وكانت البيانات الداخلة في صورة نص آسكي في الأصل. ولذا، إذا تم توقيع رسالةٍ ولكنها لم تُشفَّر وتم إعمال التحويل على كتلة البيانات كلها، فسيكون الناتج غير مقروءٍ لمراقب عابر، مما يوفر قدراً معيّناً من السرية. كخيار، يمكن تهيئة PGP بحيث تقتصر في التحويل للأساس 64 على الجزء الخاص بالتوقيع في الرسائل الموقعة غير المُشفَّرة، مما يتيح للشخص المُستلِم قراءة الرسالة بدون استخدام PGP. يبقى من الضروري في تلك الحالة استخدام PGP للتحقق من التوقيع. ويبيّن الشكل 5-2 العلاقة بين الخدمات الأربع التي ناقشناها حتى الآن.

عند المُرسِل: يتم توليد توقيعٍ للرسالة إذا كان ذلك مطلوباً، وذلك باستخدام كود تحويرٍ للرسالة الأصلية غير المضغوطة. بعد ذلك يتم ضغط الرسالة الأصلية، بالإضافة إلى التوقيع إن وُجد. يلي ذلك، إذا كانت السرية مطلوبة، تشفير كتلة البيانات (التي تضم الرسالة الأصلية المضغوطة أو التوقيع المضغوط مضافاً إليه الرسالة الأصلية)، ويُلحق بها في البداية مفتاح التشفير المتماثل المُستخدَم في ذلك التشفير بعد تشفيره بطريقة المفاتيح العامة. أخيراً، يتم تحويل الكتلة الناتجة كلها للأساس 64.



الشكل 5-2: إرسال رسائل PGP واستقبالها.

عند المُستلِم: يتم أولاً تحويل كتلة البيانات الواصلة من الأساس 64 إلى الصيغة الثنائية. بعد ذلك، إذا كانت الرسالة قد شُفِّرت، يستعيد المُستلِم مفتاح الجلسة ويُزيل تشفير الرسالة. ثم يتم إزالة ضغط كتلة البيانات الناتجة. إذا كانت الرسالة موقَّعة، يستعيد المُستلِم كود التحوير المُرسَل ويقارنه بكود التحوير الذي يقوم بحسابه من الرسالة.

❖ **التقطيع وإعادة التجميع (Segmentation and Reassembly):**

غالباً ما تضع تسهيلات البريد الإلكتروني حداً أقصى لطول الرسالة المسموح بها. فعلى سبيل المثال، يفرض عددٌ من المرافق على شبكة الإنترنت حداً أقصى قدره 50000 بايت. يعني هذا أن أي رسالة يزيد طولها عن ذلك يجب أن تُقسَّم إلى قطع أصغر وتُرسل كلٌّ منها على حدة.

للتعامل مع هذا القيد، يقسّم PGP تلقائياً الرسالة الطويلة أكثر من اللازم إلى قطع تكون من الصغر بحيث يمكن إرسالها عبر البريد الإلكتروني. تتم عملية التقطيع تلك بعد الانتهاء من كل عمليات المعالجة الأخرى، بما في ذلك التحويل إلى الأساس 64. وعليه، فإن مفتاح الجلسة والتوقيع يظهران مرة واحدة فقط في الرسالة كلها، وذلك في بداية أول قطعة. ومن ثم يتعين على PGP عند الطرف المُستلِم نزع جميع ترويسات البريد الإلكتروني وإعادة تجميع الكتلة الأصلية قبل تنفيذ الخطوات المبينة في الشكل 5-2 (b).

5-1-3 مفاتيح التشفير وسلاسل المفاتيح

يستخدم PGP أربعة أنواع من المفاتيح: مفاتيح تشفير متماثل للاستخدام مرة واحدة، ومفاتيح عامة، ومفاتيح خاصة، ومفاتيح متماثلة مبنية على كلمة مرور (سنشرحها لاحقاً). ويمكن تحديد ثلاثة متطلبات مستقلة فيما يتعلق بتلك المفاتيح:

1. مطلوب وسيلة لتوليد مفاتيح جلسة لا يمكن التنبؤ بها.

2. نود أن يكون لدى المُستخدِم عدة أزواج من المفاتيح العامة والمفاتيح الخاصة، وذلك لأن المُستخدِم قد يرغب من حين لآخر في تغيير زوج المفاتيح المُستخدَم. وعندما يحدث ذلك، فإن أي رسائل تكون في طريقها للاستلام والمعالجة تكون قد تم تحضيرها بمفتاح متقادم. كما أن المُستلِمين يكونون على علم بالمفتاح العام القديم فقط إلى أن يصلهم التحديث المطلوب. علاوةً على الحاجة إلى تغيير المفاتيح مع مرور الوقت، فإن المُستخدِم قد يرغب في أن يكون بحوذته عدة أزواجٍ من المفاتيح في نفس الوقت للتفاعل مع مجموعاتٍ مختلفةٍ من المراسلين أو ببساطة لتعزيز الأمن عن طريق الحد من كمية البيانات المُشفَّرة بمفتاحٍ بعينه. نخلُص من ذلك إلى أنه لا يوجد تناظر واحد إلى واحد بين المستخدمين ومفاتيحهم العامة. ومن ثَمَّ، فهناك حاجة إلى وسائل للتعرف على مفاتيح بعينها.

3. يتعين على كل كيان PGP الاحتفاظ بملف خاص يضم أزواج المفاتيح العامة والمفاتيح الخاصة المناظرة لها، وكذلك المفاتيح العامة للمراسلين.

سنتناول كلاً من هذه المتطلبات تباعاً بشيء من التفصيل.

❖ توليد مفتاح الجلسة:

يرتبط كل مفتاح جلسة برسالةٍ واحدة، ويُستخدم فقط لغرض التشفير وإزالة التشفير لتلك الرسالة. لاحظ أن تشفير الرسائل وإزالة تشفيرها في PGP يتم باستخدام خوارزمية تشفير متماثل. تستخدم CAST-128 وIDEA مفاتيح بطول 128 بتاً، بينما تستخدم 3DES مفاتيح بطول 168 بتاً. سنفترض في المناقشة التالية استخدام خوارزمية CAST-128.

يتم توليد أعداد عشوائية بطول 128 بتاً بواسطة خوارزمية CAST-128 نفسها. يدخل إلى مولِّد الأعداد العشوائية مفتاحٌ طوله 128 بتاً وكتلتان طول كلٍّ منهما 64 بتاً يتم التعامل معهما كرسالة غير مُشفَّرة مطلوب تشفيرها. باستخدام نمط التشفير بالتغذية المرتدَّة (CFB) يُنتِج مُشفِّر CAST-128 كتلتين مُشفَّرتين يتم

وصلهما معاً لتكوين مفتاح الجلسة بطول 128 بتاً. الخوارزمية المُستخدَمة مبنية على أساس الخوارزمية الواردة في المعيار ANSI X12.17.

الرسالة الأصلية الداخلة إلى مولّد الأعداد العشوائية التي تتألف من كتلتين بطول 64 بتاً لكلٍّ منهما، هي نفسها مستمدة من سلسلة أعداد عشوائية بطول 128 بتاً. تعتمد تلك الأعداد على ضربات مفاتيح يقوم بها المُستخدِم. يُستخدَم كلٌّ من توقيت ضربات المفاتيح والمفاتيح التي يتم ضغطها في توليد سلسلة الأرقام العشوائية. فإذا كان المُستخدِم يضرب مفاتيح عشوائية بسرعته المعتادة فسيتم توليد أرقام ذات عشوائية معقولة. يتم أيضاً دمج هذا المُدخل العشوائي بمفتاح الجلسة السابق الذي أنتجته CAST-128 للحصول على مُدخَل المفتاح لمولّد الأعداد العشوائية. ونتيجةً لعملية الخلط الفعّال الذي تقوم به CAST-128، يتم إنتاج سلسلة من مفاتيح الجلسات لا يمكن التنبؤ بها عملياً.

يناقش الملحق C-5 أساليب توليد الأعداد العشوائية في PGP بمزيد من التفصيل.

❖ معرّفات المفاتيح:

كما أوضحنا آنفاً، يصحب كل رسالة مُشفَّرة تشفيرٌ لمفتاح الجلسة الذي استُخدِم في تشفير تلك الرسالة. يتم تشفير مفتاح الجلسة بالمفتاح العام للمُستلِم. وعليه، فإن المُستلِم فقط هو الذي يكون بإمكانه استعادة مفتاح الجلسة ومن ثَمَّ استعادة الرسالة. إذا استعمل كل مستخدم زوجاً واحداً من أزواج المفاتيح العامة والمفاتيح الخاصة، فسيعرف المُستلِم تلقائياً أي مفتاح يحتاجه لإزالة تشفير مفتاح الجلسة: ألا وهو المفتاح الخاص الوحيد الخاص بالمُستلِم. ولكنه كما ذكرنا ضمن المتطلبات أعلاه فإن كل مستخدم يمكن أن يكون لديه عدة أزواج من المفاتيح العامة والمفاتيح الخاصة التي يقوم باستعمالها.

كيف إذن يتسنى للمُستلِم معرفة المفتاح العام الذي استُخدم لتشفير مفتاح الجلسة؟ يتلخص أحد الحلول البسيطة في نقل المفتاح العام المُستخدَم لذلك مع الرسالة نفسها. يقوم المُستلِم عندئذٍ بالتحقق من أن المفتاح بالفعل هو أحد مفاتيحه العامة ويتابع العمل. هذا الحل سيعمل بنجاح، ولكنه سيبدد حيز الرسالة المرسلة بدون داعٍ. في خوارزمية RSA قد يصل طول المفتاح إلى مئات الخانات العشرية. يعتمد حلٌّ آخر على ربط كل مفتاح عام بمعرِّف فريد (على الأقل على مستوى كل مستخدم)، أي أن اسم المُستخدِم (user ID) ومعرّف المفتاح (key ID) يكونان كافيين معاً لتحديد مفتاحٍ ما من دون لبس. في هذا الحل نحتاج فقط لإرسال معرِّف المفتاح وهو أقصر بكثير من المفتاح نفسه. ومع ذلك فهذا الحل يثير مشاكل تتعلق بالأعباء الإدارية الإضافية التي يجلبها، حيث ينبغي تخصيص معرِّفات المفاتيح وتخزينها بحيث يمكن لكلٍّ من المُرسِل والمُستلِم معرفة المفتاح العام من معرِّف المفتاح. قد يبدو ذلك عبئاً لا مبرر له.

الحل الذي اعتمده بروتوكول PGP هو تخصيص معرِّف مفتاح لكل مفتاح عام (أي يكون فريداً من نوعه ضمن دائرة كل مستخدم وذلك باحتمالية كبيرة جداً).[1] يتكون معرِّف المفتاح المناظر لكل مفتاحٍ عامٍ من الـ 64 بتاً الأدنى وزناً (least significant). بمعنى أن معرِّف المفتاح PU_a هو ($PU_a \bmod 2^{64}$). يُعدُّ هذا طولاً كبيراً بما فيه الكفاية بحيث يجعل احتمال وجود معرِّف مزدوج ضئيلاً للغاية.

نحتاج أيضاً إلى معرِّف مفتاحٍ للتوقيع الرقمي في PGP. نظراً لأن المُرسِل قد يستخدم واحداً من عدة مفاتيح خاصة لتشفير خلاصة الرسالة، يحتاج المُستلِم لمعرفة المفتاح العام الذي ينبغي استخدامه. ومن ثمَّ يتضمن جزء التوقيع الرقمي في الرسالة معرِّف مفتاح من 64 بتاً يحدد المفتاح العام المطلوب. عند استلام الرسالة يتحقق

[1] مرّ بنا في السابق مقدمة عن مفاهيم الاحتمالات، وذلك في الجزء 8-3 لتحديد ما إذا كان عددٌ ما أولياً. أثناء تصميم الخوازميات، غالباً ما يؤدي استخدام أساليب الاحتمالات إلى الحصول على حلٍّ أقل تعقيداً، أو أقل استهلاكاً للوقت، أو كليهما.

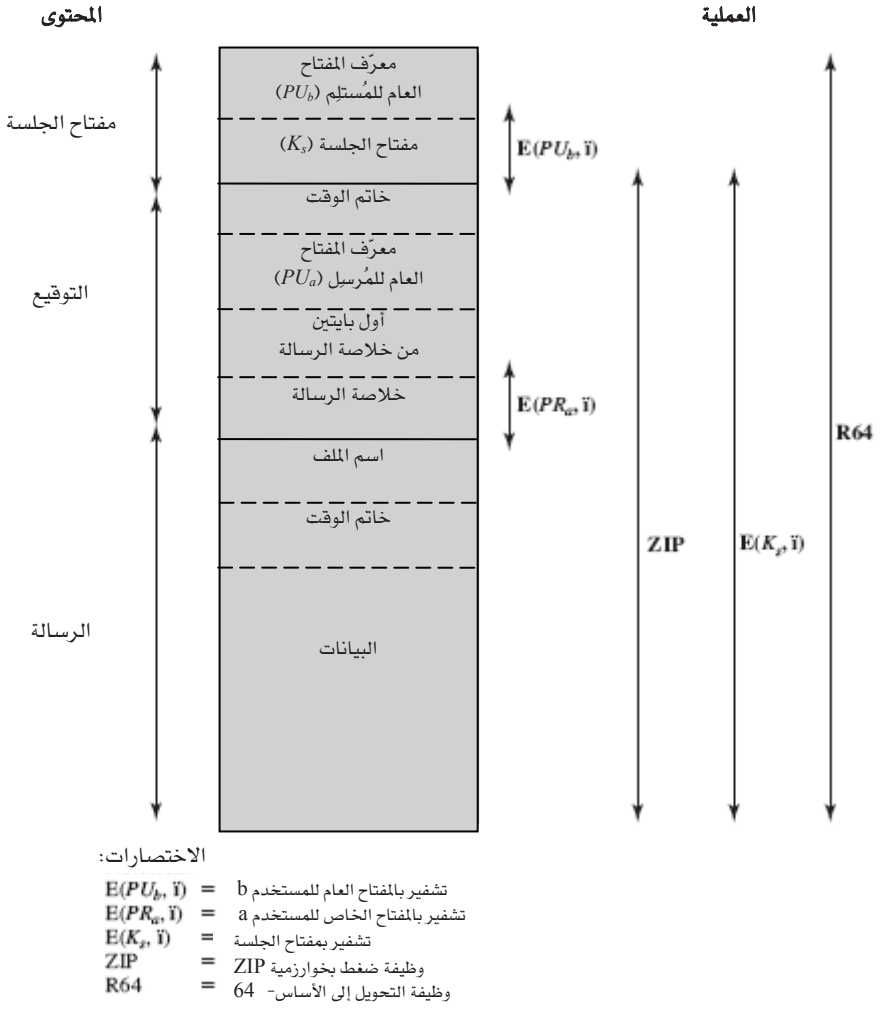
المُستلِم من أن معرِّف المفتاح يخص مفتاحاً عاماً يعرفه لذلك المُرسِل ثم يواصل عملية التحقق من التوقيع.

الآن وقد انتهينا من تقديم مفهوم معرّف المفتاح، بوسعنا إلقاء نظرة أكثر تفحصاً على صيغة الرسالة المُرسَلة والمبيّنة في الشكل 5-3. تتألف الرسالة من ثلاثة أجزاء: جزء الرسالة، وجزء التوقيع (اختياري)، وجزء مفتاح الجلسة (اختياري).

يشمل الجزء الخاص بالرسالة البيانات الفعلية المراد تخزينها أو نقلها، فضلاً عن اسم الملف وخاتم الوقت الذي يحدد وقت الإنشاء. أما الجزء الخاص بالتوقيع فيتضمن ما يلي:

1. خاتم الوقت: وقت توليد التوقيع.
2. خلاصة الرسالة (message digest): خلاصة SHA-1 بطول 160 بتاً، مُشفَّرة بمفتاح المُرسِل الخاص للتوقيع. تُحسب الخلاصة على خاتم الوقت ملحقاً به حقل البيانات في الجزء الخاص بالرسالة. يضمن إدراج حقل خاتم الوقت الخاص بالتوقيع في خلاصة الرسالة الحماية ضد الهجمات من نوع إعادة الإرسال (replay attacks). يضمن استبعاد حقل اسم الملف وحقل خاتم الوقت أثناء حساب خلاصة الرسالة أن تكون التوقيعات المنفصلة تماماً كالتوقيعات المتصلة التي تُلحق في بداية الرسالة. يتم حساب التوقيعات الرقمية المنفصلة على ملفٍ مستقلٍ بدون تضمين أيٍ من حقول ترويسة الرسالة.
3. أول بايتين من خلاصة الرسالة: لتمكين المُستلِم من التحقق مما إذا كان المفتاح العام الصحيح قد استُخدِم لإزالة تشفير خلاصة الرسالة، وذلك بمقارنة هذه النسخة غير المُشفَّرة من أول بايتين منها مع أول بايتين من الناتج من إزالة تشفير خلاصة الرسالة. يُستخدَم هذان البايتان أيضاً كتسلسل لفحص الإطار (frame check sequence) بطول 16 بتاً للرسالة.

4. معرِّف المفتاح العام للمُرسِل: يحدد المفتاح العام الذي ينبغي استخدامه لإزالة تشفير خلاصة الرسالة، ومن ثَمَّ يحدد المفتاح الخاص الذي استُخدِم لتشفير خلاصة الرسالة.



الشكل 5-3: الصيغة العامة لرسالة PGP (من A إلى B).

يمكن ضغط الجزء الخاص بالرسالة والجزء الاختياري الخاص بالتوقيع باستخدام خوارزمية ZIP، كما يمكن تشفيرهما باستخدام مفتاح الجلسة.

يتضمن الجزء الخاص بمفتاح الجلسة مفتاح الجلسة ومعرِّف المفتاح العام للمُستلِم الذي استخدمه المُرسِل لتشفير مفتاح الجلسة. عادةً ما يتم تمثيل الكتلة كلها بتكويد الأساس 64.

❖ حلقات المفاتيح

رأينا كيف تؤدي معرِّفات المفاتيح دوراً مهماً في تشغيل PGP، وأن كل رسالة PGP توفر كلاً من السرية والتوثيق تتضمن اثنين من معرِّفات المفاتيح. ينبغي تنظيم تلك المفاتيح وتخزينها بطريقة منهجية لضمان الاستخدام الكفء والفعّال لها من قِبَل جميع الأطراف المعنية. تتلخص الطريقة التي يستخدمها بروتوكول PGP في توفير زوجٍ من هياكل البيانات في كل عقدة، واحد لتخزين أزواج المفاتيح العامة والخاصة التي تمتلكها تلك العقدة، والآخر لتخزين المفاتيح العامة للمستخدمين الآخرين المعروفين لدى العقدة. يُطلَق على هياكل البيانات تلك حلقة المفاتيح الخاصة وحلقة المفاتيح العامة، على الترتيب.

يوضّح الشكل 5-4 الهيكل العام لحلقة مفاتيح عامة. يمكننا النظر إلى تلك الحلقة كجدول، يمثل كل صف فيه زوجاً من المفاتيح العامة والخاصة التي يمتلكها ذلك المُستخدِم. يحتوي كل صف في الجدول على البنود التالية:

- خاتم الوقت: تاريخ توليد هذا الزوج من المفاتيح ووقته.
- معرّف المفتاح: الـ 64 بتاً الأدنى وزناً في المفتاح العام ضمن هذا الزوج من المفاتيح.
- المفتاح العام: المفتاح العام ضمن هذا الزوج من المفاتيح.
- المفتاح الخاص: المفتاح الخاص ضمن هذا الزوج من المفاتيح، ويكون هذا الحقل مُشفَّراً.

حلقة المفاتيح الخاصة

اسم المستخدم *	المفتاح الخاص مشفراً	المفتاح العام	معرّف المفتاح*	خاتم الوقت
⋮	⋮	⋮	⋮	⋮
User i	$E(H(P_i), PR_i)$	PU_i	$PU_i \bmod 2^{64}$	T_i
⋮	⋮	⋮	⋮	⋮

حلقة المفاتيح العامة

النقة في التوقيعات	التوقيعات	شرعية المفتاح	اسم المستخدم *	ثقة المالك	المفتاح العام	معرّف المفتاح*	خاتم الوقت
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
		$trust_flag_i$	User i	$trust_flag_i$	PU_i	$PU_i \bmod 2^{64}$	T_i
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

* = المفتاح يُستخدم كمؤشر فهرس

الشكل 5-4: الهيكل العام لحلقات المفاتيح العامة والخاصة.

- اسم المُستخدِم: وعادةً ما يكون عنوان البريد الإلكتروني للمستخدِم (مثلاً stallings@acm.org)، ومع ذلك يمكن للمستخدِم ربط اسم آخر بزوج المفاتيح مثل: Stallings، وWStallings، وWilliamStallings، وهكذا. يمكن أيضاً استخدام نفس اسم المُستخدِم أكثر من مرة.

يمكن فهرسة حلقة المفاتيح الخاصة بناءً على اسم المُستخدِم أو معرّف المفتاح، وسنرى لاحقاً أمثلة للحاجة لكلتا الطريقتين للفهرسة.

رغم أن المقصود هو تخزين حلقة المفاتيح الخاصة فقط على الجهاز التابع للمستخدم الذي أنشأ أزواج المفاتيح ويملكها، وأن يُقصَر الوصول إليها على ذلك

المُستخدِم فقط، فإنه من المنطقي جعل قيمة المفتاح الخاص آمنة بأكبر قدرٍ ممكن. وعليه، لا يتم تخزين المفتاح الخاص نفسه في حلقة المفاتيح، ولكن يخزّن ذلك المفتاح بعد تشفيره باستخدام CAST-128 (أو IDEA أو 3DES). يتلخص الإجراء المُستخدَم في الخطوات الآتية:

1. يختار المُستخدِم كلمة المرور التي ستُستعمل لتشفير المفاتيح الخاصة.
2. عندما يولّد النظام زوجاً جديداً من المفاتيح العامة والخاصة بواسطة خوارزمية RSA، فإنه يطلب من المُستخدِم إدخال كلمة المرور. يتم توليد كود تحوير طوله 160 بتاً لكلمة المرور باستخدام SHA-1، ثم التخلص من كلمة المرور الأصلية.
3. يقوم النظام بتشفير المفتاح الخاص بواسطة CAST-128 مستخدماً 128 بتاً من كود التحوير كمفتاح. بعد ذلك يتم التخلص من كود التحوير. يتم تخزين المفتاح الخاص المُشفَّر في حلقة المفاتيح الخاصة.

في وقت لاحق عندما يقوم المُستخدِم بالدخول على حلقة المفاتيح الخاصة لاستعادة مفتاحٍ خاص، سيتعين عليه إدخال كلمة المرور. يقوم PGP باستعادة المفتاح الخاص المُشفَّر، وتوليد كود التحوير لكلمة المرور، ثم إزالة تشفير المفتاح الخاص بواسطة CAST-128 مع كود التحوير. يُعدُّ هذا نظاماً مختصراً وفعّالاً للغاية. كما هو الحال في أي نظام يقوم على كلمات السر، يعتمد أمن النظام على أمن كلمة السر. لتجنب الاحتفاظ بها مكتوبة، ينبغي على المُستخدِم استعمال كلمة مرور يسهل تذكرها ولكن يصعب تخمينها. ويبيّن الشكل 5-4 أيضاً الهيكل العام لحلقة مفاتيح عامة. يُستخدَم هيكل البيانات هذا لتخزين المفاتيح العامة للمستخدمين الآخرين المعروفين لدى ذلك المُستخدِم. لنتجاهل مؤقتاً بعض الحقول المبينة في الجدول ونركّز على وصف الحقول الآتية:

1. خاتم الوقت: تاريخ توليد هذا المُدخل ووقته.
2. مُعرِّف المفتاح: ويتألف من الـ 64 بتاً الأدنى وزناً في المفتاح العام لهذا المُدخل.

3. المفتاح العام: المفتاح العام لهذا المُدخل.
4. اسم المُستخدِم: يُعرِّف مالك هذا المفتاح؛ حيث يمكن ربط عدة أسماء مستخدمين بنفس المفتاح العام.

يمكن فهرسة حلقة المفاتيح العامة بناءً على اسم المُستخدِم أو مُعرِّف المفتاح؛ وسنرى لاحقاً أمثلة للحاجة لكلتا الطريقتين للفهرسة. نحن الآن في وضع يسمح لنا بتوضيح كيفية استخدام حلقات المفاتيح تلك أثناء إرسال الرسالة واستقبالها. للتبسيط سنهمل عمليتي ضغط البيانات والتحويل للأساس 64 في المناقشة التالية. لنأخذ في الاعتبار أولاً عملية نقل الرسالة (الشكل 5-5) ونفترض أن الرسالة ينبغي توقيعها وتشفيرها. يقوم كيان PGP المُرسِل بالخطوات الآتية:

1. توقيع الرسالة

a. يستخرج بروتوكول PGP المفتاح الخاص للمُرسِل من حلقة المفاتيح الخاصة باستخدام your_userid كمؤشر للفهرس. إذا لم يتم توفير your_userid في أمر العملية فسيتم استخراج أول مفتاح خاص في حلقة المفاتيح.
b. يطلب PGP من المُستخدِم إدخال كلمة المرور لاستخراج المفتاح الخاص غير المُشفَّر.
c. يتم إنشاء الجزء الخاص بالتوقيع في الرسالة.

2. تشفير الرسالة

a. يولِّد PGP مفتاح جلسة ويستخدمه لتشفير الرسالة.
b. يستخرج PGP المفتاح العام للمُستلِم من حلقة المفاتيح الخاصة باستخدام her_userid كمؤشر للفهرس.
c. يتم إنشاء الجزء الخاص بمفتاح الجلسة في الرسالة.

يقوم كيان PGP المُستلِم بالخطوات الآتية (الشكل 5-6):

1. إزالة تشفير الرسالة

a. يستخرج PGP المفتاح الخاص للمُستلِم من حلقة المفاتيح الخاصة، وذلك باستخدام حقل مُعرِّف المفتاح في جزء مفتاح الجلسة ضمن الرسالة كمؤشر للفهرس.

b. يطلب PGP من المُستخدِم إدخال كلمة المرور لاستعادة المفتاح الخاص غير المُشفَّر.

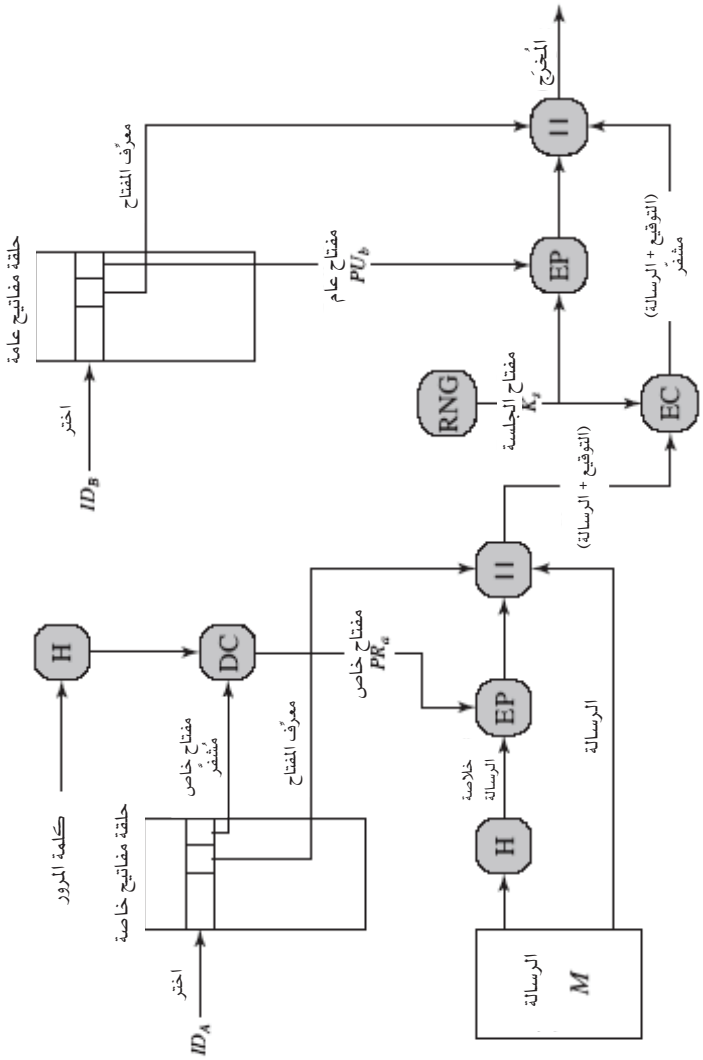
c. عندئذٍ يستخرج PGP مفتاح الجلسة ويستخدمه لإزالة تشفير الرسالة.

2. توثيق الرسالة

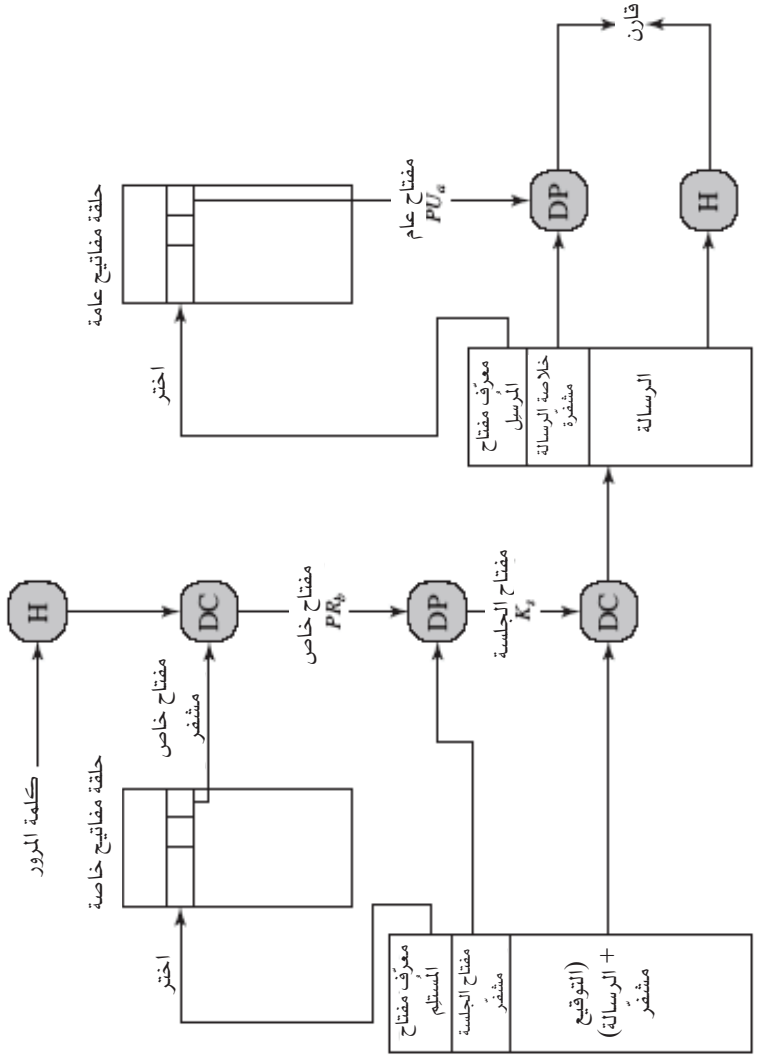
a. يستخرج PGP المفتاح العام للمُستلِم من حلقة المفاتيح العامة، وذلك باستخدام حقل مُعرِّف المفتاح في جزء مفتاح التوقيع ضمن الرسالة كمؤشر للفهرس.

b. يستعيد PGP خلاصة الرسالة المُستلَمة.

c. يحسب PGP خلاصة الرسالة التي تم استلامها ويقارن النتيجة بخلاصة الرسالة المتضمنة في الرسالة المُستلَمة لإتمام عملية التوثيق.



الشكل 5-5: توليد الرسالة في PGP (من المُستخدم A إلى المُستخدم B؛ بدون ضغط للبيانات أو تحويلها للأساس 64).



الشكل 5-6: استقبال الرسالة في PGP (من المُستخدم A إلى المُستخدم B؛ بدون ضغط للبيانات أو تحويلها للأساس 64).

5-1-4 إدارة المفاتيح العامة

كما يتضح من المناقشة حتى الآن، يتضمن PGP مجموعة فعّالة وذكية من الوظائف والصيغ المتشابكة التي تتضافر فيما بينها لتوفير خدمة فعالة للخصوصية والتوثيق. لاستكمال الحديث عن النظام، نحتاج لتناول جزء مهم وأخير، ألا وهو إدارة المفاتيح العامة. تجسد وثائق PGP أهمية هذا الموضوع في العبارة الآتية:

> يُعدُّ هذا الأمر برمّته ـ والخاص بحماية المفاتيح العامة من العبث بها ـ أحد أصعب المشاكل في التطبيقات العملية للمفاتيح العامة. إنه نقطة الضعف الأساسية في أنظمة التشفير بالمفاتيح العامة، ويرتبط الكثير من التعقيد البرمجي في تلك الأنظمة بحل تلك المشكلة.

يوفر بروتوكول PGP إطاراً لحل تلك المشكلة، ويوجد عدد من الخيارات المقترحة التي يمكن استخدامها. ولكن لمّا كان الهدف هو الاستفادة من PGP في تشكيلة متنوعة من البيئات الرسمية وغير الرسمية، فقد حال ذلك دون اتباع خطة جامدة لإدارة المفاتيح العامة، كتلك التي تم اتباعها مع بروتوكول S/MIME الذي سنتناوله لاحقاً في هذا الفصل.

❖ طرق إدارة المفاتيح العامة:

يكمن جوهر المشكلة في أنه يتعين على المُستخدِم A تكوين حلقة مفاتيح تضم المفاتيح العامة للمستخدمين الآخرين للتعامل معهم باستخدام PGP. لنفترض أن حلقة المفاتيح لدى A تتضمن مفتاحاً عاماً منسوباً لـ B ولكنه مملوكٌ في حقيقة الأمر لـ C. يمكن أن يحدث ذلك، على سبيل المثال، إذا حصل A على مفتاح من نظام لوحة إعلانات (bulletin board system (BBS)) كان قد استخدمه B لوضع المفتاح العام ولكن C تمكّن من الاستحواذ على ذلك المفتاح. النتيجة أننا الآن نواجه تهديدين: الأول هو أنه بإمكان C إرسال رسائل إلى A وتزييف توقيع B؛

بحيث يقبل A الرسالة على أنها قادمة من B. والثاني أن C سيكون بوسعه قراءة أي رسالة مُشفَّرة من A إلى B.

هناك عدة طرق يمكن استخدامها لتقليل خطر احتواء حلقة المفاتيح العامة لمُستخدِمٍ ما على مفاتيح زائفة. لنفترض أن A يرغب في الحصول على مفتاح عام موثوق لـ B. فيما يلي بعض الطرق الممكنة لتحقيق ذلك:

1. الحصول على المفتاح مادياً من B. بوسع B تخزين مفتاحه العام (PU_b) على قرص مرن وتسليمه إلى A. يمكن لـ A بعد ذلك تحميل المفتاح من القرص المرن إلى نظامه. تُعدُّ هذه وسيلةً آمنةً جداً ولكن واضحٌ أنها غير عملية.
2. التحقق من المفتاح عن طريق الاتصال بالهاتف. إذا كان بوسع A الحصول على رقم هاتف B، يمكنه أن يتصل به هاتفياً ويطلب منه إملاء المفتاح عليه بصيغة الأساس 64. كبديل عملي أكثر، يمكن لـ B إرسال مفتاحه إلى A في رسالة بريد إلكتروني. بوسع A بعد ذلك توليد خلاصة للمفتاح (key digest) بطول 160 بتاً باستخدام SHA-1 وعرضها بصيغة ست – عشرية (hexadecimal)، ويطلق على خلاصة المفتاح تلك "بصمة المفتاح". يمكن لـ A عندئذٍ الاتصال هاتفياً بـ B ويطلب منه أن يملي عليه بصمة المفتاح على الهاتف. فإذا تطابقت البصمتان فهذا يعني التحقق من صحة المفتاح.
3. الحصول على المفتاح العام لـ B من شخص موثوق فيه لدى كلا الطرفين مثل D (المُوثَّق). لهذا الغرض، يقوم D بإنشاء وثيقةٍ موقَّعة. تضم تلك الوثيقة الموقَّعة مفتاح B العام، ووقت إنشاء المفتاح، وفترة صلاحية المفتاح. يولّد D خلاصة لتلك الشهادة باستخدام SHA-1، ويشفرها بمفتاحه الخاص ويُلحق التوقيع بالشهادة. ونظراً لأن D فقط هو الذي يمكن أن يكون قد ولَّد التوقيع، لا يمكن لشخص آخر أن ينشئ مفتاحاً عاماً مزيفاً ويدعي أنه موقّع من قِبَل D. يمكن إرسال الشهادة الموقَّعة مباشرةً إلى A من B أو D، أو يمكن وضعها على لوحة إعلانات.

4. الحصول على المفتاح العام لـ B من سلطة تصديق موثوقة. مرة أخرى، تقوم تلك السلطة بإصدار شهادة مفتاح عام وتوقيعها. بوسع A بعد ذلك الاتصال بتلك السلطة وتقديم اسم مستخدم، وبناءً عليه يتلقى شهادة موقّعة.

في الحالة الثالثة والرابعة، سيتعيّن على A أن يكون بحوزته نسخةٌ من المفتاح العام للمُوثِّق، وأن تكون لديه الثقة في أن ذلك المفتاح صحيح. في نهاية المطاف، يرجع الأمر إلى A في تعيين مستوى الثقة المطلوب في الشخص الذي يمكن أن يقوم بمهمة المُوثِّق.

❖ استخدام الثقة

رغم أن بروتوكول PGP لم يشتمل على أي تحديدٍ لكيفية إنشاء سُلطات التصديق أو بناء الثقة، فإنه يوفر وسيلة ملائمة لاستخدام الثقة وربطها بالمفاتيح العامة وكذلك استغلال معلومات الثقة.

فيما يلي وصفٌ للهيكل الأساسي. يُعدُّ كل مُدخل في حلقة المفاتيح العامة بمثابة شهادة على النحو المبين في الجزء السابق. يرتبط بكل مُدخل حقلٌ لشرعية المفتاح (key legitimacy field) يبيّن إلى أي مدى سيكون PGP على ثقة من صحة هذا المفتاح العام لذلك المُستخدِم. فكلّما كان مستوى الثقة عالياً، كان الارتباط أقوى بين اسم المُستخدِم والمفتاح. يقوم PGP بحساب قيمة ذلك الحقل. يرتبط بكل مُدخل أيضاً صفر أو أكثر من التوقيعات على تلك الشهادة التي جمعها صاحب حلقة المفاتيح. وبدوره، يرتبط بكل توقيع حقلٌ للثقة في ذلك التوقيع (signature trust field) يُشير إلى الدرجة التي يثق بها هذا المُستخدِم لـ PGP في الجهة الموقّعة بوصفها جهة مخوَّلة بتصديق المفاتيح العامة. يتم حساب حقل شرعية المفتاح من مجموع حقول الثقة في التوقيعات لذلك المُدخل. وأخيراً، يُعرِّف كلّ مُدخل مفتاحاً عاماً مرتبطاً بمالك معين، ويتم تضمين حقل للثقة في المالك (owner trust field) يبيّن درجة الثقة في مالك ذلك المفتاح العام بوصفه جهة مخوَّلة بتوقيع شهادات

المفاتيح العامة الأخرى؛ يتم تحديد مستوى الثقة هذا عن طريق المُستخدِم. يمكننا اعتبار حقول الثقة في التوقيع نسخاً مخبأة (cached copies) من حقل الثقة في المالك من مُدخل آخر.

يُوجد كلٌّ من الحقول الثلاثة المذكورة في الفقرة السابقة في هيكل يُعرف ببايت عَلَم الثقة (trust flag byte). ويبيّن الجدول 5-2 محتوى بايت عَلَم الثقة هذا لكلٍّ من تلك الاستخدامات الثلاثة. لنفترض أننا نتعامل مع حلقة المفاتيح العامة للمستخدم A. يمكننا عندئذٍ وصف عمليات التعامل مع الثقة على النحو الآتي:

1. عندما يقوم A بإدراج مفتاحٍ عامٍ جديدٍ في حلقة المفاتيح العامة، يجب على PGP تعيين قيمة لعَلَم الثقة ترتبط بمالك ذلك المفتاح العام. إذا كان المالك هو A – ومن ثَمَّ يظهر هذا المفتاح العام أيضاً في حلقة المفتاح الخاصة – فإن حقل الثقة يأخذ تلقائياً القيمة "ثقة متناهية" (ultimate trust)، وإلا فإن PGP يسأل A عن تقييمه للثقة التي ستخصص لمالك ذلك المفتاح، وعلى A أن يُدخل المستوى المطلوب. يمكن للمستخدم الرد بأن ذلك المالك غير معروف، أو غير موثوق، أو موثوق لحدٍّ ما، أو أنه موثوق تماماً.
2. عند إدخال مفتاحٍ عامٍّ جديد، يمكن أن يُلحَق به توقيع أو أكثر. ويمكن إضافة توقيعات جديدة فيما بعد. عند إضافة توقيعٍ إلى مُدخل، يقوم PGP بالبحث في حلقة المفاتيح العامة لتحديد ما إذا كان مؤلِّف هذا التوقيع من بين مالكي المفتاح العام المعروفين. إذا كان الأمر كذلك يُعطَى حقل SIGTRUST نفس قيمة OWNERTRUST الخاصة بذلك المالك. وإلا فتُخصص له القيمة "مستخدم غير معرَّف" (unknown user).
3. يُحسَب حقل شرعية المفتاح على أساس قيم حقول الثقة في التوقيعات الموجودة في هذا المُدخل. فإذا وُجد توقيعٌ واحدٌ على الأقل له القيمة "ثقة متناهية"، فإن حقل شرعية المفتاح يأخذ القيمة "شرعية كاملة". إذا لم يكن الأمر كذلك، يقوم PGP بحساب مجموعٍ موزونٍ (weighted sum) من قيم الثقة المتاحة. يُعطى الوزن 1/X للتوقيعات التي هي محل ثقة "دائماً"،

بينما يُعطى الوزن 1/Y للتوقيعات التي هي محل ثقة "عادةً"؛ حيث X وY متغيران يحددهما المُستخدِم. وعندما يصل مجموع الأوزان لمقدمي أزواج المفاتيح وأسماء المستخدمين إلى 1، يُعدُّ مستوى الربط (binding) جديراً بالثقة ويأخذ حقل شرعية المفتاح القيمة "شرعية كاملة". وعليه، فإنه في حالة تعذر وجود "ثقة متناهية"، يحتاج الأمر إلى X من التوقيعات التي هي محل ثقة دائماً، أو Y من التوقيعات التي هي محل ثقة عادةً، أو أي تشكيلة منهما معاً.

الجدول 5-2: محتويات بايت عَلَم الثقة.

(a) قيمة الثقة الممنوحة لمالك مفتاح عام (تظهر بعد رزمة المفتاح؛ يحددها المُستخدِم)	(b) قيمة الثقة الممنوحة لزوج معرّف مفتاح عام/ اسم مستخدم (تظهر بعد رزمة اسم المُستخدِم؛ يحسبها PGP)	(c) قيمة الثقة الممنوحة لتوقيع (تظهر بعد رزمة التوقيع؛ نسخة مخبأة من حقل OWNERTRUST (الثقة في المالك) لصاحب التوقيع)
حقل OWNERTRUST (الثقة في المالك) - ثقة غير معرّفة - مُستخدِم غير معروف - لا يوثق به عادةً لتوقيع مفاتيح أخرى - يوثق به عادةً لتوقيع مفاتيح أخرى - يوثق به دائماً لتوقيع مفاتيح أخرى - هذا المفتاح موجود في حلقة مفاتيح سرية (ثقة متناهية) **بت BUCKSTOP (المسؤول)** - يأخذ القيمة 1 إذا كان هذا المفتاح موجوداً في مجموعة مفاتيح سرية	**حقل KEYLEGIT (شرعية المفتاح)** - ثقة غير معروفة أو غير معرّفة - ملكية المفتاح غير موثوقة - ثقة جزئية في ملكية المفتاح - ثقة كاملة في ملكية المفتاح **بت WARNONLY (تحذير فقط)** - يأخذ القيمة 1 اذا كان المُستخدِم يريد أن يتم إنذاره فقط إذا استُخدم في عملية التشفير مفتاح لم يتم التحقق منه تماماً	**حقل SIGTRUST (الثقة في التوقيع)** - ثقة غير معرّفة - مُستخدِم غير معروف - لا يوثق به عادةً لتوقيع مفاتيح أخرى - يوثق به عادةً لتوقيع مفاتيح أخرى - يوثق به دائماً لتوقيع مفاتيح أخرى - هذا المفتاح موجود في حلقة مفاتيح سرية (ثقة متناهية) **بت CONTIG (مسار ثقة متصل)** - يأخذ القيمة 1 إذا كان التوقيع يقود إلى مسار تصديق موثوق متصل يعود إلى صاحب حلقة المفاتيح الذي يتمتع بثقة متناهية

يعالج PGP بشكلٍ دوري حلقة المفاتيح العامة لتحقيق الاتساق (consistency). تتم هذه العملية أساساً من أعلى إلى أسفل؛ حيث يقوم PGP لكل حقل OWNERTRUST بمسح حلقة المفاتيح بحثاً عن جميع التوقيعات من تأليف ذلك المالك لتحديث قيم حقول "الثقة في التوقيع" بها لتساوي قيمة حقل "الثقة في المالك". تبدأ العملية بالمفاتيح التي تتمتع بثقة متناهية، ثم تُحسب قيمة حقل "شرعية المفتاح" لكل مفتاحٍ حسب التوقيعات الملحقة به.

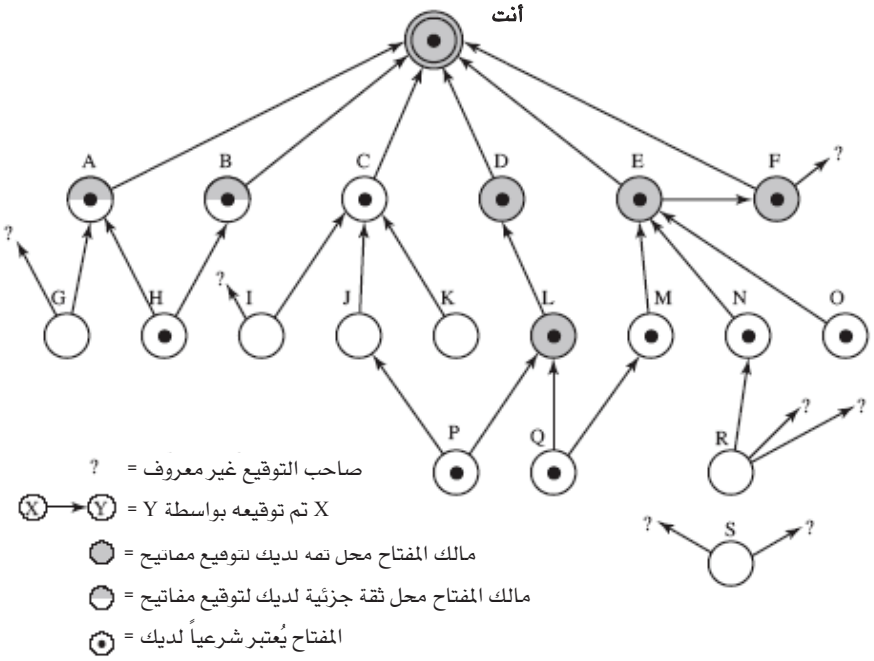
يوضّح الشكل 5-7 مثالاً لطريقة الربط ما بين الثقة في التوقيع وشرعية المفتاح.[2] ويبيّن الشكل هيكل حلقة مفاتيح عامة تُمكّن مستخدمها من تحصيل عددٍ من المفاتيح العامة، بعضها مباشرةً من مالكيها وبعضها الآخر من طرف ثالث كخادم مفاتيح (key server) مثلاً.

ترمز العقدة "أنت" للمُدخل المناظر لهذا المُستخدِم في حلقة المفاتيح. هذا المفتاح شرعي، وقيمة حقل "الثقة في المالك" له هي "ثقة متناهية". تكون قيمة "الثقة في المالك" لكل العقد الأخرى في حلقة المفاتيح غير معرَّفة (undefined) إلا إذا كان المُستخدِم قد أعطاها قيمة أخرى. في هذا المثال، حدّد المُستخدِم أنه يثق دائماً في المستخدمين D وE وF وL لتوقيع مفاتيح أخرى. ويبيّن الشكل أن هذا المُستخدِم يثق جزئياً في المستخدمين A وB لتوقيع مفاتيح أخرى.

وهكذا تدل نسبة التظليل للعقد في الشكل 5-7 على مستوى الثقة التي يوليها ذلك المستخدم للمستخدمين الآخرين. ويبيّن الهيكل الشجري المفاتيح التي تم توقيعها من قِبَل كل مستخدمٍ آخر. إذا تم توقيع مفتاح بواسطة مستخدمٍ يوجد مفتاحه أيضاً في نفس حلقة المفاتيح تلك، يُوَصِّل السهم المفتاح الموقَّع إلى الجهة الموقِّعة. إذا ما تم توقيع مفتاح بواسطة مستخدمٍ لا يوجد مفتاحه في نفس حلقة المفاتيح، يُوَصِّل السهم المفتاح الموقَّع إلى علامة استفهام للإشارة إلى أن الموقِّع لهذا المستخدم غير معروف.

---

[2] قام فيل زيمرمان مشكوراً بتوفير هذا الشكل للمؤلف.



الشكل 5-7: مثال على نموذج الثقة في PGP.

يوضّح الشكل 5-7 النقاط الآتية:

1. لاحظ أن كل المفاتيح التي يتمتع أصحابها كلياً أو جزئياً بثقة هذا المُستخدِم تم توقيعها من قِبَل هذا المُستخدِم نفسه، باستثناء العقدة L. إن وجود توقيع المُستخدِم ليس ضرورياً دائماً، كما يدل على ذلك وجود العقدة L، غير أن الممارسة العملية تبين أن معظم المستخدمين يقومون في الغالب بتوقيع المفاتيح لمعظم المستخدمين الذين يثقون بهم. فعلى سبيل المثال، رغم أن مفتاح E قد وقّعه الموثّق F، فقد آثر مُستخدِم حلقة المفاتيح في حالتنا هذه توقيع مفتاح E مباشرة.

2. نفترض هنا أن توقيعين بثقة جزئية كافيان لتوثيق مفتاح. وعليه، فإن PGP يُعدُّ مفتاح المستخدِم H شرعياً لأنه تم توقيعه من قِبَل كلٍّ من A وB، وكلاهما موثوق جزئياً.

3. يمكن اعتبار المفتاح شرعياً إذا وقَّعه مُستخدِم واحد موثوق تماماً أو مستخدمين اثنين موثوقين جزئياً، لكن مستخدِم ذلك المفتاح لن يكون موثوقاً لتوقيع مفاتيح أخرى. فعلى سبيل المثال، يُعدُّ مفتاح N شرعياً لأنه موقَّعٌ من قِبَل E، والذي يثق فيه هذا المستخدم، ولكن N لا يتمتع بالثقة لتوقيع مفاتيح أخرى، حيث إن هذا المستخدم لم يعط N قيمة الثقة اللازمة لذلك. وعليه، فرغم أن مفتاح R موقَّع من قِبَل N، فإن PGP لا يُعدُّ مفتاح R شرعياً. إن هذا الوضع منطقيٌ للغاية. فإذا كنت ترغب في إرسال رسالة خاصة إلى شخصٍ ما، فليس من الضروري أن تثق به بأي شكلٍ من الأشكال. من الضروري فقط أن تكون على يقين من أنك لديك المفتاح العام الصحيح لذلك الشخص.

4. يبيّن الشكل 5-7 أيضاً مثالاً لعقدة منفصلة "يتيمة" هي العقدة S التي تحمل توقيعين من مجهولين. قد يكون هذا المفتاح قد تم الحصول عليه من خادم مفاتيح. لا يفترض PGP أن هذا المفتاح شرعيٌ لمجرد أنه جاء من خادم مفاتيح حسن السمعة. يتعين على مستخدم حلقة المفاتيح إعلام PGP بأن هذا المفتاح شرعيٌ بتوقيعه أو بإخطار PGP بأنه على استعداد للثقة تماماً في أحد المستخدمين الموقعين على ذلك المفتاح.

نقطة أخيرة: ذكرنا آنفاً أنه يمكن أن ترتبط عدة أسماء مستخدمين بنفس المفتاح العام في حلقة المفاتيح العامة. ويمكن أن يحدث ذلك بسبب قيام شخص بتغيير اسمه أو بسبب أنه تم تقديمه من خلال توقيع تحت عدة أسماء مختلفة تشير مثلاً إلى عدة عناوين بريد إلكتروني لنفس الشخص. ومن ثَمَّ، يمكننا اعتبار المفتاح العام جذر شجرة. للمفتاح العام عددٌ من أسماء المستخدمين مرتبطة به، وكل اسم مُستخدِم تحته عدد من التوقيعات. يعتمد الارتباط بين اسم مستخدم

بعينه ومفتاحٍ ما على التوقيعات المرتبطة بذلك الاسم للمستخدم وذلك المفتاح، بينما يُعدُّ مستوى الثقة في ذلك المفتاح (للاستخدام في توقيع مفاتيح أخرى) دالة في كل التوقيعات المعتمِدة عليه.

❖ إلغاء المفاتيح العامة:

قد يرغب مستخدمٌ في إلغاء مفتاحه العام الحالي إما لاشتباهه في تمكن أحد من الاستحواذ على أحد مفاتيحه العامة، أو لمجرد رغبته في تجنب استخدام نفس المفتاح لفترة طويلة. لاحظ أن الاستحواذ على المفتاح يقتضي تمكّن الخَصْم بشكلٍ أو بآخر من الحصول على نسخة غير مُشفَّرة من مفتاحك الخاص، أو تمكنه من الحصول على كلٍّ من المفتاح الخاص من حلقة المفتاح الخاصة بك وكلمة المرور التي تستخدمها.

يعني اصطلاح "إلغاء المفتاح العام" قيام المالك بإصدار شهادة إلغاء للمفتاح والتوقيع عليها. تأخذ تلك الشهادة نفس شكل شهادة التوقيع المعتادة ولكنها تتضمن ما يدل على أن الغرض منها هو إلغاء استخدام ذلك المفتاح العام. لاحظ أنه ينبغي استخدام المفتاح الخاص المناظر لتوقيع الشهادة التي تلغي مفتاحاً عاماً. وينبغي على مالك المفتاح الملغي بعد ذلك العمل على نشر تلك الشهادة على أوسع نطاق وبأسرع وقت ممكن لتمكين المراسلين المحتملين من تحديث حلقات المفاتيح العامة الخاصة بهم.

لاحظ أيضاً أنه بوسع الخَصْم الذي استحوذ على المفتاح الخاص إصدار مثل تلك الشهادة. ومع ذلك، فإن هذا سوف يحرم الخَصْم، وكذلك المالك الشرعي، من استخدام المفتاح العام، ومن ثَمَّ فإن هذا السيناريو من التهديد يُعدُّ أقل احتمالاً بكثير من الاستخدام الخبيث لمفتاح خاص مسروق.

2-5 امتدادات بريد الإنترنت متعددة الأغراض الآمنة (S/MIME)

امتدادات بريد الإنترنت متعددة الأغراض الآمنة (S/MIME) هي تعزيزات أمنية لمعيار امتدادات بريد الإنترنت متعددة الأغراض (MIME) تعتمد على تقنية من شركة RSA لأمن المعلومات. رغم أن كلاً من PGP وS/MIME يسيران قُدُماً على مسار معايير فريق عمل هندسة الإنترنت (Internet Engineering Task Force (IETF))، فإنه يبدو من المرجّح أن S/MIME سيصبح معيار الصناعة لاستخدام الشركات والمنظمات، بينما سيبقى PGP اختياراً للكثير من المستخدمين لتأمين البريد الإلكتروني الشخصي. تم تعريف S/MIME في عدد من الوثائق، من أهمها طلبات التعليقات (RFCs) أرقام 3369 و3370 و3850 و3851.

لفهم S/MIME، نحتاج أولاً إلى أن نستوعب بشكلٍ عام صيغة البريد الإلكتروني التي يستخدمها (أي صيغة معيار MIME). ولكن لإدراك أهمية MIME علينا العودة إلى معيار البريد الإلكتروني التقليدي RFC 822 الذي لا يزال يُستخدم بشكلٍ عام. ومن ثَمَّ، يبدأ هذا الجزء بمقدمةٍ عن هذين المعيارين السابقين ثم ينتقل بعد ذلك لمناقشة S/MIME.

1-2-5 معيار RFC 822

يُعرِّف معيار RFC 822 صيغةً للرسائل النصّية التي تُرسَل باستخدام البريد الإلكتروني. استُخدم RFC 822 معياراً لرسائل البريد النصّية على الإنترنت ولا يزال شائع الاستخدام حتى الآن.

في سياق RFC 822، يُنظر إلى الرسالة على أنها ظرف ومحتويات. يتضمن الظرف كل المعلومات اللازمة لنقل الرسالة وتسليمها، بينما تضم المحتويات موضوع الرسالة المطلوب تسليمه إلى المُستلِم. يتناول معيار RFC 822 محتويات الرسالة فقط، غير أنه يتضمن مجموعة من الترويسات التي يمكن أن يستخدمها نظام البريد لإعداد الظرف، والمعيار مصممٌ بحيث يُسهّل على البرامج الحصول على تلك المعلومات.

يتسم الهيكل العام للرسالة وفقاً لمعيار RFC 822 بأنه في غاية البساطة. تتكون الرسالة من عددٍ من سطور المقدمة (ترويسة)، يليها نصٌ بدون قيود (متن الرسالة). يفصل بين الترويسة والمتن سطرٌ فارغ. وبعبارة أخرى، الرسالة نص بصيغة آسكي، وجميع السطور من البداية حتى أول سطر فارغ يُفترض أنها تُشكِّل الترويسة التي يستخدمها وكيل المستخدم للبريد الإلكتروني.

يتألف سطر الترويسة عادةً من كلمة مفتاحية يليها العلامة ":" وتتبعها قيمة حقل الكلمة المفتاحية. تسمح الصيغة المعيارية بتقسيم سطر طويل إلى عدة أسطر. أكثر الكلمات المفتاحية استخداماً هي: من "From"، وإلى "To"، والموضوع "Subject"، والتاريخ "Date". فيما يلي مثال لرسالة:

Date: Tue, 16 Jan 1998 10:37:17 (EST)
From: “William Stallings” <ws@shore.net>
Subject: The Syntax in RFC 822
To: Smith@Other-host.com
Cc: Jones@Yet-Another-Host.com

Hello. This section begins the actual message body, which is delimited from the message heading by a blank line.

من الحقول الأخرى ضمن ترويسة الرسالة، حسب معيار RFC 822، حقل معرِّف الرسالة (*Message-ID*). ويتضمن هذا الحقل معرِّفاً فريداً مرتبطاً بالرسالة.

5-2-2 امتدادات بريد الإنترنت متعددة الأغراض (MIME)

يعدُّ MIME امتداداً لإطار RFC 822 يهدف لمعالجة بعض المشاكل والقيود التي تكتنف استخدام البروتوكول البسيط لنقل البريد (Simple Mail Transfer Protocol (SMTP)) أو غيره من بروتوكولات نقل البريد ومعيار RFC 822 في نقل البريد الإلكتروني. يسرد [RODR02] جوانب القصور الآتية التي يعاني منها نظام SMTP/822:

1. لا يستطيع بروتوكول SMTP إرسال ملفات تنفيذية أو غير ذلك من الكائنات الثنائية (binary objects). تُستخدم حالياً عدة أساليب لتحويل الملفات الثنائية إلى صيغة نص بحيث يمكن إرسالها عن طريق بروتوكول SMTP ونظم البريد الإلكتروني، بما في ذلك أسلوب UUencode/UUdecode الشهير على نظام التشغيل يونيكس. غير أن أيّاً من تلك الأساليب لا يرتقي إلى مستوى معيار حقيقي أو حتى معيار بحكم الواقع (de facto standard).
2. لا يُمكن لبروتوكول SMTP نقل بيانات النصوص التي تتضمن حروف اللغات القومية؛ لأن تلك الحروف تُمثّل برموز طول كلٍّ منها 8 بتات وقيم عشرية تعادل 128 أو أكبر، في حين أن بروتوكول SMTP يقتصر على صيغة آسكي بطول 7 بتات.
3. يمكن أن يرفض بروتوكول SMTP نقل رسائل البريد الإلكتروني التي يزيد حجمها عن حد معين.
4. لا تستخدم بوابات بروتوكول SMTP التي تترجم ما بين صيغة آسكي للحروف وتكويد EBCDIC مجموعة متسقة من الترحيلات (mappings)، مما يؤدي إلى مشاكل في الترجمة.
5. لا تستطيع بوابات بروتوكول SMTP لشبكات X.400 للبريد الإلكتروني التعامل مع البيانات غير النصّية الواردة في رسائل X.400.
6. بعض أنظمة SMTP المستخدمة حالياً لا تلتزم تماماً بمعيار SMTP كما ورد في RFC 821. من المشاكل الشائعة ما يأتي:
   - حذف، أو إضافة، أو إعادة ترتيب علامتَي بداية السطر (أي حرف عودة العربة (carriage return) وحرف فتح سطر جديد (linefeed)).
   - بتر أو تدوير السطور الأطول من 76 حرفاً.

- إزالة الحيز الأبيض (white space) في نهاية السطر (أي حروف الجدولة (tab) والمسافة (space)).
- حشو سطورٍ في الرسائل لتصبح بنفس الطول.
- تحويل حروف الجدولة (tab) إلى عدد مكافئ من حروف المسافة (space).

كان المقصود من MIME حل هذه المشاكل على نحو يتوافق مع الأنظمة الموجودة حالياً لتنفيذ RFC 822. توجد المواصفات في وثائق RFC بالأرقام من 2045 وحتى 2049.

❖ نظرة عامة:

تتضمن مواصفات MIME العناصر الآتية:

1. تعريف خمسة حقولٍ جديدةٍ لترويسة الرسالة يمكن إدراجها في ترويسة RFC 822. توفر الحقول الجديدة معلومات عن متن الرسالة.
2. تعريف عددٍ من الصيغ لمحتوى الرسالة، ومن ثَمَّ وضع معايير لطرق التمثيل اللازمة لدعم البريد الإلكتروني للوسائط المتعددة.
3. تعريف تكويدات النقل (transfer encodings) للتحويل من أي صيغةٍ للمحتوى إلى شكل للبيانات محميٍ من التغيير أثناء الانتقال عبر نظام البريد.

في هذا الجزء الفرعي من الكتاب نقدم الحقول الخمسة الجديدة لترويسة الرسالة. في الجزأين الفرعيين التاليين سنتناول صيغ المحتوى وتكويدات التحويل.

فيما يلي ملخص بالحقول الخمسة الجديدة لترويسة الرسالة التي تم استحداثها في MIME:

- رقم الإصدار لـ MIME (MIME-Version): يجب أن تكون له القيمة 1.0. يشير هذا الحقل إلى أن الرسالة متوافقة مع وثائق RFC رقم 2045 ورقم 2046.
- نوع المحتوى (Content-Type): يصف البيانات المتضمنة في متن الرسالة بالتفصيل الكافي لتمكين وكيل المُستلِم من اختيار وكيل أو آلية مناسبة لتمثيل البيانات للمستخدِم أو التعامل مع البيانات على نحوٍ مناسب.
- تكويدات نقل المحتوى (Content-Transfer-Encodings): تبيِّن نوع التحويلات التي استُخدمت لتمثيل متن الرسالة بشكلٍ مناسب لنقل الرسالة بالبريد الإلكتروني.
- معرِّف المحتوى (Content-ID): يُستخدَم لتعريف كيانات MIME بطريقةٍ لا تحتمل الشك في سياقات متعددة.
- وصف المحتوى (Content-Description): نصٌّ يصف كائن متن الرسالة، ويفيد ذلك بشكلٍ خاص عندما يكون الكائن من النوع غير المقروء (كالبيانات الصوتية مثلاً).

يمكن أن يظهر أيٌّ من هذه الحقول أو كلها في ترويسة RFC 822 المعتادة. يجب دعم الحقول: MIME-Version وContent-Type وContent-Transfer-Encodings كحدٍّ أدنى لتحقيق التوافق. أما الحقلان الأخيران فاختياريان ويمكن تجاهلهما على نظام المُستلِم.

❖ أنواع المحتوى في MIME :

ينصبُّ الجزء الأكبر من مواصفات MIME على تعريف تشكيلة من أنواع المحتوى. وهذا يعكس مدى الحاجة إلى توفير أساليبَ معياريةٍ للتعامل مع التشكيلة الواسعة المتوفرة حالياً من طرق تمثيل المعلومات في بيئة وسائط متعددة.

الجدول 5-3: أنواع المحتوى في معيار MIME.

النوع	النوع الفرعي	الوصف
نص (Text)	عادي (Plain)	نص غير منسّق، قد يكون آسكي أو آيزو 8859.
	مزخرف (Enriched)	يوفر مرونة أكبر في تنسيق الرسالة.
متعدد الأجزاء (Multipart)	مختلط (Mixed)	الأجزاء المختلفة مستقلة ولكن يتم إرسالها مع بعضها. ينبغي عرض تلك الأجزاء على المُستلِم بنفس ترتيب ظهورها في رسالة البريد الإلكتروني.
	متوازي (Parallel)	يختلف عن "مختلط" فقط في أنه لا يوجد ترتيب محدد لعرض الأجزاء على المُستلِم.
	بدائل (Alternative)	توفر الأجزاء المختلفة صيغاً بديلةً لنفس المعلومات، وتوضع في الرسالة بترتيب جودتها في تمثيل المعلومات الأصلية. ينبغي على نظام البريد عند المُستلِم عرض "أفضل" صيغة للمستخدم.
	خلاصة (Digest)	يشبه "مختلط"، ولكن النوع/"النوع الفرعي" المعتاد لكل جزء هو message/rfc822.
رسالة (Message)	Rfc882	المتن في حد ذاته رسالة مغلّفة متوافقة مع معيار RFC822.
	جزئي (Partial)	تُستخدم للسماح بتقطيع رسالة بريدٍ كبيرةٍ على نحو يتسم بالشفافية فيما يتعلق بالمُستلِم.
	متن خارجي (External-body)	يتضمن مؤشراً إلى كائن موجود في مكان آخر.
صورة (Image)	JPEG	الصورة بصيغة JPEG، تكويد JFIF.
صورة (Image)	GIF	الصورة بصيغة GIF.
فيديو (Video)	MPEG	بصيغة MPEG.
صوت (Audio)	أساسي (Basic)	قناة أحادية بتكويد ISDN بـ 8 بتات وخوارزمية μ-law وتؤخذ العينات بمعدل 8 كيلوهرتز.
تطبيق (Application)	PostScript	صيغة Adobe Postscript.
	سلسلة بايتات	صيغة عامة لبياناتٍ ثنائيةٍ تتألف من بايتات (طول كلٍّ منها 8 بتات).

يبيّن الجدول 5-3 أنواع المحتوى المحددة في الوثيقة RFC 2046. فهناك سبعة أنواعٍ رئيسةٍ مختلفةٍ من المحتوى، بالإضافة إلى 15 نوعاً فرعياً. ويحدد "نوع المحتوى" نوع البيانات بشكلٍ عام، بينما يحدد "النوع الفرعي" صيغة بعينها لذلك النوع من البيانات.

لا يتطلب متن الرسالة من النوع النصي (text type) استخدام أي برمجيات خاصة للحصول على معنى النص بالكامل، باستثناء دعم طاقم الحروف المحدّد. النوع الفرعي الأساسي هو "plain text" (نص عادي)، وهو مجرد سلسلة من حروف آسكي أو حروف آيزو 8859. يوفر النوع الفرعي "enriched" (مزخرف) مرونةً إضافيةً في تنسيق متن الرسالة.

يدل نوع المحتوى multipart type (متعدد الأجزاء) على أن متن الرسالة يضم عدة أجزاء مستقلة. يتضمن حقل الترويسة Content-Type متغيراً يسمى boundary (الفاصل) يُستخدم للفصل بين الأجزاء المختلفة في متن الرسالة. لا ينبغي أن تظهر تلك الفواصل في أيٍّ من أجزاء الرسالة. يبدأ كل فاصل على سطر جديد، ويتألف من شرطتين تليهما قيمة الفاصل. يبدأ الفاصل النهائي الذي يبيّن نهاية الجزء السابق من متن الرسالة، بشرطتين كذلك. قد توجد ترويسة MIME عادية بشكلٍ اختياري ضمن كل جزء من متن الرسالة.

فيما يلي مثال بسيط لرسالة متعددة الأجزاء، تتضمن جزأين يتألَّف كلٌّ منهما من نصٍّ بسيط (المثال مأخوذ من RFC 2046):

```
From: Nathaniel Borenstein <nsb@bellcore.com>
To: Ned Freed <ned@innosoft.com>
Subject: Sample message
MIME-Version: 1.0
Content-type: multipart/mixed; boundary="simple boundary"

This is the preamble. It is to be ignored, though it is a handy place for mail
composers to include an
explanatory note to non-MIME conformant readers.
--simple boundary
```

```
This is implicitly typed plain ASCII text. It does NOT
end with a linebreak.
--simple boundary
Content-type: text/plain; charset=us-ascii
This is explicitly typed plain ASCII text. It DOES end
with a linebreak.
--simple boundary--
This is the epilogue. It is also to be ignored.
```

هناك أربعة أنواع فرعية للنوع الرئيس multipart (متعدد الأجزاء)، وتشترك كلها في نفس الصيغة. ويُستخدم النوع الفرعي multipart/mixed (متعدد الأجزاء/ مختلط) عند وجود عدة أجزاءٍ مستقلة في متن الرسالة تحتاج إلى التعامل معها بترتيب معيّن. أما في النوع الفرعي multipart/parallel (متعدد الأجزاء/ متوازي) فلا يُعدُّ ترتيب الأجزاء مهماً، وإذا كان نظام المُستلِم مجهزاً بشكلٍ مناسب، فسيمكنه عرض الأجزاء المتعددة على التوازي (أي في نفس الوقت). وعلى سبيل المثال، يمكن أن يصحب عرض نصٍّ أو صورةٍ تشغيل تعليقٍ صوتي.

في النوع الفرعي multipart/alternative (متعدد الأجزاء/ بدائل) تمثل الأجزاء نفس المعلومات بطرقٍ بديلةٍ مختلفة، وفيما يلي مثال على ذلك:

```
From: Nathaniel Borenstein <nsb@bellcore.com>
To: Ned Freed <ned@innosoft.com>
Subject: Formatted text mail
MIME-Version: 1.0
Content-Type: multipart/alternative;
boundary=boundary42
--boundary42
Content-Type: text/plain; charset=us-ascii
... plain text version of message goes here....
--boundary42
Content-Type: text/enriched
.... RFC 1896 text/enriched version of same message
goes here ...
--boundary42--
```

في هذا النوع الفرعي، تُرتب أجزاء متن الرسالة بترتيب أفضليتها (الأفضل في النهاية). في هذا المثال، إذا كان نظام المُستلِم قادراً على عرض الرسالة بالنّص المزخرف فبها ونعمت، وإلا فسيعرض الرسالة بصيغة النص العادي.

يُستخدم النوع الفرعي multipart/digest (متعدد الأجزاء/خلاصة) عندما يُعامَل كل جزء من أجزاء متن الرسالة بوصفه رسالة RFC 822 لها ترويسة. يتيح هذا النوع الفرعي تكوين رسائل تمثل أجزاؤها رسائل مستقلة. فعلى سبيل المثال، قد يجمع منسّق مجموعةٍ من العاملين رسائل البريد الإلكتروني من أعضاء المجموعة، ويضمها في حزمة ثم يُرسلها جميعاً في رسالة MIME واحدة.

يوفر نوع المحتوى message (رسالة) عدداً من الإمكانات المهمة لمعيار MIME. فالنوع الفرعي message/rfc822 (رسالة/rfc822) يشير إلى أن متن الرسالة هو رسالة كاملة، تضم ترويسةً ومتناً. على غير ما يوحي به اسم هذا النوع الفرعي فإن الرسالة المغلَّفة تلك لا يتعين أن تكون رسالة RFC 822، ولكن يمكن أن تكون أي رسالة MIME.

يسمح النوع الفرعي message/partial (رسالة/جزئي) بتقطيع رسالةٍ طويلةٍ إلى عدة أجزاء يتعين تجميعها من جديد في رسالةٍ واحدةٍ عند الوجهة النهائية. ولهذا النوع الفرعي ينبغي تعريف ثلاثة متغيرات في حقل Content-Type: Message/Partial؛ وتلك المتغيرات هي: *id* (معرِّف) مشترك لكل أجزاء الرسالة الطويلة، *sequence number* (رقم تسلسلي) وهو رقم فريد يميّز كل قطعة، و*total* (المجموع) ويمثل العدد الكلي لأجزاء الرسالة.

يشير النوع الفرعي message/external-body (رسالة/متن خارجي) إلى أن البيانات الفعلية التي تحملها هذه الرسالة ليست موجودة في المتن، ولكن المتن يحتوي على المعلومات اللازمة للوصول إلى تلك البيانات. وكما هو الحال مع غيره من أنواع الرسالة، يوجد للنوع الفرعي "رسالة/متن خارجي" ترويسة خارجية ورسالة مغلَّفة يكون لها ترويستها الخاصة. والحقل الوحيد المطلوب في الترويسة الخارجية

هو حقل "نوع المحتوى" الذي يعرِّف ذلك بوصفه النوع الفرعي "رسالة/متن خارجي". والترويسة الداخلية هي ترويسة الرسالة الخاصة بالرسالة المغلَّفة. وينبغي أن يتضمن حقل "نوع المحتوى" في الترويسة الخارجية متغير "نوع الوصول" (access-type) الذي يحدد طريقة الوصول للبيانات، مثلاً عن طريق بروتوكول نقل الملفات (FTP).

يشير نوع المحتوى application (تطبيق) إلى الأنواع المختلفة الأخرى للبيانات؛ وعادةً ما تكون بيانات ثنائية لا يتم تفسيرها أو معلومات سيتم معالجتها بواسطة تطبيق يدعمه نظام البريد.

❖ تكويدات النقل في معيار MIME:

المكوِّن الأساسي الآخر في المعيار MIME، بالإضافة إلى تحديد نوع المحتوى، هو تعريف تكويدات لنقل متن الرسائل. والهدف من ذلك هو توفير خدمة موثوقة لتوصيل الرسائل عبر أكبر عدد من البيئات.

يحدِّد معيار MIME طريقتين لتكويد البيانات. في الواقع يمكن أن يأخذ حقل Content-Transfer-Encoding (تكويد نقل المحتوى) ست قيم، كما هو مبين في الجدول 5-4. غير أن ثلاثاً من تلك القيم (7 بتات، 8 بتات ، وثنائي) تدل على أنه لا يحدث تكويد معين ولكنها توفّر بعض المعلومات عن طبيعة البيانات. ولنقل البريد على بروتوكول SMTP يكون من المناسب استخدام القيمة 7 بتات. ويمكن استخدام القيمتين 8 بتات وثنائي في سياقات أخرى لنقل البريد. ومن القيم الأخرى لتكويد نقل المحتوى قيمة x-token (العلامة x) التي تدل على استخدام نظام تكويدٍ آخر سيتم تحديد اسمه. وقد يكون ذلك نظاماً خاصاً بأحد المنتجين أو بأحد التطبيقات. أما نظاما التكويد المحدّدان بالفعل في معيار MIME فهما: quoted-printable وbase64. وقد تم تعريف هذين النظامين لتوفير الخيار بين أسلوب لنقل البيانات بشكلٍ يمكن قراءته بواسطة البشر وآخر يُعدُّ آمناً بالنسبة لجميع أنواع البيانات بطريقة مختصرة بدرجة معقولة.

من المفيد استخدام تكويد النقل quoted-printable عندما تتكون البيانات في الغالب من بايتات تناظر حروف آسكي قابلة للطباعة. وفي جوهره يقوم النظام بتمثيل الحروف غير الآمنة بالمكافئ الست - عشري (hexadecimal) للكود الخاص بها كما يُدخِل علامات سطور خاصة (يمكن إزالتها) كيلا يتجاوز طول السطر في الرسالة 76 حرفاً.

الجدول 5-4 تكويدات النقل بمعيار MIME.

7bit	تُمثّل البيانات كلها على شكل سطور قصيرة من حروف آسكي.
8bit	السطور قصيرة، ولكن قد توجد حروف بغير صيغة آسكي (بايتات بالقيمة 1 للبت الأعلى وزناً).
Binary	بالإضافة إلى وجود حروف بغير صيغة آسكي قد لا تكون السطور قصيرة بما فيه الكفاية للنقل عبر بروتوكول SMTP.
Quoted-printable	تُكوّد البيانات بحيث إذا كانت في معظمها نص آسكي، فإنه سيمكن للبشر التعرف على شكل البيانات المكوّدة بشكلٍ عام.
base64	تكوّد البيانات بتمثيل الكتل من 6 بتات من البيانات الداخلة على شكل كتل من 8 بتات كلها حروف آسكي يمكن طباعتها.
x-token	نظام تكويد غير قياسي يُذكر اسمه.

يُستخدم نظام تكويد النقل base64 (الأساس 64) – ويُعرَف أيضاً بـ radix-64 – بكثرة لتكويد البيانات الثنائية بطريقة تحميها من المعالجة بواسطة برامج نقل البريد. ويُستخدم النظام أيضاً في PGP، كما هو مبيّن في الملحق B-5.

```
MIME-Version: 1.0
From: Nathaniel Borenstein <nsb@bellcore.com>
To: Ned Freed <ned@innosoft.com>
Subject: A multipart example
Content-Type: multipart/mixed;
boundary_unique-boundary-1

هذه مساحة الديباجة في رسالةٍ متعددة الأجزاء. ينبغي على نظام قراءة البريد الذي بوسعه فهم صيغة الرسائل متعددة الأجزاء
تجاهل هذه الديباجة. إذا كنت تقرأ هذا النص، فقد ترغب في تغيير قارئ البريد لديك بحيث يتمكّن من عرض الرسائل متعددة
الأجزاء على الوجه الصحيح.

--unique-boundary-1

... يظهر بعض النص هنا...

[لاحظ أن السطر الفارغ أعلاه يعني أنه لم يتم إدخال حقول ترويسة وأن هذا نصٌ يستخدم طاقم حروف آسكي الأمريكي. كان
من الممكن عمل ذلك بالكتابة صراحةً كما في الجزء الآتي.]

--unique-boundary-1
Content-type: text/plain; charset=US-ASCII

كان من الممكن أن يكون هذا جزءاً من الجزء السابق، ولكنه يوضّح الكتابة الصريحة في مقابل الكتابة الضمنية لأجزاء
المتن.

--unique-boundary-1
Content-Type: multipart/parallel; boundary_unique-boundary-2
--unique-boundary-2
Content-Type: audio/basic
Content-Transfer-Encoding: base64

.... تظهر هنا بيانات صوت مكوّدةٌ بنظام base64 على قناةٍ أحاديةٍ وخوارزمية بقانون μ ومعدل أخذ عينات قدره 8000 هرتز ...
Content-Type: image/jpeg
Content-Transfer-Encoding: base64

.... تظهر هنا بيانات صورةٍ مكوّدةٍ بنظام base64 ....

--unique-boundary-2--
--unique-boundary-1
Content-type: text/enriched

This is <bold><italic>richtext.</italic></bold> <smaller>as defined in RFC 1896</smaller>
Isn't it <bigger><bigger>cool?</bigger></bigger>

--unique-boundary-1
Content-Type: message/rfc822

From: (mailbox in US-ASCII)
To: (address in US-ASCII)
Subject: (subject in US-ASCII)
Content-Type: Text/plain; charset_ISO-8859-1
Content-Transfer-Encoding: Quoted-printable

... يظهر هنا نصٌّ إضافيٌّ بصيغة ISO-8859-1 ...

--unique-boundary-1--
```

الشكل 5-8: مثال لهيكل رسالة MIME.

❖ **مثال لرسالة بريد متعددة الأجزاء:**

يبيّن الشكل 5-8 – والمُقتبس من RFC 2045 – مخططاً لرسالةٍ معقدةٍ متعددة الأجزاء. تضم الرسالة خمسة أجزاء مطلوب عرضها على التوالي: اثنين منها يمثلان مقدمةً بنصٍّ عادي، ورسالةً مدمجةً مكوّنةً من عدة أجزاء، وجزءاً يتكون من نصٍّ مزخرف، وأخيراً رسالةً نصّيةً مغلَّفةً تتألف من طاقم حروف غير الآسكي. وتتألف الرسالة المدمجة متعددة الأجزاء من جزأين هما صورة ومقطع صوتي، مطلوب عرضهما معاً على التوازي.

❖ **الصيغة القانونية (Canonical Form):**

تُعتبر الصيغة القانونية أحد المفاهيم المهمة في المعيارين MIME وS/MIME. الصيغة القانونية هي صيغة مناسبة لنوع المحتوى، وهي مُعرَّفة بشكلٍ قياسي للاستخدام بين أنظمة مختلفة. هذا على النقيض من الصيغة المحلية (native form) التي تخص نظاماً بعينه. سيساعد الجدول 5-5 والمقتبس من RFC 2049 في توضيح هذه المسألة.

5-2-3 وظائف معيار S/MIME

يشبه معيار S/MIME حزمة PGP إلى حدٍّ كبير من حيث الوظائف العامة التي يؤديها. فكلاهما يوفر إمكانية لتوقيع الرسائل أو تشفيرها أو كلاهما معاً. في هذا الجزء سنستعرض بإيجاز إمكانات S/MIME. بعد ذلك سنلقي نظرةً أكثر تفصيلاً على تلك الإمكانات وذلك بتناول صيغ الرسائل وطرق تحضيرها.

الجدول 5-5: الصيغة المحلية والقانونية.

الصيغة المحلية (Native)	يتم إنشاء متن الرسالة الذي سيُرسَل في الصيغة المحلية للنظام. يُستخدم طاقم الحروف المحلي، وعند اللزوم، تُستخدم اصطلاحات نهاية السطر كذلك. يمكن أن يكون المتن ملفاً نصّياً من طراز Unix، أو صورة بصيغة Sun Raster، أو ملفاً مفهرساً على نظام الشغيل VMS، أو بياناتٍ سمعيةٍ بصيغة تعتمد على النظام ومخزّنةٍ فقط في الذاكرة، أو أي شيءٍ آخر يناظر النموذج المحلي لشكلٍ من أشكال المعلومات. وجوهرياً، يتم إنشاء البيانات في الصيغة "المحلية" المناظرة للنوع المحدد من قِبَل الوَسَط المُستخدَم.
الصيغة القانونية (Canonical)	يتم تحويل متن الرسالة كله، بما في ذلك المعلومات "خارج النطاق" (out-of-band)، كأطوال السجلات وربما المعلومات عن خصائص الملف، إلى صيغةٍ قانونية. يُحدد نوع الوَسَط للمتن وكذلك للخصائص المرتبطة به طبيعة الصيغة القانونية المستخدمة. وقد يتضمن التحويل إلى الصيغة القانونية المناسبة تحويل طاقم الحروف المُستخدَم، وتحويل بيانات الصوت، وضغط البيانات، أو غير ذلك من العمليات المختلفة الخاصة بأنواع الوسائط المختلفة. ولكن إذا تطلب الأمر تحويل طاقم الحروف فينبغي فهم معاني نوع الوسط بعناية، حيث يمكن أن تكون لتلك المعاني تداعيات قوية على أي تحويل لأطقم الحروف (فيما يتعلق مثلاً بالتراكيب المفهومة نحوياً في أنواع النص الفرعية غير النوع "عادي").

❖ الوظائف:

يوفر معيار S/MIME الوظائف الآتية:

- بيانات مغلَّفة (Enveloped data): ويشمل ذلك تشفير المحتوى من أي نوع وتشفير مفاتيح تشفير المحتوى لواحد أو أكثر من المُستلِمين.
- بيانات موقَّعة (Signed data): يتم إنشاء التوقيع الرقمي بأخذ خلاصة محتوى الرسالة المطلوب توقيعها وتشفير ذلك المحتوى بمفتاح التشفير الخاص بالموقِّع. ويتم بعد ذلك تكويد المحتوى مضافاً إليه التوقيع باستخدام نظام base64. ولا يمكن الاطلاع على رسالة البيانات الموقَّعة إلا من قِبَل مُستلِم تتوافر لديه إمكانات S/MIME.

- بيانات موقَّعة بدون تكويد (Clear-signed data): كما في حالة البيانات الموقَّعة، يتم إنشاء التوقيع الرقمي لمحتوى الرسالة. ولكن في هذه الحالة يتم تكويد التوقيع الرقمي فقط باستخدام نظام base64. ونتيجة لذلك، يمكن لمستلمي الرسالة ممن لا تتوافر لديهم إمكانات S/MIME عرض محتوى الرسالة، رغم عدم قدرتهم على التحقق من التوقيع.
- بيانات موقَّعة ومغلَّفة (Signed and enveloped data): يمكن أن تكون الكيانات الموقَّعة فقط وكذا المُشفَّرة فقط متراكبة (nested)، بحيث يمكن توقيع البيانات المُشفَّرة أو تشفير البيانات الموقَّعة المكوَّدة أو الموقَّعة بدون تكويد.

الجدول 5-6: الخوارزميات المستخدمة في معيار S/MIME.

الوظيفة	المتطلبات
كوّن خلاصة الرسالة لاستخدامها في توليد التوقيع الرقمي	يتعين (must) دعم SHA-1. ينبغي (should) على المُستلِم دعم MD5 للتوافق مع الإصدارات السابقة.
قم بتشفير خلاصة الرسالة للحصول على التوقيع الرقمي	يتعين على الوكلاء المُرسلِين والمستلِمين دعم DSS. ينبغي على الوكلاء المُرسلِين دعم التشفير بواسطة RSA. ينبغي على الوكلاء المستلِمين دعم التحقق من توقيعات RSA بمفاتيح بطول من 512 إلى 1024 بتاً.
قم بتشفير مفتاح الجلسة لإرساله مع الرسالة	ينبغي على الوكلاء المُرسلِين والمستلِمين دعم ديفي - هيلمان. يتعين على الوكلاء المُرسلِين والمستلِمين دعم التشفير بواسطة RSA بمفاتيح بطول من 512 إلى 1024 بتاً.
قم بتشفير الرسالة لإرسالها مع مفتاح جلسة يُستخدم مرة واحدة	يتعين على الوكلاء المُرسلِين والمستلِمين دعم التشفير بواسطة DES الثلاثية. ينبغي على الوكلاء المُرسلِين دعم التشفير بواسطة AES الثلاثية. ينبغي على الوكلاء المُرسلِين دعم التشفير بواسطة RC2/40.
قم بإنشاء كود توثيق الرسالة	يتعين على الوكلاء المستلِمين دعم HMAC مع SHA-1. ينبغي على الوكلاء المستلِمين دعم HMAC مع SHA-1.

❖ خوارزميات التشفير:

يلخص الجدول 5-6 خوارزميات التشفير المستخدمة في معيار S/MIME. يستخدم S/MIME المصطلحات الآتية لتحديد مستويات المتطلبات في المعيار (مأخوذة من وثيقة RFC 2119):

- **يتعيّن** (Must): أي أن المتطلب شرطٌ ضروريٌ وأساسيٌ للمواصفات. ويجب أن يتضمن النظام تلك السمة أو الوظيفة ليكون مطابقاً للمواصفات.
- **ينبغي** (Should): قد توجد أسباب وجيهة في ظروف معينة لتجاهل تلك السمة أو الوظيفة، ولكن يوصَى بأن يتضمن النظام تلك السمة أو الوظيفة.

يتضمن معيار S/MIME ثلاث خوارزميات للتشفير بالمفاتيح العامة. تُعدُّ خوارزمية معيار التوقيع الرقمي (Digital Signature Standard (DSS)) التي حددها معهد NIST خوارزمية التوقيع الرقمي المفضلة. ويعتبر معيار S/MIME خوارزمية ديفي - هيلمان الخوارزمية المفضلة لتشفير مفاتيح الجلسات. في الواقع، يستخدم S/MIME نوعية من خوارزمية ديفي - هيلمان توفر كلاً من التشفير وإزالة التشفير وتُعرَف باسم الجَمَل (ElGamal). ويمكن أيضاً استخدام خوارزمية RSA – التي تناولناها في الفصل الثالث – بديلاً لتشفير كلٍّ من التوقيعات ومفاتيح الجلسات. تلك هي نفس الخوارزميات المستخدمة في PGP، وهي توفر مستوى عالياً من الأمن. وفيما يتعلق بدالة التحوير المستخدمة لإنشاء التوقيع الرقمي، يتطلب المعيار استخدام الدالة SHA-1 بطول 160 بتاً، كما يوصي بدعمٍ لدى المُستلِم لدالة MD5 بطول 128 بتاً لضمان التوافق مع الإصدارات القديمة من S/MIME. كما سبق وناقشنا في الفصل الثالث، ويوجد قلقٌ له ما يبرره فيما يتعلق بأمن MD5، ولذا فمن الواضح أن SHA-1 هي البديل المفضل.

لتشفير الرسالة، يوصَى باستخدام خوارزمية DES ذات المفاتيح الثلاثة (3DES)، ولكن يتعين على الأنظمة المتوافقة مع المعيار دعم خوارزمية RC2 بطول

40 بتاً. الخوارزمية الأخيرة هي خوارزمية تشفير ضعيفة ولكنها تتوافق مع قوانين الولايات المتحدة الأمريكية الخاصة بمراقبة الصادرات.

تتضمن مواصفات S/MIME مناقشة لإجراءات تحديد الخوارزمية التي ستُستخدَم لتشفير المحتوى. وباختصار يحتاج وكيل المُرسِل لاتخاذ قرارين. الأول: تحديد ما إذا كان بوسع الوكيل المُستلِم إزالة التشفير باستخدام خوارزمية تشفير بعينها. الثاني: إذا كان الوكيل المُستلِم يستطيع فقط قبول محتوى مُشفَّر تشفيراً ضعيفاً، فعلى الوكيل المُرسِل تحديد ما إذا كان من المقبول إرسال المحتوى بتشفيرٍ ضعيف. ولدعم عملية اتخاذ القرار تلك، يمكن أن يعلن الوكيل المُرسِل عن إمكانات إزالة التشفير المتوفرة لديه مرتبة حسب الأفضلية في كل رسالة يرسلها. ويمكن للوكيل المُستلِم تخزين تلك المعلومات لاستخدامها في المستقبل.

ينبغي على الوكيل المُرسِل اتباع القواعد الآتية بالترتيب:

1. إذا كان لدى الوكيل المُرسِل قائمةٌ بإمكانات إزالة التشفير المفضلة لدى مُستلِمٍ يعتزم الإرسال إليه، فينبغي عليه اختيار أول إمكانية على القائمة (أي الأعلى تفضيلاً) يكون بوسعه استخدامها.
2. إذا لم يتوفر لدى الوكيل المُرسِل قائمةٌ بإمكانات إزالة التشفير المفضلة لدى مُستلِمٍ يعتزم الإرسال إليه، ولكنه سبق أن تلقَّى رسالةً أو أكثر من ذلك المُستلِم، فينبغي عليه استخدام نفس خوارزمية التشفير التي استخدمها ذلك المُستلِم على آخر رسالةٍ موقَّعة ومُشفَّرة وصلت منه.
3. إذا لم يتوفر لدى الوكيل المُرسِل قائمةٌ بإمكانات إزالة التشفير المفضلة لدى مُستلِمٍ يعتزم الإرسال إليه، وكان مستعداً للمخاطرة بإرسال رسالةٍ لا يستطيع المُستلِم إزالة تشفيرها، فينبغي عليه استخدام خوارزمية DES ثلاثية المفاتيح.
4. إذا لم يتوفر لدى الوكيل المُرسِل قائمةٌ بإمكانات إزالة التشفير المفضلة لدى مُستلِمٍ يعتزم الإرسال إليه، ولم يكن مستعداً للمخاطرة بإرسال رسالةٍ

لا يستطيع المُستلِم إزالة تشفيرها، فيتعين عليه استخدام خوارزمية RC2 بطول 40 بتاً (RC2/40).

إذا كان المطلوب إرسال رسالةٍ إلى عدة مُستلِمين وتعذَّر اختيار خوارزمية تشفير مشتركة للجميع، فسيحتاج الوكيل المُرسِل لإرسال رسالتين. ومع ذلك، فمن المهم في هذه الحالة ملاحظة أن أمن الرسالة يتعرض للخطر من جرَّاء إرسال نسخة منها بأمن أقل.

5-2-4 الرسائل في معيار S/MIME

يستخدم معيار S/MIME عدداً من أنواع المحتوى الجديدة المبينة في الجدول 5-7. تستخدم كل أنواع التطبيقات الجديدة الرمز PKCS للإشارة إلى مجموعة من مواصفات التشفير بالمفاتيح العامة صدرت عن مختبرات RSA وأصبحت متاحة للاستخدام في جهود تطوير S/MIME.

الجدول 5-7: أنواع المحتوى في معيار S/MIME.

النوع	النوع الفرعي	متغير SMIME	الوصف
Multipart متعدد الأجزاء	Signed موقّعة		بياناتٌ موقعةٌ بدون تكويد تتألف من جزأين هما: الرسالة والتوقيع
Application تطبيق	pkcs 7-mime	signedData بيانات موقّعة	كائن S/MIME موقّع
	pkcs 7-mime	envelopedData بيانات مغلّفة	كائن S/MIME مُشفَّر
	pkcs 7-mime	degenerate signedData بيانات موقّعة انتكاسية	كائن S/MIME يحتوي فقط على شهادات مفاتيح عامة
	pkcs 7-mime	compressedData بيانات مضغوطة	كائن S/MIME مضغوط
	pkcs 7-signature pkcs 7- توقيع	signedData بيانات موقّعة	نوع المحتوى للجزء الفرعي الخاص بالتوقيع ضمن رسالةٍ متعددة الأجزاء/موقّعة.

سوف نفحص كلاً من تلك الأنواع بالتفصيل، وذلك بعد أن نستعرض أولاً الإجراءات العامة لإعداد رسائل S/MIME.

❖ تأمين كيان MIME

يقوم معيار S/MIME بتأمين كيان MIME باستخدام التوقيع، أو التشفير، أو كليهما معاً. يمكن أن يكون كيان MIME رسالةً كاملةً (باستثناء ترويسات RFC 822)، وإذا كان نوع محتوى MIME متعدد الأجزاء يكون كيان MIME واحداً أو أكثر من الأجزاء الفرعية للرسالة. يتم إعداد كيان MIME وفقاً للقواعد المعتادة لتحضير رسائل MIME. بعد ذلك يقوم S/MIME بمعالجة كيان MIME بعد أن يضاف إليه بعض البيانات المتعلقة بالأمن، مثل مُعرِّفات الخوازميات والشهادات، وذلك لإنتاج ما يُعرَف بكائن PKCS. بعد ذلك يُعامَل كائن PKCS كمحتوى رسالة ويغلَّف في MIME (أي تضاف إليه ترويسات MIME المناسبة). وسوف تتضح تلك العملية بعد أن نستعرض كائناتٍ محددةً ونتناول بعض الأمثلة.

في كل الحالات يتم تحويل الرسالة المطلوب إرسالها إلى الصيغة القانونية (canonical form). وبالتحديد لنوعٍ معينٍ ونوعٍ فرعيٍ معين، تُستخدم الصيغة القانونية المناسبة لمحتوى الرسالة. وللرسالة متعددة الأجزاء، تُستخدم الصيغة القانونية المناسبة لكل جزءٍ فرعيٍ من أجزاء الرسالة.

يحتاج استخدام تكويد النقل (transfer encoding) إلى عنايةٍ خاصة. وفي معظم الحالات، سيؤدي تطبيق خوارزمية الأمن إلى إنتاج كائن مُمثَّل جزئياً أو كلياً على شكل بياناتٍ ثنائيةٍ اعتباطية. بعدها يتم تغليفه في رسالة MIME خارجية ويمكن عندئذٍ القيام بتكويد النقل، والذي يَستخدِم عادةً نظام base64. لكن في حالة الرسالة متعددة الأجزاء – وسنتناولها لاحقاً بتفصيل أكثر – فإن محتوى الرسالة في واحدٍ من الأجزاء الفرعية لا يتغير بفعل العملية الأمنية. وإذا لم يكن هذا المحتوى بصيغة 7 بتات، فينبغي استخدام تكويد النقل لتحويله إلى base64 أو quoted-printable، وذلك لتجنب خطر تغيير المحتوى الذي تم توليد التوقيع على أساسه.

سنلقي الآن نظرةً على أنواع المحتوى في S/MIME.

❖ البيانات المغلّفة (EnvelopedData):

يُستخدم النوع الفرعي application/pkcs7-mime في واحدةٍ من أربع فئاتٍ للمعالجة في معيار S/MIME لكلٍّ منها متغير smime-type فريد. وفي كل الحالات يُمثَّل الكيان (object) الناتج بصيغة تُعرَف بقواعد التكويد الأساسية (Basic Encoding Rules (BER))، والمعرّفة في توصية الاتحاد الدولي للاتصالات (ITU-T) رقم X.209. تتكون صيغة BER من سلسلة من البايتات الاعتباطية ومن ثم تُعدُّ بيانات ثنائية. وينبغي معالجة مثل هذ الكيان بتكويد النقل base64 في رسالة MIME الخارجية. وسنتناول أولاً البيانات المغلَّفة (envelopedData).

فيما يلي الخطوات اللازمة لإعداد كيان MIME من نوع envelopedData:

1. قم بتوليد مفتاح جلسة شبه عشوائي لخوارزمية تشفير متماثل معينة (RC2/40 أو 3DES).
2. لكل مُستلِم، قم بتشفير مفتاح الجلسة بالمفتاح العام لخوارزمية RSA للمُستلِم.
3. لكل مُستلِم، قم بتحضير كتلة تُعرَف باسم RecipientInfo (معلومات المُستلِم) تتضمن مُعرِّفاً لشهادة المفتاح العام للمُستلِم [3]، ومُعرِّفاً للخوارزمية المستخدمة لتشفير مفتاح الجلسة، ومفتاح الجلسة المُشفَّر.
4. قم بتشفير محتوى الرسالة بمفتاح الجلسة.

يتشكل الكيان envelopedData من كتل RecipientInfo متبوعة بالمحتوى المُشفَّر. يتم بعد ذلك تكويد تلك البيانات باستخدام base64. فيما يلي عينة من الرسالة (باستثناء ترويسة RFC 822):

---

[3] تلك هي شهادة X.509 التي سنناقشها في وقت لاحق من هذا الجزء.

```
Content-Type: application/pkcs7-mime; smime-type=enveloped-data;
name=smime.p7m
Content-Transfer-Encoding: base64
Content-Disposition: attachment; filename=smime.p7m
rfvbnj75.6tbBghyHhHUujhJhjH77n8HHGT9HG4VQpfyF467GhIGfHfYT6
7n8HHGghyHhHUujhJh4VQpfyF467GhIGfHfYGTrfvbnjT6jH7756tbB9H
f8HHGTrfvhJhjH776tbB9HG4VQbnj7567GhIGfHfYT6ghyHhHUujpfyF4
0GhIGfHfQbnj756YT64V
```

لاستعادة الرسالة المُشفَّرة، يقوم المُستلِم أولاً بالتخلص من تكويد base64. بعد ذلك يستخدم المُستلِم مفتاحه الخاص لاستعادة مفتاح الجلسة. وأخيراً يقوم بإزالة تشفير محتوى الرسالة باستخدام مفتاح الجلسة.

❖ البيانات الموقّعة (SignedData):

في الواقع، يمكن استخدام النوع signedData (بيانات موقَّعة) من أنوع المحتوى smime-type مع واحد أو أكثر من الموقِّعين. وللتبسيط سوف نقصر مناقشتنا هنا على حالة توقيع رقمي واحد.

فيما يلي الخطوات اللازمة لإعداد كيان MIME من نوع signedData:

1. اختر خوارزمية لحساب خلاصة الرسالة (SHA أو MD5).
2. احسب خلاصة الرسالة، أو دالة التحوير، لمحتوى الرسالة المطلوب توقيعه.
3. قم بتشفير خلاصة الرسالة بالمفتاح الخاص للموقِّع.
4. قم بتحضير كتلة تُعرَف بـ SignerInfo (معلومات الموقِّع) تتضمن معرِّفاً لشهادة المفتاح العام للموقِّع، ومعرِّفاً لخوارزمية حساب خلاصة الرسالة، ومعرِّفاً للخوارزمية المستخدمة لتشفير خلاصة الرسالة، والخلاصة المُشفَّرة للرسالة.

يتألف الكيان signedData من سلسلة من الكتل، تضم معرِّفاً لخوارزمية حساب خلاصة الرسالة، والرسالة الجاري توقيعها، والكتلة SignerInfo. ويمكن

أن يضم الكيان signedData أيضاً مجموعة من شهادات المفاتيح العامة كافية لتشكيل سلسلة تبدأ من أصل مُعترَف به أو من سلطة تصديق على مستوى عالٍ وتنتهي بالموقِّع. يتم بعد ذلك تكويد تلك البيانات باستخدام base64. وفيما يلي عينة من الرسالة (باستثناء ترويسة RFC 822):

```
Content-Type: application/pkcs7-mime; smime-type=signed-data;
name=smime.p7m
Content-Transfer-Encoding: base64
Content-Disposition: attachment; filename=smime.p7m
567GhIGfHfYT6ghyHhHUujpfyF4f8HHGTrfvhJhjH776tbB9HG4VQbnj7
77n8HHGT9HG4VQpfyF467GhIGfHfYT6rfvbnj756tbBghyHhHUujhJhjH
HUujhJh4VQpfyF467GhIGfHfYGTrfvbnjT6jH7756tbB9H7n8HHGghyHh
6YT64V0GhIGfHfQbnj75
```

لاستعادة الرسالة الموقَّعة والتحقق من التوقيع، يقوم المُستلِم أولاً بالتخلص من تكويد base64. بعد ذلك يستخدم المُستلِم المفتاح العام للموقِّع لإزالة تشفير خلاصة الرسالة. يحسب المُستلِم بشكلٍ مستقل خلاصة الرسالة ويقارنها بخلاصة الرسالة الواصلة بعد فك تشفيرها للتحقق من التوقيع.

❖ توقيع رسائل بدون تكويد (Clear Signing):

يتم توقيع رسائل بدون تكويد باستخدام نوع المحتوى "متعدد الأجزاء" (multiple) والنوع الفرعي "موقَّع" (signed). كما ذكرنا آنفاً، لا تتضمن عملية التوقيع هذه عملية تحويل الرسالة المطلوب توقيعها، ومن ثم يتم إرسال الرسالة "بدون تكويد". ومن ثَمَّ فإن المُستلِمين الذين تتوفر لديهم إمكانات MIME وليس إمكانات S/MIME سيكون بوسعهم قراءة الرسالة الواردة.

تتألف الرسالة متعددة الأجزاء/موقَّعة (multipart/signed) من جزأين. ويمكن أن يكون الجزء الأول أي نوع محتوى MIME ولكن ينبغي إعداده بحيث لا يتعرض للتغيير أثناء نقله من المصدر إلى الوجهة النهائية. ويعني ذلك أنه إذا لم يكن الجزء الأول بصيغة 7 بتات فيتعين تكويده باستخدام base64 أو quoted-printable. بعد

ذلك يتم معالجة هذا الجزء بنفس الطريقة التي تعالَج بها البيانات الموقّعة (signedData)، ولكن في هذه الحالة يتم إنشاء كائن بصيغة signedData يكون فيه حقل محتوى الرسالة فارغاً. وذلك الكائن هو توقيع منفصل. ويتم بعد ذلك تكويده للنقل باستخدام base64 ليصبح الجزء الثاني من الرسالة. ولهذا الجزء الثاني محتوى من النوع application (تطبيق) والنوع الفرعي pkcs7-signature. وفيما يلي عينة رسالة:

```
Content-Type: multipart/signed;
protocol=”application/pkcs7-signature”;
micalg=sha1; boundary=boundary42
--boundary42
Content-Type: text/plain
This is a clear-signed message.
--boundary42
Content-Type: application/pkcs7-signature; name=smime.p7s
Content-Transfer-Encoding: base64
Content-Disposition: attachment; filename=smime.p7s
ghyHhHUujhJhjH77n8HHGTrfvbnj756tbB9HG4VQpfyF467GhIGfHfYT6
4VQpfyF467GhIGfHfYT6jH77n8HHGghyHhHUujhJh756tbB9HGTrfvbnj
n8HHGTrfvhJhjH776tbB9HG4VQbnj7567GhIGfHfYT6ghyHhHUujpfyF4
7GhIGfHfYT64VQbnj756
--boundary42--
```

يشير متغير البروتوكول إلى أن هذا كيان "غير مكوّد وموقّع" مؤلف من جزأين. يبيّن المتغير micalg نوع خلاصة الرسالة المستخدمة. ويمكن للمُستلِم التحقق من التوقيع بأخذ خلاصة الرسالة للجزء الأول ومقارنتها بخلاصة الرسالة التي يتم استعادتها من التوقيع في الجزء الثاني.

❖ **طلب التسجيل (Registration Request):**

في العادة يقدّم التطبيق أو المستخدم طلباً إلى هيئة توثيق الشهادات للحصول على شهادة مفتاح عام. ويُستخدَم الكيان application/pkcs10 من كائنات S/MIME في نقل طلب الشهادة. ويتضمن طلب الشهادة كتلة certificationRequestInfo (معلومات طلب الشهادة) يتبعها معرِّف لخوارزمية تشفير المفتاح العام، ثم التوقيع على الكتلة certificationRequestInfo الذي يتم إنشاؤه باستخدام المفتاح الخاص للمُرسِل. تتضمن الكتلة certificationRequestInfo اسماً لموضوع الشهادة (الكيان المطلوب توثيق مفتاحه العام) وسلسلة بتات تمثل المفتاح العام للمُستخدِم.

❖ **رسالة "الشهادات فقط" (Certificates-Only):**

يمكن إرسال رسالة تحتوي فقط على شهادات أو على قائمة إلغاء شهادات (Certificate Revocation List (CRL)) استجابةً لطلب تسجيل. وهذه الرسالة من النوع application/pkcs7-mime، ولها متغير من نوع smime-type قيمته degenerate. والخطوات اللازمة هي نفس خطوات إنشاء رسالة بيانات موقَّعة signedData، لكن لا يوجد محتوى للرسالة ويكون حقل signerInfo فارغاً.

5-2-5 معالجة الشهادة في S/MIME

يستخدم معيار S/MIME شهادات مفاتيح عامة متوافقة مع الإصدار 3 من X.509 (انظر الفصل الرابع). وفي بعض النواحي يُعدُّ نظام إدارة المفاتيح الذي يستخدمه S/MIME هجيناً من التوثيق الهرمي الصارم في X.509 والشبكة العنكبوتية للثقة في PGP. وكما هو الحال في نموذج PGP، يجب على المديرين والمستخدمين في S/MIME تزويد كل زبون بقائمة من المفاتيح الموثوقة وقوائم إلغاء الشهادات. أي أن المسؤولية محلية في الحفاظ على الشهادات اللازمة للتحقق من

التوقيعات الواردة ولتشفير الرسائل الصادرة. ومن جانب آخر، يتم توقيع الشهادات من سلطات التصديق.

❖ دور وكيل المستخدم:

يضطلع مستخدم S/MIME بعدة مهام تتعلق بإدارة المفاتيح نلخصها فيما يلي:

- توليد المفاتيح: يتعين على مستخدم بعض المرافق الإدارية ذات الصلة (كتلك المرتبطة بإدارة الشبكة المحلية) أن يكون قادراً على توليد أزواج منفصلة لمفاتيح ديفي - هيلمان وDSS، كما ينبغي أن يكون قادراً على توليد أزواج مفاتيح RSA. يتعين توليد كل زوج من المفاتيح من مصدر جيد لمُدخل غير قطعي (nondeterministic) من الأعداد العشوائية ويجب حمايته بشكلٍ آمن. ينبغي على وكيل المُستلِم توليد أزواج مفاتيح RSA بطول في الحدود من 768 إلى 1024 بتاً، ويتعين ألا يولد مفاتيح بطول أقل من 512 بتاً.
- التسجيل: يجب أن يكون المفتاح العام المُستخدَم مسجلاً لدى سلطة تصديق للحصول على شهادة X.509 للمفتاح العام.
- تخزين واستعادة الشهادات: يحتاج المُستخدِم للوصول إلى قائمةٍ محلية للشهادات لكي يتمكن من التحقق من التوقيعات الواردة وتشفير الرسائل الصادرة. يمكن الاحتفاظ بهذه القائمة لدى المُستخدِم أو لدى كيانٍ محليٍ بالنيابة عن عددٍ من المستخدمين.

❖ شهادات VeriSign:

يقوم عدد من الشركات بتوفير خدمات سلطة تصديق (certification authority (CA)). على سبيل المثال، صممت شركة نورتل حلاً لسلطات التصديق لاستخدام الشركات يمكنه توفير الدعم لـ S/MIME داخل المنظمة. هناك عدد من مؤسسات التصديق القائمة على الإنترنت، بما في ذلك VeriSign وGTE ومصلحة بريد الولايات

المتحدة الأمريكية. تُعدُّ خدمة VeriSign لتصديق الشهادات أكثر تلك الخدمات انتشاراً، وسوف نقدم هنا وصفاً موجزاً لها.

توفر VeriSign خدمةً لتصديق الشهادات متوافقة مع S/MIME وعِدّة تطبيقات أخرى غيرها. تُصدر VeriSign شهادات X.509 تحت اسم المنتج VeriSign Digital ID (هوية VeriSign الرقمية). بحلول أوائل عام 1998، كان هناك ما يزيد على 35,000 موقع ويب تجاري يستخدم هويات VeriSign الرقمية للخادمات، كما تم إصدار أكثر من مليون هوية VeriSign رقمية لمستخدمي متصفحات نيتسكيب ومايكروسوفت.

تعتمد المعلومات الواردة في هوية VeriSign الرقمية على نوع الهوية واستخدامها. كحد أدنى، تتضمن كلُّ هويةٍ رقميةٍ ما يلي:

- المفتاح العام للمالك.
- الاسم أو الاسم المستعار للمالك.
- تاريخ انتهاء الهوية الرقمية.
- الرقم التسلسلي للهوية الرقمية.
- اسم سلطة التصديق التي أصدرت الهوية الرقمية.
- التوقيع الرقمي لسلطة التصديق التي أصدرت الهوية الرقمية.

يمكن أن تتضمن الهويات الرقمية أيضاً معلوماتٍ أخرى يوفرها المستخدم مثل:

- العنوان.
- عنوان البريد الإلكتروني.
- معلومات التسجيل الأساسية (البلد، والرمز البريدي، والعمر، والجنس).

الجدول 5-8: فئات شهادات VeriSign للمفاتيح العامة.

	موجز إثبات الهوية	حماية المفتاح الخاص لسلطة الإصدار (IA)	حماية المفتاح الخاص للمتقدم للحصول على الشهادة والمُشترك	التطبيقات التي ينفذها المستخدمون أو يفكرون في تنفيذها
الفئة 1	بحث آلي عن الاسم وعنوان البريد الإلكتروني بدون لبس	PCA: عتاد موثوق CA: برمجيات موثوقة أو عتاد موثوق	يوصى ببرمجيات تشفير (محمية بـ PIN)، ولكنها ليست مطلوبة	تصفّح الويب وبعض استخدامات البريد الإلكتروني
الفئة 2	كما في الفئة 1، علاوةً على التحقق الآلي من معلومات التسجيل، بالإضافة إلى التحقق الآلي من العنوان	PCA وCA: عتاد موثوق	مطلوب برمجيات تشفير (محمية بـ PIN)	البريد الإلكتروني الفردي وداخل الشركة أو بين الشركات، الاشتراك على الخط مباشرةً (online)، تبديل كلمة السر، والتحقق من البرمجيات.
الفئة 3	كما في الفئة 1، علاوةً على التواجد الشخصي ووثائق الهوية، بالإضافة إلى التحقق الآلي كما في الفئة 2 من هويات الأفراد والسجلات التجارية (أو الملفات) للمنظمات	PCA وCA: عتاد موثوق	مطلوب برمجيات تشفير (محمية بـ PIN)؛ يوصى بعلامة عتاد (hardware token) ولكنها ليست مطلوبة	المعاملات البنكية الإلكترونية، الوصول إلى قواعد البيانات، المعاملات البنكية الشخصية، الخدمات المتاحة للمشتركين على الخط مباشرةً، خدمات التحقق من سلامة المحتوى، خادم التجارة الإلكترونية، التحقق من البرمجيات، توثيق LRAA، والتشفير القوي لخوادم بعينها.

IA : سلطة إصدار
CA : سلطة توثيق
PCA : سلطة التوثيق الرئيسة العامة لـ VeriSign
PIN : رقم التعريف الشخصي
LRAA : مدير سلطة التسجيل المحلي

توفر VeriSign ثلاثة مستويات أو فئات من الأمن لشهادات المفاتيح العامة، على النحو المبين بإيجاز في الجدول 5-8. يطلب المستخدِم شهادةً على الإنترنت من موقع VeriSign أو مواقع الإنترنت الأخرى المشاركة. وتتم معالجة الطلبات من الفئة 1

والفئة 2 على الخط مباشرةً (online)، وفي معظم الحالات لا تستغرق عملية الاعتماد بضع ثوانٍ. وباختصار، تُتَّبع الإجراءات التالية:

- للهويات الرقمية من الفئة 1، تتحقق VeriSign من عنوان البريد الإلكتروني للمستخدم من خلال إرسال رقم التعريف الشخصي (PIN) ومعلومات التقاط الهوية الرقمية إلى عنوان البريد الإلكتروني الوارد في الطلب.
- للهويات الرقمية من الفئة 2، تتحقق VeriSign من المعلومات الواردة في الطلب آلياً من خلال المقارنة مع قاعدة بيانات الزبائن، بالإضافة إلى القيام بكل خطوات التحقق المرتبطة بالهويات الرقمية من الفئة 1. وأخيراً يُرسَل تأكيدٌ إلى العنوان البريدي المحدّد في الطلب لتنبيه المستخدم إلى أنه قد تم إصدار هوية رقمية باسمه.
- للهويات الرقمية من الفئة 3، تحتاج VeriSign إلى مستوى أعلى من الإجراءات للتحقق من الهوية، حيث يتعين على مقدم الطلب إثبات هويته من خلال تقديم أوراق اعتماده موثقة أو التقدم بالطلب شخصياً.

6-2-5 خدمات الأمن المحسَّنة

حتى وقت كتابة هذا الجزء، تم اقتراح ثلاث خدمات أمنية محسَّنة في مسودةٍ للإنترنت. قد تتغير تفاصيل تلك الخدمات، كما قد تضاف خدمات أخرى. والخدمات الثلاث هي:

- إيصالات موقَّعة (Signed receipts): يُمكن أن يُطلَب إيصالٌ موقَّعٌ ضمن كائن SignedData. عند الرد بإيصالٍ موقَّع، يتم توفير إثبات بالاستلام لمنشئ الرسالة يُمكِّن المنشئ من أن يُثبت لطرف ثالث أن المُستلِم قد تلقّى الرسالة. تتلخص العملية في قيام المُستلِم بتوقيع الرسالة الأصلية بكاملها مضافاً إليها التوقيع الأصلي (أي توقيع المُرسِل)، ويضيف التوقيع الجديد لتكوين رسالة S/MIME جديدة.

- وسْمات الأمن (Security labels): يمكن إدراج الوسمة الأمنية ضمن الصفات الموثقة لكائن بياناتٍ موقَّعة (SignedData). تشمل تلك الوسمة مجموعةً من المعلومات الأمنية بخصوص حساسية المحتوى الذي تتم حمايته بتغليف S/MIME. يمكن استخدام الوسمة الأمنية للتحكم في الوصول، وذلك ببيان أي المستخدمين يمكنهم الوصول إلى الكائن. ومن بين استخداماتها الأخرى تحديد الأولوية (سري، مقيّد، وغير ذلك)، أو تحديد الفئة من الناس الذين يمكنهم الاطلاع على المعلومات بناءً على الدور الذي يقومون به (مثلاً فريق الرعاية الصحية للمريض، ووكلاء الفواتير الطبية، وغير ذلك).
- قوائم البريد الآمنة (Secure mailing lists): عند قيام مُستخدِم بإرسال رسالةٍ الى عدة مُستلِمين، يتطلب الأمر قدراً معيّناً من المعالجة لكل واحدٍ منهم، بما في ذلك استخدام المفتاح العام لكل مُستلِم. يمكن إعفاء المُستخدِم من القيام بهذا العمل عن طريق الاستعانة بخدمات وكيل S/MIME لقوائم البريد (Mail List Agent (MLA)). يمكن لوكيل قوائم البريد أخذ رسالة واحدة والقيام بإجراء التشفير المتعلق بكل مُستلِم ثم تمرير الرسالة. سيحتاج منشئ الرسالة فقط إلى إرسال الرسالة إلى وكيل قوائم البريد مُشفَّرةً بالمفتاح العام لذلك الوكيل.

3-5 مصادر للمعلومات على الويب

- صفحة PGP الرئيسة: موقع PGP على الويب، والتابع لشركة PGP، وهي الشركة الرائدة في مجال توريد PGP تجارياً.
- صفحة PGP الرئيسة الدولية: صُمّمت للترويج لاستخدام PGP في جميع أنحاء العالم. وتتضمن وثائق وروابط مهمة.
- موقع معهد ماساتشوستس للتكنولوجيا لتوزيع PGP: موقع رائد للتوزيع المجاني لـ PGP. يتضمن إجاباتٍ عن الأسئلة المشهورة، ومعلوماتٍ إضافية، وروابط لمواقع PGP أخرى.
- ميثاق PGP: أحدث وثائق RFC ومُسودات الإنترنت للمواصفات المفتوحة لـ PGP.
- ميثاق S/MIME: أحدث وثائق RFC ومُسودات الإنترنت للمواصفات المفتوحة لـ S/MIME.

4-5 مصطلحات رئيسة

detached signature	توقيع منفصل
electronic mail	البريد الإلكتروني
Multipurpose Internet Mail Extensions (MIME)	امتدادات بريد الإنترنت متعددة الأغراض
Pretty Good Privacy (PGP)	خصوصية جيدة جداً
radix-64	الأساس 64
session key	مفتاح الجلسة
Secure Multipurpose Internet Mail Extensions (S/MIME)	امتدادات بريد الإنترنت متعددة الأغراض الآمنة
trust	ثقة
ZIP	خوارزمية ضغط البيانات ZIP

5-5 أسئلة مراجعة ومسائل

5-5-1 أسئلة للمراجعة

5-1 ما الوظائف الخمس الرئيسة التي يوفرها بروتوكول PGP؟

5-2 ما الفائدة من استخدام توقيع منفصل؟

5-3 لماذا يتم توليد التوقيع في PGP قبل القيام بعملية الضغط؟

5-4 عرِّف المقصود بالتحويل للأساس 64.

5-5 لماذا يُعدُّ التحويل للأساس 64 مفيداً لتطبيقات البريد الإلكتروني؟

5-6 لماذا نحتاج إلى عمليتي تقطيع البيانات وإعادة تجميعها في PGP؟

5-7 كيف يستخدم بروتوكول PGP مفهوم الثقة؟

5-8 عرِّف طلب التعليقات RFC 822.

5-9 عرِّف معيار MIME.

5-10 عرِّف معيار S/MIME.

5-5-2 مسائل

5-1 يستخدم بروتوكول PGP نمط التغذية المرتدَّة للشفرة (CFB) لخوارزمية CAST-128، بينما تستخدم معظم تطبيقات التشفير المتماثل (باستثناء تشفير المفاتيح) نمط سلسلة كتل الشفرة (CBC). لدينا:

$$\text{CBC: } C_i = E(K, [C_{i-1} \oplus P_i]);$$
$$P_i = C_{i-1} \oplus D(K, C_i)$$
$$\text{CFB: } C_i = P_i \oplus E(K, C_{i-1});$$
$$P_i = C_i \oplus E(K, C_{i-1})$$

يظهر أن هذين الأسلوبين يوفران نفس المستوى من الأمن. اذكر سبباً لاستخدام PGP لنمط CFB.

5-2 في نظام PGP، ما العدد المتوسط من مفاتيح الجلسات التي يتم توليدها قبل إنتاج مفتاح سبق توليده؟

3-5 في نظام PGP، ما احتمال أن يحصل مستخدم لديه N مفتاحاً عاماً على مفتاح واحد مزدوج على الأقل؟

4-5 يتم إرسال أول 16 بتاً من خلاصة الرسالة في PGP بشكلٍ واضح (أي غير مُشفَّر).

a. إلى أي حد يقلل ذلك من مستوى الأمن الذي توفره خوارزمية التحويل؟

b. إلى أي حد ينجح ذلك في واقع الأمر في تحقيق الغرض المطلوب، ألا وهو المساعدة في تحديد ما إذا كان المفتاح الصحيح لخوارزمية RSA قد استُخدِم لإزالة تشفير الخلاصة؟

5-5 في الشكل 5-4، يتضمن كل مُدخل في حلقة المفاتيح العامة حقلاً يبيّن مدى الثقة بمالك ذلك المفتاح العام. لماذا لا يُعدُّ ذلك كافياً؟ بمعنى: إذا كان هذا المالك موثوقاً وهذا المفتاح يُفترض أنه المفتاح العام للمالك، فلماذا لا تكون تلك الثقة كافية للسماح لبروتوكول PGP باستخدام ذلك المفتاح العام؟

6-5 اعتبر التحويل للأساس 64 (R64) شكلاً من أشكال التشفير. في هذه الحالة لا يوجد أي مفتاح. ولكن افترض وجود خَصْم لا يعرف سوى أنه قد استُخدِم نوع من خوارزميات الاستبدال لتشفير النص الانجليزي ولم يخمّن أنها R64. ما مدى فعالية تلك الخوارزمية ضد تحليل الشفرة؟

7-5 اختار فيل زيمرمان لبروتوكول PGP ثلاث خوارزميات تشفير متماثل هي: IDEA، وDES ثلاثية المفاتيح، وCAST-128. اذكر الأسباب وراء ما إذا كان كلٌّ من خوارزميات التشفير المتماثل الآتية التي ورد وصفها في هذا الكتاب مناسبةً أو غير مناسبةٍ لبروتوكول PGP: DES، وDES بمفتاحين، وAES.

الملحق A-5

ضغط البيانات باستخدام خوارزمية ZIP

يستخدم بروتوكول PGP حزمة برمجيات لضغط البيانات تعرف بـ ZIP (كتبها جين- لاب جيلي ومارك أدلر وريتشارد ويلز). وهي حزمة برمجياتٍ مجانيةٍ مكتوبةٍ بلغة C يمكن تشغيلها كبرنامج خدمي (utility) على نظام التشغيل يونيكس وبعض أنظمة التشغيل الأخرى. من حيث الوظائف التي تؤديها، تُعدُّ ZIP مكافئةً لـ PKZIP التي قامت بتطويرها شركة PKWARE ووفرتها على نطاق واسع كحزمة برمجيات مجانية لأنظمة تشغيل ويندوز. لعل خوارزمية ZIP هي أكثر أساليب ضغط البيانات شيوعاً للعمل عبر منصات التشغيل المختلفة؛ وتوجد منها إصدارات مجانية وأخرى للمشاركة (shareware) للماكنتوش وغيرها من النظم، بما في ذلك أنظمة ويندوز ويونيكس.

نشأت ZIP والخوارزميات المماثلة لها من الأبحاث التي قام بها يعقوب زيف وإبراهام ليمبل. ففي عام 1977 قاما بوصف أسلوب يعتمد على استعمال مخزن مؤقت (buffer) يستخدم نافذة منزلقة (sliding window) تتضمن أحدث نص تمت معالجته [ZIV77]. تُعرف تلك الخوارزمية عموماً بخوارزمية LZ77، وتُستخدم نسخة منها في أنظمة ZIP لضغط البيانات (PKZIP، وgzip، وzipit، إلخ).

تستغل خوارزميات LZ77 ومشتقاتها حقيقة أن الكلمات والعبارات ضمن تسلسل نص معيّن (أو أنماط الصورة في حالة GIF) يُحتمل أن تتكرر. وعند حدوث تكرار، يمكن استبدال التسلسل المكرّر بكود قصير.

يقوم برنامج الضغط بمسح النص بحثاً عن ذلك التكرار ويُنتج أثناء المسح أكواداً تحل محل التسلسل المكرّر. مع مرور الوقت، يُعاد استخدام تلك الأكواد لتمثيل تسلسلات جديدة. وينبغي تحديد الخوارزمية بحيث يمكن لبرنامج إزالة الضغط استخلاص المقابلة الحالية بين الأكواد والتسلسلات المناظرة في مصدر البيانات.

قبل البحث في تفاصيل LZ77، لنلقِ نظرة على مثال بسيط.[4] لنأخذ بعين الاعتبار العبارة الآتية:

the brown fox jumped over the brown foxy jumping frog

يبلغ طول هذه العبارة 53 بايتاً (أي 424 بتاً). تقوم الخوارزمية بمعالجة هذا النص من اليسار إلى اليمين. في البداية، يُمثَّل كل حرف بنمط من 9 بتات يتألف من بت قيمته 1 يتبعه كود الآسكي المناظر للحرف بطول 8 بتات. مع مواصلة المعالجة، تبحث الخوارزمية عن تسلسلاتٍ مكرّرة. عند مصادفة تكرارٍ تواصل الخوارزمية عملية المسح إلى أن ينتهي الجزء المكرّر. وبعبارة أخرى، في كل مرة يحدث تكرار، تضم الخوارزمية أكبر عدد ممكن من الحروف. أول تسلسل مكرر تتم مصادفته هو التسلسل the brown fox. يُستبدل هذا التسلسل المكرر بمؤشر للتسلسل السابق وبطول التسلسل. في هذه الحالة يقع التسلسل السابق the brown fox قبل 26 حرفاً ويبلغ طول التسلسل 13 حرفاً. في هذا المثال، افترض خيارين للتكويد؛ مؤشر مقاسه 8 بتات وطول تسلسل مقاسه 4 بتات أو مؤشر 12 بتاً وطول 6 بتات؛ بالإضافة إلى ترويسة طولها بتان للدلالة على الخيار المستخدم، حيث تشير 00 للخيار الأول و01 للخيار الثاني. وهكذا، فإن التكرار الثاني لـ the brown fox يتم تكويده كـ $<00_b> <26_d> <13_d>$، أي 1101 00011010 00.

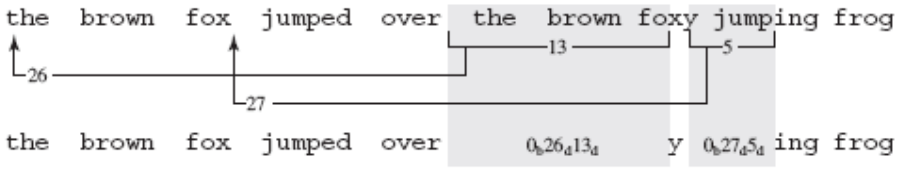
الأجزاء المتبقية من الرسالة المضغوطة هي الحرف y؛ والتسلسل $<00_b> <27_d> <5_d>$ يحل محل التسلسل الذي يضم حرف المسافة تليه jump؛ وتسلسل الحروف ing frog.

[4] مبني على مثال في [WEIS93].

يوضّح الشكل 5-9 التقابلات المستخدمة في ضغط الرسالة. تتألّف الرسالة المضغوطة من 35 حرفاً كل منها بطول 9 بتات بالإضافة إلى كودين، أي بمجموع كلي قدره

$$35 \times 9 + 2 \times 14 = 343 \text{ بتاً}$$

وذلك مقابل 424 بتاً للرسالة غير المضغوطة، أي بنسبة ضغط قدرها 1.24.



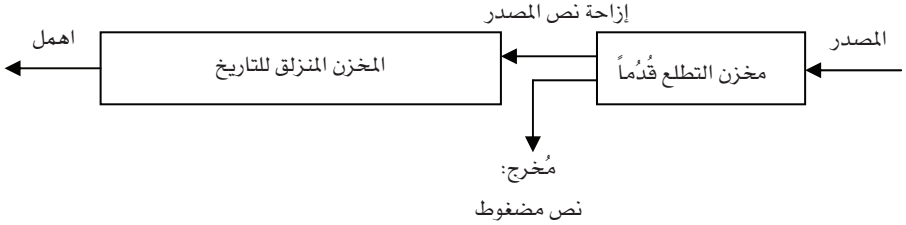
الشكل 5-9: مثال على نظام LZ77.

خوارزمية الضغط

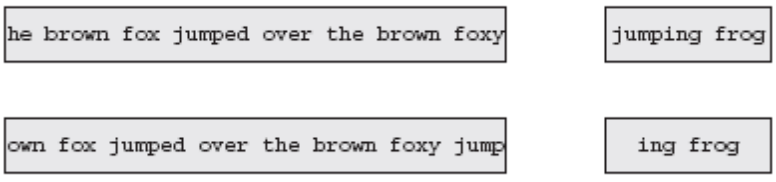
تستخدم خوارزمية الضغط LZ77 ومشتقاتها اثنين من المخازن المؤقتة (buffers). يضم المخزن المنزلق للتاريخ (sliding history buffer) آخر N حرف تم معالجتها من بيانات المصدر، بينما يضم مخزن التطلع قُدُماً (look-ahead buffer) الـ L حرف التالية (انظر الشكل 5-10 (a)). تحاول الخوارزمية إيجاد مطابقة بين اثنين أو أكثر من الحروف في بداية مخزن التطلع قُدُماً وتسلسل الحروف ضمن المخزن المنزلق للتاريخ. إذا تعذَّر وجود مطابقة يتم إخراج أول حرف في مخزن التطلع قُدُماً على شكل 9 بتات، حيث يتم إضافته إلى النافذة المنزلقة، بينما يُستبعَد أقدم حرف في النافذة المنزلقة. إذا وُجدَت مطابقةً، تواصل الخوارزمية عملية المسح لتعظيم طول تلك المطابقة. بعد ذلك يتم إخراج التسلسل المطابق على شكل ثلاثة متغيرات (مبيّن الخيار المستخدم، ومؤشر، وطول). عند الحصول على مطابقة

لتسلسل طوله K حرف، تتم إزاحة أقدم K حرف في النافذة المنزلقة إلى الخارج، بينما يتم إزاحة تسلسل المطابقة الذي تم تكويده بطول K حرف إلى داخل تلك النافذة.

يبيّن الشكل 5-10 (b) عمل هذا النظام على مثال تسلسل النص الذي نحن بصدده. نفترض في الشكل استخدام نافذة منزلقة بطول 39 حرفاً ومخزناً للتطلع قُدُماً بطول 13 حرفاً. في الجزء العلوي من الشكل، تم معالجة أول 40 حرفاً وتوجد النسخة غير المضغوطة من أحدث 39 حرفاً من تلك الحروف في النافذة المنزلقة. يوجد ما تبقّى من بيانات المصدر في مخزن التطلع قُدُماً. تقوم خوارزمية الضغط بتحديد المطابقة التالية، فتزيح الحروف الخمسة لتسلسل المطابقة من مخزن التطلع قُدُماً الى النافذة المنزلقة، وتقوم بإخراج الكود الذي يمثل ذلك التسلسل. ويبيّن الجزء السفلي من الشكل الوضع في كلٍ من المخزنين المؤقتين بعد تلك العملية.



(a) الهيكل العام



(b) مثال

الشكل 5-10: نظام ضغط البيانات باستخدام LZ77.

رغم أن خوارزمية LZ77 فعَّالة ولها القدرة على التكيف مع طبيعة المُدخل الفعلي، فإنها تعاني من بعض العيوب. فالخوارزمية تستخدم نافذة محدودة الطول للبحث عن مطابقات في النص السابق. لكتل النص الطويلة جداً بالنسبة لمقاس النافذة، يتم إغفال الكثير من المطابقات المحتملة. ويمكن زيادة حجم النافذة، ولكن هذه الطريقة لها عيبان: (1) زيادة وقت المعالجة الذي تستغرقه الخوارزمية حيث يتعيّن مقارنة تسلسل الحروف في مخزن التطلع قُدُماً مع محتويات النافذة المنزلقة عند كل موضع في تلك النافذة، و(2) يجب أن يكون حقل <المؤشر> أكبر لكي يستوعب أطول القفزات إلى مواضع المطابقة.

خوارزمية إزالة الضغط

إزالة الضغط من نص مضغوط بخوارزمية LZ77 هي عملية بسيطة. فعلى خوارزمية إزالة الضغط تخزين آخر N حرف من النص الذي تم إزالة ضغطه. وعند مصادفة تسلسل مُكوَّد، تستخدم الخوارزمية حقلي <المؤشر> و<الطول> لاستبدال الكود بتسلسل النص الفعلي المناظر له.

الملحق B-5

التحويل للأساس 64

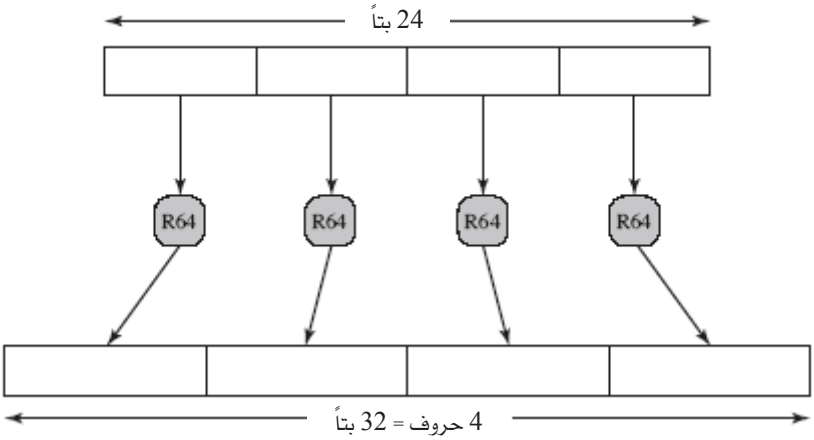
يستخدم كلٌّ من PGP وS/MIME أسلوب تكويد يُعرف بالتحويل للأساس 64، حيث يتم استبدال البيانات الثنائية الداخلة ذات القيم الاعتباطية بمُخرج يمثل التسلسل المقابل من الحروف القابلة للطباعة. تتسم عملية التكويد المستخدمة بالخصائص الآتية:

1. مجال الدالة المستخدمة هو طاقم حروف يمكن تمثيله عالمياً وفي جميع المواقع، وليس كوداً ثنائياً معيناً لذلك الطاقم من الحروف. وعليه، فإن الحروف نفسها يمكن تكويدها بأي شكلٍ من الأشكال التي يتطلبها نظام معيّن. فعلى سبيل المثال، يُمثَّل الحرف "E" في نظام مبني على الكود ASCII بالعدد الست عشري 45، بينما يُمثَّل في نظام مبني على الكود EBCDIC بالعدد الست عشري C5.
2. يتألَّف طاقم الحروف من 65 حرفاً قابلة للطباعة، يُستخدم أحدها للحشو. في وجود $2^6 = 64$ حرفاً، يمكن استخدام كل حرف لتمثيل 6 بتات من البيانات الثنائية الداخلة.
3. لا يتضمن طاقم الحروف أي حروف تحكم (control characters). وعليه، تستطيع الرسالة المكوَّدة في الأساس 64 اجتياز أنظمة معالجة البريد التي تمسح تدفق البيانات بحثاً عن حروف التحكم.
4. لا يُستخدم حرف الشّرطة " - ". لهذا الحرف أهميته في صيغة RFC 822، ولذلك ينبغي تجنبه.

ويبيّن الجدول 5-9 المقابلة بين قيم المُدخل بطول 6 بتات والحرف المناظر ضمن طاقم الحروف المستخدم. يتألف طاقم الحروف من الحروف الأبجدية والأرقام من 0 إلى 9 بالإضافة إلى علامة زائد "+" و"/". يُستخدم "=" كحرف حشو.

الجدول 5-9: التكويد بالأساس 64.

الحرف المستخدم للتكويد	القيمة العشرية للبتات الستة	الحرف المستخدم للتكويد	القيمة العشرية للبتات الستة	الحرف المستخدم للتكويد	القيمة العشرية للبتات الستة	الحرف المستخدم للتكويد	القيمة العشرية للبتات الستة
A	0	Q	16	g	32	W	48
B	1	R	17	h	33	X	49
C	2	S	18	i	34	Y	50
D	3	T	19	j	35	Z	51
E	4	U	20	k	36	0	52
F	5	V	21	l	37	1	53
G	6	W	22	m	38	2	54
H	7	X	23	n	39	3	55
I	8	Y	24	o	40	4	56
J	9	Z	25	p	41	5	57
K	10	a	26	q	42	6	58
L	11	b	27	r	43	7	59
M	12	c	28	s	44	8	60
N	13	d	29	t	45	9	61
O	14	e	30	u	46	+	62
P	15	f	31	v	47	/	63
						=	(للحشو)



الشكل 5-11: تكويد البيانات الثنائية على شكل حروف قابلة للطباعة باستخدام صيغة الأساس 64.

يوضّح الشكل 5-11 نظام المقابلة البسيط المستخدم. تتم معالجة المُدخل الثنائي بأخذه على شكل كتل يضم كلٌّ منها 3 بايتات (أي 24 بتاً). يتم استبدال كل مجموعة من 6 بتات في الـ 24 بتاً بالحرف المناظر. كما هو مبين في الشكل، يتم تكويد الحروف بصيغة آسكي بـ 8 بتات لكل حرف. في هذه الحالة النمطية، يتم تمديد كل 24 بتاً في المُدخل إلى 32 بتاً في الناتج.

على سبيل المثال، افترض تسلسل النص الآتي:

00100011 01011100 10010001

ويمكن التعبير عنه بالصيغة الست عشرية 235C91. بترتيب هذا التسلسل ككتل كلٍّ منها 6 بتات نحصل على:

001000 110101 110010 010001

القيم العشرية المناظرة لتلك الكتل هي: 17 ,50 ,53 ,8. من الجدول 5-9 نحصل على تكويد الأساس 64 المناظر لتلك القيم كتسلسل الحروف: I1yR. إذا تم تمثيل تلك الحروف بصيغة آسكي بـ 8 بتات مع بت التكافؤ ثابتة عند القيمة 0، فإننا نحصل على:

01001001 00110001 01111001 01010010

أي 49317952 بالصيغة الست عشرية. باختصار:

البيانات الداخلة	
التمثيل الثنائي	00100011 01011100 10010001
التمثيل الست عشري	235C91
تكويد البيانات الداخلة بالأساس 64	
التمثيل كحروف	I1yR
التمثيل بصيغة آسكي (8 بتات، بت التكافؤ = 0)	01001001 00110001 01111001 01010010
التمثيل الست عشري	49317952

الملحق C-5

توليد الأعداد العشوائية في PGP

يستخدم بروتوكول PGP نظاماً قوياً ومعقداً لتوليد الأعداد العشوائية وشبه العشوائية (pseudorandom) لعدة أغراض. يولّد PGP أرقاماً عشوائية من محتوى ضربات المفاتيح التي يقوم بها المُستخدِم وتوقيتها، كما يولِّد أرقاماً شبه عشوائية من خوارزمية مبنية على المعيار ANSI X9.17. يستخدم PGP تلك الأعداد للأغراض الآتية:

- الأعداد العشوائية الحقيقية:
  - توليد أزواج المفاتيح لخوارزمية RSA.
  - توفير البذرة الأولية لمولِّد الأعداد شبه العشوائية.
  - توفير مُدخلات إضافية أثناء توليد الأعداد شبه العشوائية.
- الأعداد شبه العشوائية:
  - توليد مفاتيح الجلسات.
  - توليد المتجه الأوّليّ (IV) للاستخدام مع مفتاح الجلسة في نمط التشفير CFB.

الأعداد العشوائية الحقيقية

يحتفظ PGP بمخزن مؤقت للبتات العشوائية طوله 256 بايتاً. في كل مرة يتوقع PGP ضربة مفتاح من المُستخدِم يقوم بتسجيل الوقت الذي يبدأ فيه الانتظار بصيغة 32 بتاً. وعندما يتلقى ضربة المفتاح يُسجّل وقت الضغط على المفتاح وكذلك قيمة المفتاح المضروب بصيغة 8 بتات. تُستخدم المعلومات عن الوقت وعن المفتاح المضروب لتوليد مفتاح، ويُستخدم ذلك المفتاح بدوره لتشفير القيمة الحالية للمخزن المؤقت للبتات العشوائية.

الأعداد شبه العشوائية

يستخدم مولّد الأعداد شبه العشوائية بذرةً من 24 بايتاً ويقوم بتوليد مفتاح جلسةٍ بطول 16 بايتاً ومتجهاً أوّلياً بطول 8 بايتات، وكذلك بذرة جديدة لاستخدامها في الدورة التالية لتوليد عدد شبه عشوائي. وتَستخدِم الخوارزمية أسلوب تشفير متماثل يُعرَف باسم CAST-128. وتَستخدِم هياكل البيانات الآتية:

1. المُدخلات:

- randseed.bin (بطول 24 بايتاً لتمثيل البذرة العشوائية): إذا كان هذا الملف فارغاً، فإنه يُملأ بـ 24 بايتاً من القيم العشوائية الحقيقية.
- message (الرسالة): يكون مفتاح الجلسة والمتجه الأوّلي اللذان سيُستخدمان لتشفير رسالة هما نفسهما دالة في تلك الرسالة. يساعد ذلك في زيادة عشوائية كلٍّ من مفتاح الجلسة والمتجه الأوّلي، واذا كان الخَصْم يعرف بالفعل محتوى الرسالة غير المُشفَّرة، فلا حاجة ظاهرة للحصول على مفتاح الجلسة الذي يُستخدَم لمرة واحدة فقط.

2. المُخرجات

- K (بطول 24 بايتاً): تتضمن البايتات الستة عشر الأولى K[0..15] مفتاح الجلسة، بينما تتضمن البايتات الثمانية الأخيرة K[16..23] المتجه الأوّلي.
- randseed.bin (بطول 24 بايتاً): يحتوي هذا الملف على قيمة جديدة للبذرة العشوائية.

3. هياكل البيانات الداخلية

- dtbuf (بطول 8 بايتات): في البداية يتم تخزين القيمة الحالية للوقت والتاريخ في البايتات الأربعة الأولى dtbuf[0..3]. ويكافئ هذا المخزن المؤقت المتغير DT في خوارزمية X12.17.
- rkey (بطول 16 بايتاً): مفتاح تشفير CAST-128 المستخدم في جميع مراحل الخوارزمية.

- rseed (بطول 16 بايتاً): يكافئ المتغير Vi في خوارزمية X12.17.
- rbuf (بطول 8 بايتات): عددٌ شبه عشوائي تولّده الخوارزمية. ويكافئ هذا المخزن المؤقت المتغير Ri في خوارزمية X12.17.
- K' (بطول 24 بايتاً): مخزن مؤقت للقيمة الجديدة لـ randseed.bin.

تتألّف الخوارزمية من تسع خطوات: من G1 إلى G9. الخطوتان الأولى والأخيرة هما خطوتا تشويش بهدف تقليل القيمة التي يمثلها الملف randseed.bin إذا ما وقع في يد خَصْم. وتكافئ الخطوات المتبقية ثلاثة تكرارات لخوارزمية X12.17 كما هو مبين في الشكل 5-12. وباختصار:

G1. [الغسلة القَبلية للبذرة السابقة]

a. انسخ randseed.bin إلى K[0..23].

b. خذ دالة التحوير للرسالة (سيكون قد تم توليدها بالفعل إذا كان المطلوب توقيع الرسالة، وإلا فستستخدم أول 4 كيلو بايت من الرسالة). استخدم ناتج العملية مفتاحاً، استخدم متجه أوّلي (IV) مصفّر، وقم بتشفير K في نمط CFB؛ وخزّن النتيجة في K.

G2. [حدِّد البذرة الأولية]

a. ضع قيمة الوقت الحالي بطول 32 بتاً في dtbuf [0..3]. قم بتصفير dtbuf [4..7]. انسخ rkey ← K[0..15]. انسخ rseed ← K[16..23].

b. قم بتشفير dtbuf بطول 64 بتاً باستخدام المفتاح rkey بطول 128 بتاً في نمط ECB؛ وخزّن النتيجة في dtbuf.

G3. [التحضير لتوليد البايتات العشوائية]

ضع rcount ← 0 وk ← 23. سيتم تنفيذ الحلقة في الخطوات G4 إلى G7 24 مرة (k = 23...0)، أي مرة واحدة لكل بايت عشوائية تُنتج وتوضع في المتغير K. المتغير rcount هو عدد البايتات العشوائية غير المستخدمة في rbuf. سيَعِدّ هذا المتغير تنازلياً من 8 حتى 0 ثلاث مرات لتوليد 24 بايتاً.

G4. [البايتات متاحة؟]

إذا كان rcount = 0 اذهب إلى G5 وإلا فاذهب إلى G7. تقوم الخطوات G5 وG6 بتنفيذ خوارزمية X12.17 مرة واحدة لتوليد دفعة جديدة من ثمانية بايتات عشوائية.

G5. [توليد بايتات عشوائية جديدة]

a. rseed ← rseed ⊕ dtbuf.

b. rbuf ← Erkey [rseed] في نمط ECB.

G6. [توليد البذرة التالية]

a. rseed ← rbuf ⊕ dtbuf.

b. rseed ← Erkey [rseed] في نمط ECB.

c. rcount ← 8.

G7. [نقل البايتات واحد في كل مرة من rbuf إلى K]

a. rcount ← (rcount – 1).

b. ولّد بايت عشوائي b، ثم ضع b ⊕ rbuf [rcount] ← K[k].

G8. [هل انتهيت؟]

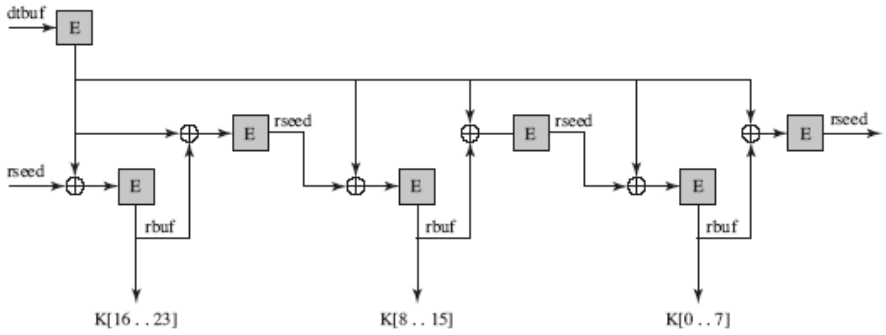
إذا كان k = 0 اذهب إلى G9 وإلا ضع k-1 ← k واذهب إلى G4.

G9. [الغسلة البعدية للبذرة، والعودة بالنتيجة]

a. قم بتوليد 24 بايت أخرى باستعمال الطريقة في الخطوات G4 - G7، باستثناء عملية الـ XOR مع البايت العشوائية في الخطوة G7. ضع النتيجة في المخزن المؤقت 'K.

b. قم بتشفير 'K بالمفتاح K[0...15] والمتجه الأوّلي IV K[16...23] في نمط CFB؛ خزّن النتيجة في randseed.bin.

c. ارجع بقيمة K.



الشكل 5-12: توليد مفتاح الجلسة والمتجه الأوّلي في PGP.
(الخطوات من G2 وحتى G8).

ينبغي ألا يكون ممكناً تحديد مفتاح الجلسة من الـ 24 بايتاً الجديدة التي يتم توليدها في الخطوة G9 (a). ومع ذلك، فللتأكد من أن الملف المخزّن randseed.bin لا يقدم أي معلومات عن أحدث مفتاح جلسة، يتم تشفير البايتات الـ 24 الجديدة وتخزين النتيجة لتكون البذرة الجديدة.

حريٌّ بهذه الخوارزمية المعقدة توفير أرقام شبه عشوائية قوية في مجال التشفير.

الفصل السادس

6 أمن بروتوكول الإنترنت

محتويات الفصل:

6-1 لمحة عامة عن أمن بروتوكول الإنترنت

6-2 البنية الأمنية لبروتوكول الإنترنت (IP)

6-3 ترويسة التوثيق

6-4 التغليف الأمني للحمولة

6-5 الجمع بين الارتباطات الأمنية

6-6 إدارة المفاتيح

6-7 توصيات للمطالعة

6-8 مصادر للمعلومات على الويب

6-9 مصطلحات رئيسة

6-10 أسئلة للمراجعة ومسائل

الملحق A-6: التشبيك البيني وبروتوكول الإنترنت

"إذا قام جاسوس بإفشاء سر قبل أوانه، فلابد من تنفيذ حكم الإعدام فيه وفيمن أُفشى إليه ذلك السر."

— من كتاب فن الحرب، لصن تزو.

النقاط الرئيسة

- أمن بروتوكول الإنترنت (IPSec) هو إمكانيات يمكن إضافتها إلى أيٍّ من نسخ بروتوكول الإنترنت الحالية (IPv4 أو IPv6) بواسطة ترويسات إضافية.
- يشمل أمن بروتوكول الإنترنت IPSec ثلاثة مجالات وظيفية: التوثيق، والسرية، وإدارة المفاتيح.
- يستخدم التوثيق كود HMAC للتحقّق من الرسالة. ولتوثيق رسالة، يمكن تطبيق عملية التوثيق على رزمة بروتوكول الإنترنت الأصلية كاملةً (نمط النفق) أوعلى كل الرزمة باستثناء ترويسة بروتوكول الإنترنت (نمط النقل).
- يتم توفير السرية عن طريق صيغة تشفير تُعرف باسم "التغليف الأمني للحمولة". يمكن استيعاب كلٍّ من نمطي النفق والنقل.
- يحدّد أمن بروتوكول الإنترنت (IPSec) عدداً من أساليب إدارة المفاتيح.

لقد طوَّر مجتمع الإنترنت آليات أمن خاصة بتطبيقات معينة في عدد من مجالات التطبيق، بما في ذلك البريد الإلكتروني (S/MIME, PGP)، نظم زبون/خادم (Kerberos)، نظم الوصول إلى شبكة الويب (طبقة المقابس الآمنة SSL)، وغيرها. ومع ذلك فلدى المُستخدِمين بعض المخاوف الأمنية فيما يتعلق بكل طبقات البروتوكول. فعلى سبيل المثال، تستطيع مؤسسة ما تشغيل شبكة TCP/IP خاصة وآمنة من خلال حظر روابط المواقع غير الموثوق بها، وتشفير الرزم المغادرة لحدود المؤسسة، وتوثيق (authenticating) الرزم التي تدخل حدود المؤسسة. وهكذا فمن خلال تنفيذ الأمن على مستوى بروتوكول الإنترنت (IP)، يمكن لمؤسسة أن

تضمن أمن شبكاتها ليس فقط للتطبيقات التي تمتلك آليات أمنية ولكن أيضا بالنسبة للكثير من التطبيقات التي لا تمتلك تلك الآليات.

يشمل الأمن على مستوى بروتوكول الإنترنت (IP) ثلاث وظائف أساسية: التوثيق (authentication)، والسرية (confidentiality)، وإدارة المفاتيح (key management). تضمن آلية التوثيق أن الرزمة المُستلمة قد أُرسِلت فعلاً من قِبَل الطرف المعرّف كمصدر لها في ترويسة الرزمة. بالإضافة لذلك، تضمن تلك الآلية أن الرزمة لم يتم تغييرها أثناء انتقالها. أما وظيفة السرية فتتيح لطرفي الاتصال تشفير رسائلهم المتبادلة لمنع تنصت أي طرف ثالث، بينما تختص وظيفة إدارة المفاتيح بعملية التبادل الآمن للمفاتيح.

نبدأ هذا الفصل بلمحة عامة عن أمن بروتوكول الإنترنت (IPSec) ومقدمة لبنية ذلك البروتوكول. بعد ذلك نلقي نظرةً تفصيليةً على مجالات كلٍّ من الوظائف الأساسية الثلاث. ويتضمن ملحق هذا الفصل مراجعةً لبروتوكولات الإنترنت.

1-6 لمحة عامة عن أمن بروتوكول الإنترنت

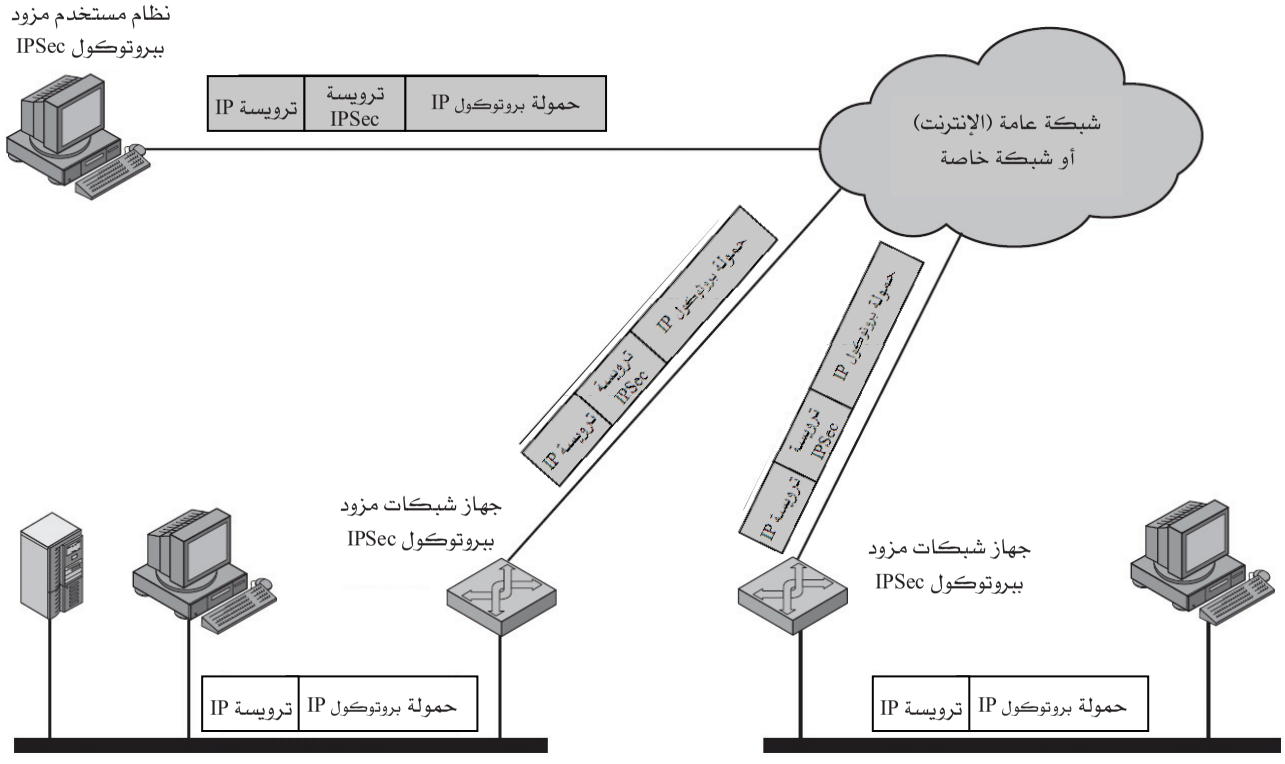
استجابةً لتلك القضايا، قام مجلس البنية المعمارية للإنترنت (IAB) بإدراج التوثيق والتشفير كسِمات أمن ضرورية في الجيل المقبل من بروتوكول الإنترنت، والذي صدَر باسم IPv6. ولحسن الحظ، فقد صُممت تلك الإمكانات الأمنية بحيث يمكن استخدامها مع كلٍّ من إصدار IPv4 الحالي وإصدار IPv6 المستقبلي. وهذا يعني أن منتجي البرمجيات يمكنهم البدء بتوفير تلك الميزات من الآن، وقد قام كثيرٌ منهم بالفعل بتضمين تلك الإمكانات الأمنية (IPSec) في منتجاتهم.

1-1-6 تطبيقات أمن بروتوكول الإنترنت (IPSec)

يوفر أمن بروتوكول الإنترنت (IPSec) إمكانية تأمين الاتصالات عبر الشبكات المحلية والشبكات الواسعة سواءً كانت عامة أو خاصة، وكذلك عبر الإنترنت. ومن أمثلة تلك الاستخدامات ما يأتي:

- تأمين الاتصال بمكتب فرعي عبر الإنترنت: يمكن لشركة إنشاء شبكة خاصة افتراضية ((virtual private network (VPN آمنة عبر الإنترنت أو عبر شبكة واسعة عامة (WAN). وبهذه الطريقة يمكن للشركات الاعتماد بشكلٍ كبير على شبكة الإنترنت ومن ثم تقليل احتياجاتها من الشبكات الخاصة، مما يوفر عليها كثيراً من التكاليف وأعباء إدارة الشبكة.
- الوصول الآمن عن بُعد عبر الإنترنت: يمكن لمستخدمٍ طريفٍ مزوّد ببروتوكولات أمن الإنترنت (IP security protocols) إجراء مكالمة محلية لموفر خدمة الإنترنت (ISP) والوصول من خلالها بشكلٍ آمن إلى شبكة الشركة، مما يقلل من تكاليف سفر عدد من الموظفين وانتقالهم إلى موقع الشركة يومياً.
- إنشاء توصيلات إكسترانت وإنترانت مشتركة مع الشركاء: يمكن أن يُستخدم بروتوكول IPSec لتأمين الاتصال مع المؤسسات الأخرى، بما يضمن الموثوقية والسرية ويوفر آلية لتبادل المفاتيح.
- تعزيز أمن التجارة الإلكترونية: رغم أن بعض تطبيقات الويب وبعض تطبيقات التجارة الإلكترونية مزودة ببروتوكولات أمنية مبيّتة كجزءٍ منها، فإن استخدام IPSec يعزز من هذا الأمن.

تكمن الميزة الرئيسة لبروتوكول IPSec – التي تمكنه من دعم تلك التشكيلة الكبيرة من التطبيقات المتنوعة – في أنه يمكنه تشفير و/أو توثيق كل حركة مرور للبيانات على مستوى بروتوكول الإنترنت (IP). ولذا، فإنه يمكن تأمين كافة التطبيقات الموزعة، بما في ذلك الوصول عن بُعد (remote logon)، وتطبيقات زبون/خادم، والبريد الإلكتروني، ونقل الملفات، والوصول إلى شبكة الويب، وهلم جرا.



الشكل 6-1: أحد سيناريوهات بروتوكول IP.

يوضّح الشكل 6-1 مثالاً لأحد السيناريوهات النمطية لاستخدام بروتوكول IPSec. تمتلك مؤسسة عدة شبكات محلية في مواقع متفرقة. يُستخدم بروتوكول الإنترنت IP غير الآمن على كلٍّ من تلك الشبكات المحلية. أمَّا حركة البيانات بعيداً عن مواقع المؤسسة فتستخدم بروتوكول IPSec عبر شبكة واسعة (WAN) عامة أو خاصة، حيث يجري تنفيذ تلك البروتوكولات في أجهزة الشبكات، كالموجّهات (routers) أو جدران الحماية النارية (firewalls)، والتي تربط كلاً من الشبكات المحلية بالعالم الخارجي. تقوم أجهزة بروتوكول IPSec تلك على الشبكة في العادة بتشفير (encrypt) وضغط (compress) كل البيانات التي تمر من الشبكة المحلية إلى الشبكة الواسعة العامة، وكذلك إزالة تشفير (decrypt) وفك ضغط (decompress) كل البيانات التي تمر من الشبكة الواسعة إلى الشبكة المحلية. ويلاحظ أن تلك العمليات تكون شفافة بالنسبة لمحطات العمل والخادمات على الشبكة المحلية. يمكن أيضاً تأمين نقلٍ آمن للبيانات لأفراد المُستخدِمين الذين يتصلون بالشبكة الواسعة (WAN) عن طريق وصلة مودم هاتفي. ينبغي أن تقوم محطات العمل التي يستعملها هؤلاء المُستخدِمين بتنفيذ بروتوكولات IPSec لتوفير الأمن لهم.

6-1-2 فوائد أمن بروتوكول الإنترنت

عدّد [MARK97] المزايا الآتية لبروتوكول IPSec:

- عند تنفيذ بروتوكول IPSec في جدار حماية (firewall) أو موجّه (router)، فإنه يوفر أمناً قوياً يمكن تطبيقه على كل حركة مرور البيانات التي تعبر حدود شبكة المؤسسة في كلا الاتجاهين. أما داخل نطاق المؤسسة أو مجموعة العمل فلن تعاني حركة المرور من الأعباء الإضافية الناجمة عن المعالجة الأمنية التي يتطلبها بروتوكول IPSec.
- عند تنفيذ بروتوكول IPSec في جدار حماية فإنه يقاوم محاولات التجاوز إذا تعيّن على كل حركة المرور الواردة من الخارج استخدام بروتوكول

الإنترنت (IP) وكان جدار الحماية هو المعبر الوحيد لدخول البيانات من الإنترنت إلى المؤسسة.

- نظراً لوجود بروتوكول IPSec أسفل طبقة النقل (TCP، UDP) فإنه يكون شفافاً بالنسبة للتطبيقات، لذا فلا حاجة لتغيير أي برمجيات على نظام لمستخدم ولا على نظام لخادم إذا ما كان IPSec مُنفَّذاً في الجدار الناري أو الموجّهات. وحتى في حالة تنفيذ IPSec على الأنظمة الطرفية، فإن برمجيات الطبقة العليا (بما في ذلك التطبيقات) لن تتأثر.
- يمكن أن يكون بروتوكول IPSec شفافاً للمستخدمين الطرفيين، ولذا فلا حاجة لتدريب المُستخدِمين على آليات الأمن، ولا لإصدار مفاتيح خاصة لكل مستخدم أو إلغاء تلك المفاتيح عندما يترك المُستخدِم العمل.
- يمكن أن يوفر بروتوكول IPSec الأمن لأفراد المُستخدِمين إذا لزم الأمر، وهذا أمر مفيد للعاملين الذين يبعدون عن موقع العمل وكذلك لإقامة شبكة فرعية افتراضية آمنة داخل المؤسسة للتطبيقات الحساسة.

6-1-3 تطبيقات التوجيه

بالإضافة إلى دعم المُستخدِمين الطرفيين وحماية أنظمة المؤسسة وشبكاتها، يمكن لبروتوكول IPSec أن يؤدي دوراً حيوياً في بنية التوجيه المُستخدَمة لتوصيل الشبكات مع بعضها. ويعدّد المرجع [HUIT98] الأمثلة الآتية لاستخدام بروتوكول IPSec في التأكد مما يأتي:

- أن الإعلان عن موجّه (أي موجّه جديد يعلن عن نفسه) يأتي من موجّه موثوق.
- أن الإعلان عن جار (أي موجّه يسعى إلى إنشاء أو الحفاظ على علاقة جوار مع موجّه في مجال توجيه آخر) يأتي من موجّه موثوق.
- أن رسالة إعادة التوجيه تأتي من الموجّه الذي تم إرسال الرزمة الأولية إليه.
- أن عملية تحديث جداول التوجيه (routing update) غير مزيفة.

بدون هذه التدابير الأمنية يمكن للخَصْم عرقلة الاتصالات أو تحويل اتجاه بعض حركة المرور. ينبغي أن تعمل بروتوكولات التوجيه – مثل بروتوكول OSPF – فوق الارتباطات الأمنية بين الموجّهات ويتم تعريفها بواسطة بروتوكول IPSec.

2-6 البنية الأمنية لبروتوكول الإنترنت (IP)

لقد أصبحت مواصفات بروتوكول IPSec معقدةً للغاية. وللتعرف على البنية العامة لهذا البروتوكول سنبدأ باستعراض الوثائق التي تُعرِّف البروتوكول، ثم نناقش خدمات IPSec ونتناول مفهوم ارتباطات الأمن (security association).

1-2-6 وثائق أمن بروتوكول الإنترنت

تتكون مواصفات أمن بروتوكول الإنترنت من عدد من الوثائق، أهمها ما صدر في نوفمبر عام 1998 وتشمل طلبات التعليقات (RFC) أرقام 2401، 2402، 2406، 2408:

- طلب التعليقات 2401: لمحة عامة عن البنية الأمنية لـ IPSec .
- طلب التعليقات 2402: وصف لتوثيق الرزمة كامتداد لـ IPv4 وIPv6.
- طلب التعليقات 2406: وصف لتشفير الرزمة كامتداد لـ IPv4 وIPv6.
- طلب التعليقات 2408: مواصفات الإمكانات الخاصة بإدارة المفاتيح.

يُعدُّ دعم هذه السمات إلزامياً لـ IPv6 واختيارياً لـ IPv4. في كلتا الحالتين، يتم تنفيذ السمات الأمنية كترويسات مُلحقة تتبع ترويسة بروتوكول الإنترنت (IP) الرئيسة. وتُعرف الترويسة المُلحقة للتوثيق بترويسة التوثيق والترويسة المُلحقة للتشفير بترويسة حمولة الأمن المغلّفة (Encapsulating Security Payload (ESP)).

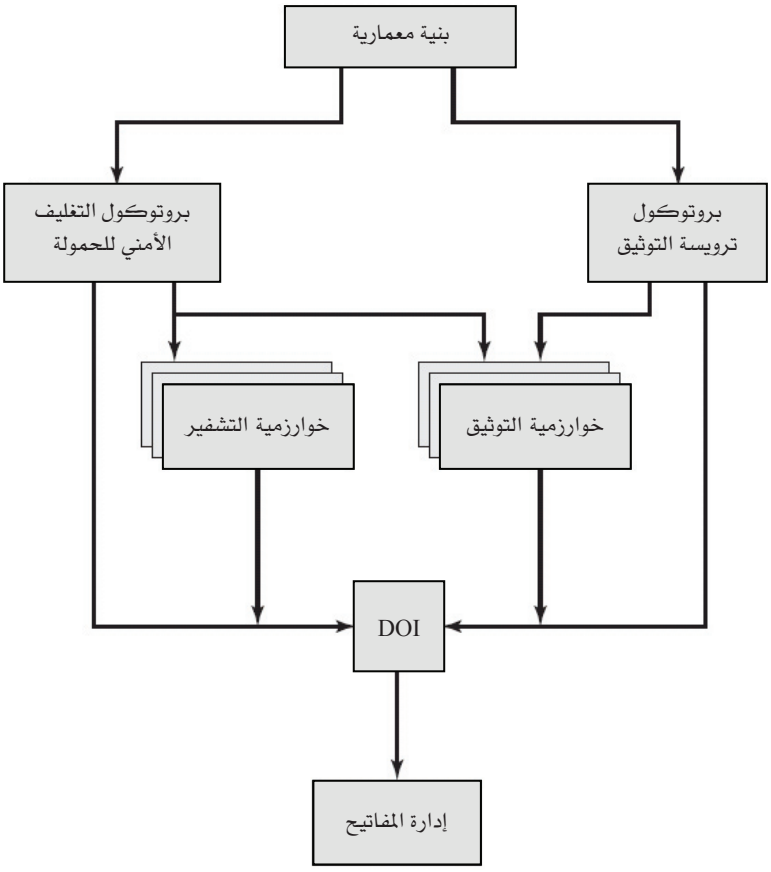
بالإضافة إلى الوثائق الأربع المذكورة، فقد تم نشر عدد من المسودات الإضافية بواسطة مجموعة عمل بروتوكول أمن الإنترنت (IP Security Protocol

(Working Group من قِبَل فريق مهام هندسة الإنترنت (IETF). وتقسم الوثائق إلى سبع مجموعات، كما هو مبين في الشكل 6-2 (طلب التعليقات 2401):

- البنية: تتناول المفاهيم العامة، والمتطلبات الأمنية، والتعريفات، وآليات تقنية IPSec.
- حمولة الأمن المغلّفة (ESP): تغطي صيغة الرزمة (format) والأمور العامة المتصلة باستخدام ESP لتشفير الرزمة وكذلك توثيقها (إن لزم الأمر).
- ترويسة التوثيق ((Authentication Header (AH): تغطي صيغة الرزمة والأمور العامة التي تتعلق باستخدام AH لتوثيق الرزمة.
- خوارزمية التشفير: مجموعة من الوثائق التي تصف كيفية استخدام خوارزميات تشفير مختلفة لـ ESP.
- خوارزمية التوثيق: مجموعة من الوثائق التي تصف كيفية استخدام خوارزميات التوثيق المختلفة لـ AH ولخيار التوثيق في ESP.
- إدارة المفاتيح: الوثائق التي تصف أنظمة إدارة المفاتيح.
- نطاق التفسير: يتضمن القيم المطلوبة للوثائق الأخرى حتى يمكن ربط الوثائق بعضها ببعض. ويشمل ذلك أسماء خوارزميات التشفير والتوثيق المصرح بها، وكذلك قيم مُعامِلات التشغيل كأعمار المفاتيح.

6-2-2 خدمات أمن بروتوكول الإنترنت (IPSec)

يقدم بروتوكول IPSec خدمات أمنية لطبقة بروتوكول الإنترنت (IP) من خلال تمكين النظام من اختيار البروتوكولات الأمنية المطلوبة، وتحديد خوارزميات الخدمات، وحفظ مفاتيح التشفير اللازمة لتفعيل الخدمات المطلوبة. ولتوفير الأمن يستخدم IPSec بروتوكولين: الأول عبارة عن بروتوكول توثيق يتم تحديده بواسطة ترويسة البروتوكول – أي ترويسة التوثيق (AH) – أما الثاني فبروتوكول مشترك للتشفير والتوثيق يتم تحديده من قِبَل صيغة (format) رزمة هذا البروتوكول؛ حمولة الأمن المغلفة (ESP). يقدم IPSec الخدمات الأمنية الآتية:



الشكل 6-2: عرض عام لوثائق بروتوكول IPSec.

الجدول 6-1 خدمات بروتوكول IPSEC.

	التغليف الأمني للحمولة (تشفير مع توثيق)	التغليف الأمني للحمولة (تشفير فقط)	ترويسة التوثيق
التحكم في الوصول	✓	✓	✓
السلامة اللاتوصيلية	✓		✓
توثيق منشأ البيانات	✓		✓
رفض الرزم المعاد إرسالها	✓	✓	✓
السرية	✓	✓	
سرية محدودة لتدفق حركة المرور	✓	✓	

- التحكم في الوصول (Access control).
- السلامة اللاتوصيلية (Connectionless integrity).
- توثيق منشأ البيانات (Data origin authentication).
- رفض الرزم المعاد إرسالها (Rejection of replayed packets) (شكل من أشكال سلامة التسلسل الجزئي).
- السرية (Confidentiality).
- سرية محدودة لتدفق حركة المرور (Limited traffic flow confidentiality).

يبيّن الجدول 6-1 الخدمات التي يوفرها بروتوكولا AH وESP. هناك حالتان لبروتوكول ESP؛ مع وبدون خيار التوثيق. و يُعدُّ كلٌّ من AH وESP وسيلة للتحكم في الوصول مبنية على أساس توزيع مفاتيح التشفير وإدارة تدفق حركة المرور بالنسبة لتلك البروتوكولات الأمنية.

6-2-3 الارتباطات الأمنية

يُعدُّ الارتباط الأمني (security association (SA)) مفهوماً أساسياً في آليات كلٍّ من التوثيق والسرية لبروتوكول الإنترنت (IP). ويُعَرَّف الارتباط بأنه علاقة في اتجاه واحد بين مرسل ومستقبِل توفر خدماتٍ أمنية لحركة المرور على هذه العلاقة. عند الحاجة لإقامة علاقة أقران (peer relationship) لتبادل المعلومات بطريقة أمنة في كِلا الاتجاهين، فإنه ينبغي استخدام ارتباطين أمنيين. وتُقَدم الخدمات الأمنية لارتباط أمنى للاستخدام إما من قِبَل AH أو ESP ، ولكن ليس كليهما.

ويحدّد الارتباط الأمني تماماً بمعرفة ثلاثة مُعامِلات:

- مؤشر المُعامِلات الأمنية (Security Parameters Index (SPI)): سلسلة من البتات تخصص لهذا الارتباط الأمني وتكون ذات دلالة محلية. ويتم نقل SPI في ترويسات AH وESP لتمكين نظام المستقبِل من اختيار الارتباط الأمني الذي تتم على أساسه معالجة الرزمة المستلَمة.

- عنوان IP للوجهة: هذا هو عنوان الوِجهة للارتباط الأمني، والذي يمكن أن يكون نظاماً طرفياً أو أحد أنظمة الشبكة كجدارٍ ناري مثلاً أو موجّه. وفي الوقت الحالي، يُسمح فقط بالعناوين أحادية الإرسال (unicast).
- معرِّف بروتوكول الأمن: يبيّن ما إذا كان هذا الارتباط ارتباطاً أمنياً خاصاً بـ AH أو ESP.

لذلك، ففي أي رزمة IP [1]، يُعَرَّف الارتباط الأمني بواسطة عنوان الوِجهة في ترويسة IPv4 أو IPv6 بالإضافة إلى مؤشر المُعامِلات الأمنية (SPI) في ترويسة الامتداد المُلحقة (AH أو ESP).

❖ مُعامِلات الارتباط الأمني:

في كل تنفيذ لبروتوكول IPSec، توجد قاعدة بيانات أساسية [2] للارتباطات الأمنية تحدّد المُعامِلات الخاصة بكل ارتباط أمني. وعادةً ما يُعَرَّف الارتباط الأمني بالمُعامِلات الآتية:

- عدّاد الرقم التسلسلي: بقيمة طولها 32 بتاً، ويستخدم في توليد حقل الرقم التسلسلي في ترويسة AH أو ESP، كما سيوضّح في الجزء 3-6 (لازم لجميع التنفيذات).
- فيض الرقم التسلسلي: مؤشر يدل على ما إذا كان فيض عداد الرقم التسلسلي ينبغي أن يؤدي إلى حدث يتطلب التدقيق وإلى منع انتقال المزيد من الرزم على هذا الارتباط الأمني أم لا (لازم لجميع التنفيذات).
- النافذة المضادة لإعادة الإرسال: تُستخدم لتحديد ما إذا كانت رزمة AH أو ESP الواصلة قد أعيد إرسالها، كما سيوضّح بالجزء 3-6 (لازمة لجميع التنفيذات).

---

[1] في هذا الفصل تشير رزمة IP إلى وحدة بيانات IPV4 أو رزمة بيانات IPV6.

[2] أساسية بمعنى أن الوظائف التي توفرها قاعدة بيانات الارتباط الأمني يجب أن تكون موجودة في أي تنفيذ لـ IPSec، ولكن الطريقة التي يتم بها توفير تلك الوظائف ترجع إلى المنفّذ.

- معلومات AH: وتشمل خوارزمية التوثيق،والمفاتيح، وأعمار المفاتيح، والمتغيرات ذات الصلة المستخدمة مع AH (لازمة لتنفيذات AH).
- معلومات (ESP): وتشمل خوارزمية التشفير والتوثيق، ومفاتيحها، وقيم التهيئة الخاصة بها، وأعمار المفاتيح، والمتغيرات ذات الصلة المستخدمة مع ESP (لازمة لتنفيذات ESP).
- عمر هذا الارتباط الأمني: وهو الفترة الزمنية أو عدد البايتات التي يجب بعدها إما استبدال الارتباط الأمني بآخر جديد (ومُعامِل أمني SPI جديد) أو إنهاؤه، بالإضافة إلى مؤشر للدلالة على أيٍّ من هذين الإجراءين ينبغي اتخاذه (لازم لجميع التنفيذات).
- نمط بروتوكول IPSec: ويشمل نمط النفق (Tunnel)، ونمط النقل (Transport)، ونمط الرمز العام (Wildcard) (لازم لجميع التنفيذات). وستتم مناقشة هذه الأنماط لاحقاً في هذا الجزء.
- MTU [3] للمسار: وهي أقصى وحدة إرسال مُلاحَظَة للمسار والمتغيرات التي طالت فترة تواجدها (لازمة لجميع التنفيذات).

تُقرَن آلية إدارة المفاتيح التي يتم استخدامها لتوزيع المفاتيح بآليات التوثيق والخصوصية عن طريق المُعامِل الأمني فقط. ومن ثَمَّ يتم تحديد التوثيق والخصوصية بشكلٍ مستقل عن أي آلية محدّدة لإدارة المفاتيح.

[3] MTU = Maximum Transmission Unit : الوحدة القصوى للإرسال، وهو أقصى حجم ممكن لرزمة بحيث يمكن إرسالها من دون تجزئة

❖ مُعامِلات اختيار الارتباط الأمني (SA Selectors):

يوفر بروتوكول IPSec للمستخدم قدرا كبيرا من المرونة من حيث طريقة تطبيق خدمات IPSec على حركة مرور بروتوكول الإنترنت (IP). وكما سنرى لاحقا، يمكننا الجمع بين عدة ارتباطات أمنية بأكثر من طريقة للحصول على التشكيلة المطلوبة للمُستخدِم. وعلاوةً على ذلك، يتميز IPSec بقدرة عالية على الفرز والتمييز بين حركة المرور المحميّة بـ IPSec والأخرى المسموح لها بتجاوز قيود IPSec، وفي الحالة الأولى الربط بين حركة مرور IP والارتباطات الأمنية الخاصة بها.

يتم الربط بين حركة مرور IP وارتباطات أمنية معينة (أوبدون أي ارتباطات أمنية في حالة حركة المرور المسموح لها بتجاوز IPSec) بواسطة قاعدة بيانات السياسة الأمنية ((Security Policy Database (SPD). في أبسط أشكالها، تحتوي قاعدة بيانات السياسة الأمنية على مُدخَلات يُعَرِّفُ كلٌّ منها مجموعةً فرعيةً من حركة مرور IP ويشير إلى الارتباط الأمني الخاص بها. أما في بيئات أكثر تعقيداً، فقد يوجد عدة مُدخَلات تربط بين نفس المجموعة الفرعية من حركة مرور IP وعدة ارتباطات أمنية أو بين ارتباط أمني واحد وعدة مجموعات فرعية من حركة مرور IP. ويمكن للقارئ الاطلاع على وثائق IPSec ذات الصلة للحصول على تفاصيل أكثر.

ويُعرَّف كل مُدخل لقاعدة البيانات SPD بمجموعة من قيم حقول بروتوكول الإنترنت (IP) وبروتوكولات الطبقة العليا، تسمى مُعامِلات الاختيار (selectors). وفي الواقع، تُستخدَم تلك المُعامِلات لتصفية حركة المرور الصادرة لربطها بارتباط أمني معين. وتعالَج كل رزمة IP صادرة كما يأتي:

1. قارن قيم حقول مُعامِلات الاختيار في الرزمة بقاعدة بيانات السياسة الأمنية (SPD) لإيجاد مُدخل مطابق، والذي قد يشير إلى صفر أو عدد أكبر من الارتباطات أمنية.
2. حدّد الارتباطات الأمنية لتلك الرزمة، إن وجدت، بالإضافة إلى مؤشر المُعامِلات الأمنية (SPI) لها.
3. قم بمعالجة IPSec المطلوبة (أي معالجة AH أوESP).

ويتحدّد مُدخل قاعدة بيانات السياسة الأمنية بمُعامِلات الاختيار الآتية:

- عناوين IP للوِجهة: والتي قد تكون عنواناً واحداً، أو قائمة تضم عناوين محدّدة، أو مدى متصلاً من العناوين، أوعنوان قناع (mask) برمز عام (wildcard). والصنفان الأخيران مطلوبان لدعم عددٍ من أنظمة الوِجهة لها نفس الارتباط الأمني (كأن تكون مثلاً خلف نفس جدار الحماية الناري).
- عناوين IP للمصدر: والتي قد تكون عنواناً واحداً، أو قائمة تضم عناوين محدّدة، أو مدى متصلاً من العناوين، أوعنوان قناع برمز عام (wildcard). الصنفان الأخيران مطلوبان لدعم عددٍ من أنظمة المصدر لها نفس الارتباط الأمني (كأن تكون مثلاً خلف نفس جدار الحماية الناري).

- هوية المستخدم (UserID): مُعَرِّف للمستخدم من نظام التشغيل. ليس هذا حقلاً في بروتوكول الإنترنت IP ولا في ترويسات الطبقات العليا ولكنه يكون متاحاً إذا ما كان IPSec يعمل على نفس نظام تشغيل المُستخدِم.
- مستوى حساسية البيانات: يُستخدم للأنظمة التي تقوم بتأمين تدفق المعلومات (مثلاً سرِّي (secret) أو غير سرِّي (unclassified)).
- بروتوكول طبقة النقل: يتم الحصول عليه من حقل "الترويسة التالية" (Next Header) لبروتوكول IPv4 أو IPv6. وقد يكون ذلك رقم بروتوكول واحد، أو قائمة بأرقام بروتوكولات محدّدة، أو مدى متصل من أرقام البروتوكولات.
- منافذ المصدر والوِجهة: وقد تكون قيمة منفذ TCP أو UDP واحد، أو قائمة بتلك المنافذ، أومنفذ بالرمز العام (wildcard).

4-2-6 نمط النقل ونمط النفق

يدعم كلٌّ من AH وESP نمطين للاستخدام هما نمط النقل ونمط النفق. ويمكن فهم هذين النمطين بشكلٍ أفضل في سياق وصف AH وESP، كما سيتم في الجزأين 6-3 و6-4 على التوالي. أما هنا فنكتفي بلمحة موجزة.

❖ نمط النقل:

يوفر نمط النقل الحماية لبروتوكولات الطبقة العليا في المقام الأول، بمعنى أن حماية نمط النقل تمتد إلى الحمولة لرزمة بروتوكول الإنترنت (IP). ومن الأمثلة على ذلك قطعة TCP أو UDP أو رزمة ICMP، وجميعها تعمل مباشرة فوق بروتوكول الإنترنت في رصة بروتوكول المضيف. وعادةً ما يُستخدم نمط النقل في الاتصالات من طرف إلى طرف بين مضيفين (على سبيل المثال، عميل وخادم، أو محطتي عمل). وعندما يقوم مضيف بتشغيل AH أو ESP على IPv4، تكون الحمولة هي البيانات التي عادةً ما تتبع ترويسة بروتوكول الإنترنت (IP). أما في حالة IPv6، فتكون الحمولة هي البيانات التي تتبع عادةً كلاً من ترويسة بروتوكول الإنترنت (IP) وأيّاً من ترويسات امتدادات IPv6 المُستخدَمة، مع احتمال استثناء ترويسة خيارات الوِجهة التي يمكن أن تُضَمَّن في الحماية.

يقوم ESP في نمط النقل بتشفير الحمولة لبروتوكول الإنترنت (IP) وتوثيقه اختياريا ولا يتم ذلك مع ترويسة بروتوكول الإنترنت (IP). أما AH فيقوم في نمط النقل بتوثيق الحمولة لبروتوكول الإنترنت (IP) وأجزاء مختارة من ترويسة بروتوكول الإنترنت (IP).

❖ نمط النفق:

يوفر نمط النفق الحماية لرزمة بروتوكول الإنترنت (IP) بكاملها. ولتحقيق ذلك، تُضاف حقول AH أو ESP إلى رزمة بروتوكول الإنترنت، ثم تُعامَل الرزمة بأسرها بالإضافة إلى الحقول الأمنية على أنها الحمولة لرزمة IP "خارجية" جديدة بترويسة IP خارجية جديدة. وتسير الرزمة الأصلية (أو الداخلية) بأكملها عبر "نفق"

من نقطة على شبكة بروتوكول الإنترنت (IP) إلى النقطة التي تليها، ولا يمكن لأيٍّ من الموجّهات على طول المسار فحص ترويسة بروتوكول الإنترنت الداخلية. ولأن الرزمة الأصلية مغلّفة، فإن الرزمة الأحدث والأكبر قد يكون لها عنواني مصدر ووِجهة مختلفين تماما مما يرفع من مستوى الأمن. ويُستخدَم نمط النفق عندما يكون أحد طرفي الارتباط الأمني أو كلاهما بوابة أمنية تُطبّق IPSec كجدار حماية ناري أو موجّه.

وفي نمط النفق، يستطيع عددٌ من المضيفين على شبكات خلف جدران نارية الاتصال الآمن مع بعضهم بعضاً دون تنفيذ IPSec. وتُنقل الرزم غير المحمية الصادرة عن المضيفين في مثل هذا الاتصال بنمط النفق عبر نفق خلال الشبكات الخارجية باستخدام الارتباطات الأمنية المثبَّتة عن طريق برمجيات IPSec في الجدران النارية أو الموجّهات الآمنة على حدود الشبكة المحلية.

فيما يأتي مثالٌ على كيفية عمل نمط النفق ببروتوكول IPSec. يُرسِل المضيف A رزمة IP على الشبكة بعنوان وِجهة للمضيف B على شبكة أخرى. تُوجَّه هذه الرزمة من المضيف الأصلي (المُرسِل) إلى جدار الحماية أو إلى موجّه آمن على حدود شبكة المضيف A، يقوم جدار الحماية بتصفية جميع الرزم الصادرة لتحديد مدى الحاجة لمعالجة IPSec. إذا كانت تلك الرزمة من المضيف A إلى المضيف B تتطلب IPSec، يقوم جدار الحماية بمعالجتها تبعاً لـ IPSec وتغليفها بترويسة IP خارجية. ويكون عنوان IP للمصدر لتلك الرزمة الخارجية هو عنوان جدار الحماية ذاك، أما عنوان الوِجهة فيمكن أن يكون عنوان جدار الحماية الذي يشكِّل حدود الشبكة المحلية للمضيف B. وتُوجَّه تلك الرزمة إلى جدار حماية المضيف B. تقوم موجّهات المسار بتفحُّص ترويسة IP الخارجية فقط. وعند جدار حماية المضيف B، تُنزع ترويسة بروتوكول الإنترنت الخارجية ويتم تسليم الرزمة الداخلية للمضيف B.

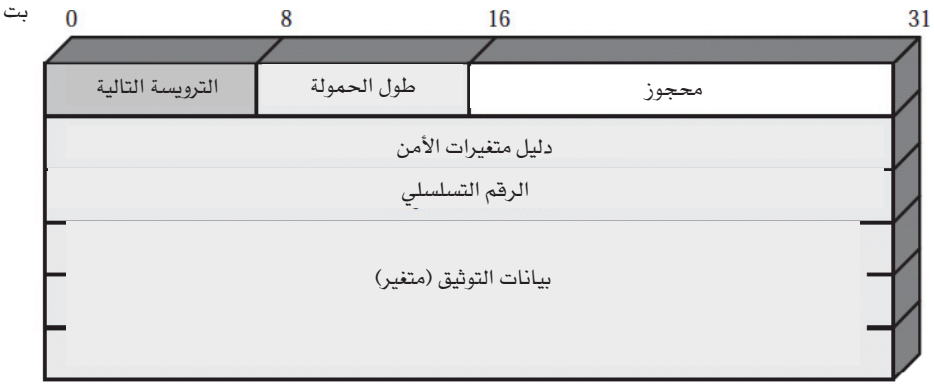
يقوم ESP في نمط النفق بتشفير رزمة بروتوكول الإنترنت الداخلية كاملةً وتوثيقها اختياريا، بما في ذلك ترويسة IP الداخلية. يقوم AH في نمط النقل بتوثيق رزمة بروتوكول الإنترنت الداخلية كاملةً وأجزاء مختارة من ترويسة IP الخارجية.
ويلخص الجدول 6-2 طريقة عمل نمطي النقل والنفق.

3-6 ترويسة التوثيق

توفّر ترويسة التوثيق دعماً لسلامة بيانات رزم بروتوكول الإنترنت (IP) وتوثيقها. تضمن ميزة سلامة البيانات استحالة أن تمر أي تغيّرات في محتوى الرزمة خلال مسارها بدون اكتشاف. أما عملية التوثيق فتُمكّن أيّ نظام طرفي أو جهاز على الشبكة من توثيق المستخدم أو التطبيق، ثم تقوم بتصفية حركة المرور بناءً على ذلك، كما تمنع هجمات تزييف العنوان (spoofing) المنتشرة حالياً على الإنترنت. تحمي ترويسة التوثيق أيضاً ضد هجوم إعادة الإرسال (replay attack) الذي سيتم وصفه لاحقاً في هذا الجزء.

الجدول 6-2 وظائف نمطي النفق والنقل.

	ارتباط أمني بنمط النقل	ارتباط أمني بنمط النفق
ترويسة التوثيق	توثيق حمولة بروتوكول IP وأجزاء محدّدة من ترويسة التوثيق، وترويسات IPv6 ملحقة	توثيق ترويسة بروتوكول IP الداخلية كاملةً (الترويسة الداخلية + حمولة بروتوكول IP) + أجزاء محدّدة من ترويسة بروتوكول IP الخارجية وترويسات IPv6 الملحقة الخارجية
التغليف الآمني للحمولة	تشفير حمولة بروتوكول IP بالإضافة إلى أيٍّ من ترويسات IPv6 الملحقة والتي تلي ترويسة التغليف الآمن للحمولة	تشفير رزمة بروتوكول IP الداخلية كاملةً
التغليف الأمني للحمولة مع التوثيق	تشفير حمولة بروتوكول IP بالإضافة إلى أيٍّ من ترويسات IPv6 الملحقة والتي تلي ترويسة التغليف الآمن للحمولة. توثيق حمولة بروتوكول IP ولكن ليس ترويسة بروتوكول IP.	تشفير حمولة بروتوكول IP الداخلية كاملةً. توثيق حمولة بروتوكول IP الداخلية كاملةً



الشكل 6-3: ترويسة التوثيق لبروتوكول IPSec.

يعتمد التوثيق على استعمال كود التحقق من الرسالة (ماك) (Message Authentication Code (MAC))، وكما هو موضّح في الفصل الثالث، يجب أن يشترك كلا الطرفين في المفتاح السري.

تتكون ترويسة التوثيق من الحقول الآتية (الشكل 6-3):

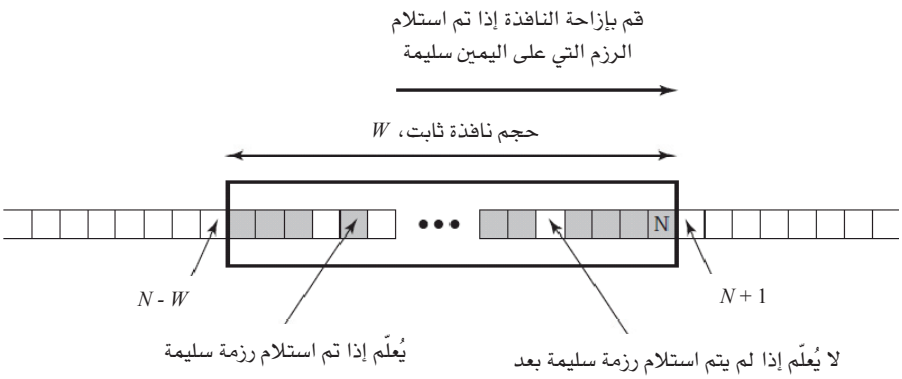
- الترويسة التالية (Next Header) (8 بتات): وتحدّد نوع الترويسة اللاحقة مباشرةً لهذه الترويسة.
- طول الحمولة (Payload Length) (8 بتات): وهو طول ترويسة التوثيق (مُقاساً بعدد الكلمات بطول 32 بتاً) ناقص 2. وعلى سبيل المثال، الطول المفترَض لحقل بيانات التوثيق هو 96 بتاً، أي ثلاث كلمات كلٌّ منها 32 بتاً). وبإضافة ثلاث كلمات هي طول الترويسة الثابتة، يكون طول الترويسة الكلي هو ستّ كلمات، ففي هذه الحالة يحتوي حقل طول الحمولة على القيمة 4 (أي 6 - 2 = 4).
- محجوز (Reserved) (16 بتاً): للاستخدام في المستقبل.
- مؤشر المُعامِلات الأمنية (Security Parameters Index): (32 بتاً): ويحدّد الارتباط الأمني.

- الرقم التسلسلي (Sequence Number) (32 بتاً): قيمة متزايدة لعدّاد، سيُناقَش لاحقاً.
- بيانات التوثيق (Authentication Data) (بطول متغيِّر): حقل متغير الطول (يجب أن يكون عدداً صحيحاً من كلمات 32- بت) ويتضمّن قيمة التحقق من السلامة (Integrity Check Value (ICV))، أو كود التحقق من الرسالة (ماك)، لهذه الرزمة، سيُناقَش لاحقاً.

6-3-1 الخدمة المضادة لإعادة الإرسال (Anti-Replay Service)

في هجوم إعادة الإرسال يحصل المهاجم على نسخة من الرزمة الموثّقة ثم يرسلها لاحقاً إلى الوِجهة المقصودة. وقد يؤدي استلام نسخ مكرّرة وموثقة من رزم بروتوكول الإنترنت (IP) إلى إرباك الخدمة بشكلٍ أو بآخر، أو إلى عواقب وخيمةٍ أخرى. وقد صُمِّم حقل الرقم التسلسلي لإحباط مثل تلك الهجمات. ولفهم ذلك سنناقش أولاً عملية توليد الرقم التسلسلي من قِبَل المُرسِل، ثم ننظر في الكيفية التي تتم بها معالجة هذا الرقم عند المُستلِم.

عندما يتم تأسيس ارتباط أمني جديد، يضع المُرسِل القيمة (0) في عداد الرقم التسلسلي. وفي كل مرة يتم فيها إرسال رزمة على هذا الارتباط الأمني، يقوم المُرسِل بزيادة قيمة العداد بواحد ويسجّل تلك القيمة في حقل الرقم التسلسلي. ومن ثَمَّ، فإن أول قيمة تُستخدم تكون 1. إذا كانت الخدمة المضادة لإعادة الإرسال مفعَّلة (وهو المُفترض أساساً) فينبغي ألا يسمح المُرسِل للرقم التسلسلي بالعودة من أقصى قيمة للعد (2^{32}-1) إلى نقطة الصفر حتى لا يسمح بوجود عدة رزم شرعية لها نفس الرقم التسلسلي. لذا فعند وصول العداد إلى القيمة (2^{32}-1) يجب على المُرسِل إنهاء ذلك الارتباط الأمني والتفاوض على ارتباط أمني جديد بمفتاح جديد.



الشـــكل 6-4: الآلية المضادة لإعادة الإرسال.

ونظراً لأن بروتوكول الإنترنت هوخدمة لاتوصيلية وغير موثوقة، فإن البروتوكول لا يضمن أن الرزم سيتم تسليمها بالترتيب ولا يضمن حتى أن كافة الرزم سيتم تسليمها. ولذلك فإن وثيقة IPSec الخاصة بالتوثيق تفرض على المتلقي أن يُنشئ نافذةً بحجم W، (بقيمة مُفترضة $W = 64$). ويُمثِّل أعلى رقم تسلسلي، N، ورد حتى الآن من الرزم الصالحة الحافة اليمنى للنافذة. يتم وضع علامة بالشريحة (fragment) المناظـرة في تلك النافذة (الشـكل 6-4) لأي رزمـة برقـم تسلسلي في النطـاق من $N - W + 1$ إلى N وتم استلامها بشكلٍ صحيح (أي موثّقة بشكلٍ مناسب). وعند استلام رزمة يتم معالجتها لدى المُستلم على النحو الآتي:

1. إذا وقعت الرزمة المستلَمة ضمن النافذة وكانت جديدة، يتم التحقق من كود ماك، فإذا كانت الرزمة موثّقة توضع علامة بالشريحة المناظرة في النافذة.
2. إذا كانت الرزمة المستلَمة على يمين النافذة وكانت جديدة، يتم التحقق من كود ماك، فإذا كانت الرزمة موثّقة يتم تقديم النافذة إلى اليمين فيكون الرقم التسلسلي هذا هو الحافة اليمنى للنافذة كما يتم وضع علامة بالشريحة المناظرة في النافذة.

3. إذا كانت الرزمة المستلَمة على يسار النافذة، أو في حالة فشل توثيقها، يتم تجاهل تلك الرزمة وتدوين هذا الحدث للمراجعة.

6-3-2 قيمة التحقق من السلامة (Integrity Check Value (ICV))

يحمل حقل بيانات التوثيق قيمة تُعرف بقيمة التحقق من السلامة (ICV). وهذه القيمة هي رمز توثيق الرسالة أو نسخة مبتورة (truncated) من الكود الناتج من خوارزمية ماك. وتنص المواصفات الحالية على ضرورة الالتزام بدعم ما يأتي:

- HMAC-MD5-96
- HMAC-SHA-1-96

وكلاهما يستخدم خوارزمية HMAC، الأولى مع خوارزمية خلاصة الرسائل MD5، والثانية مع كود التحوير الآمن SHA-1 (سبق مناقشة كلٍّ من تلك الخوارزميات في الفصل الثالث). في كلتا الحالتين يتم حساب القيمة الكاملة لـ HMAC ولكن بعد ذلك يتم بترها واستخدام الـ 96 بتاً الأولى فقط، والتي هي قيمة الطول المُفترَض لحقل بيانات التوثيق.

ويتم حساب كود التحقق من الرسالة (الماك) على:

- حقول ترويسة IP التي لا تتغير أثناء الانتقال (ثابتة) أو تلك التي يمكن التنبؤ بقيمتها عند وصولها إلى نقطة النهاية لهذا الارتباط الأمني لترويسة التوثيق. أما الحقول التي قد تتغير أثناء الانتقال، أو التي لا يمكن التنبؤ بقيمتها عند وصولها فيتم تعيين 0 كقيمة مبدئية لها بكلٍّ من المصدر والوِجهة لأغراض حسابية.
- ترويسة التوثيق AH فيما عدا حقل بيانات التوثيق. يتم تعيين 0 كقيمة مبدئية لحقل بيانات التوثيق بكلٍّ من المصدر والوِجهة لأغراض حسابية.
- بيانات بروتوكول المستوى الأعلى بأكملها، والتي يُفترَض أن تكون غير قابلة للتغيير أثناء الانتقال (على سبيل المثال، قطعة TCP أو رزمة IP داخلية في نمط النفق).

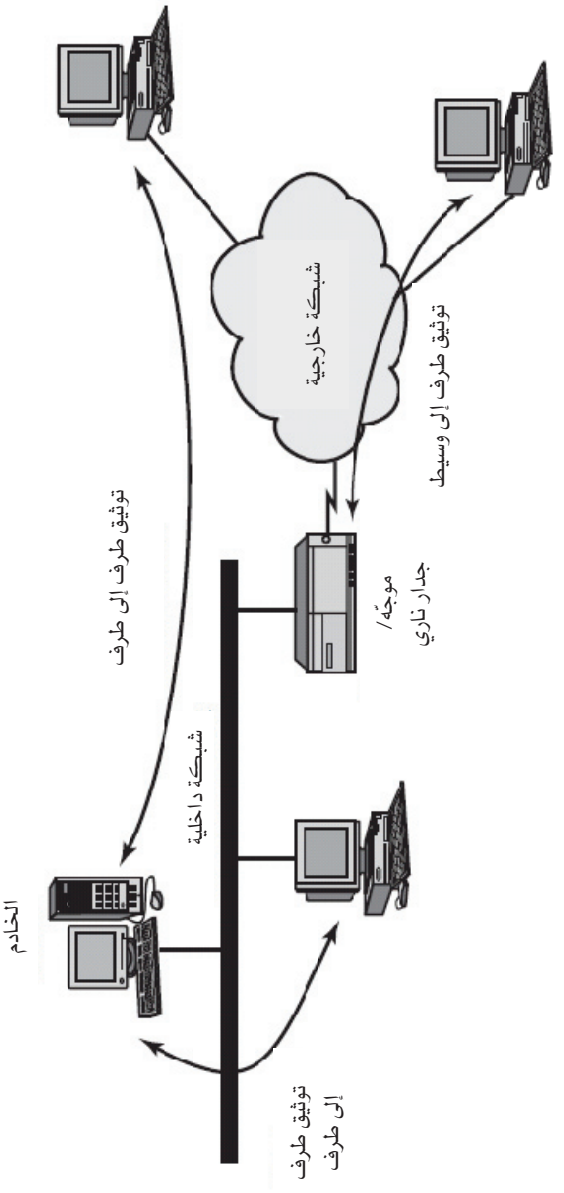
وكأمثلة على الحقول الثابتة في بروتوكول IPv4، هناك طول ترويسة الإنترنت وعنوان المصدر. ويعد عنوان الوِجهة (سواءً كان توجيه المصدر صارماً أو فضفاضاً) مثالاً على حقل قابل للتغيُّر ولكن يمكن التنبؤ به. أما أمثلة الحقول القابلة للتغيير ويتم تصفيرها قبل حساب ICV فتضم العُمر الافتراضي وحقول المجموع التدقيقي للترويسة. لاحظ أن كلاً من حقل عنوان المصدر وحقل عنوان الوِجهة محميّان، مما يمنع تزييف العناوين أو انتحالها.

بالنسبة لبروتوكول IPv6، فمن الأمثلة في الترويسة الرئيسة: رقم الإصدار (حقل ثابت)، وعنوان الوِجهة (حقل قابل للتغيُّر ولكن يمكن التنبؤ به)، ووسمة التدفق (قابل للتغير، ويُصفَّر لأغراض حسابية).

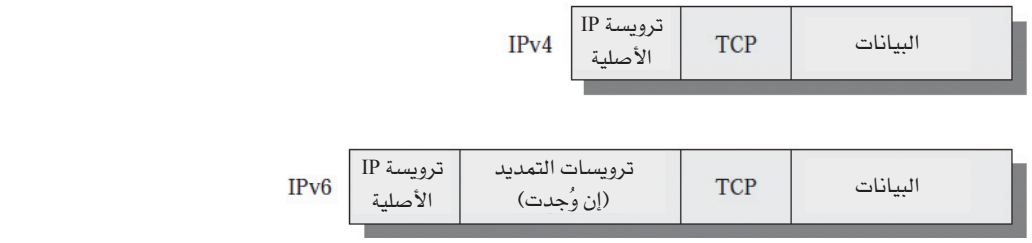
6-3-3 نمطا النقل والنفق

يوضّح الشكل 6-5 أسلوبين يُمكن من خلالهما استخدام خدمة توثيق IPSec. في الأسلوب الأول، يتم توفير خدمة التوثيق مباشرةً بين الخادم ومحطات عمل الزبون؛ ويمكن أن تكون محطات العمل تلك على نفس شبكة الخادم أو على شبكة خارجية. وطالما أن محطة العمل والخادم يشتركان في مفتاحٍ سريٍ محمي فإن عملية التوثيق تكون آمنة، يستخدم هذا الأسلوب الارتباط الأمني في نمط النقل. في الأسلوب الآخر، تُوثّق محطة العمل البعيدة نفسها للجدار الناري للمؤسسة، إما للنفاذ إلى الشبكة الداخلية بأكملها أو لأن الخادم المطلوب لا يدعم ميزة التوثيق، يستخدم هذا الأسلوب الارتباط الأمني في نمط النفق.

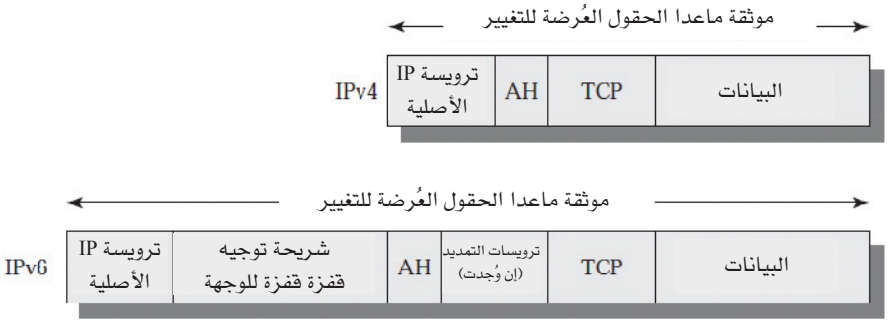
في هذا الجزء الفرعي سنتناول مجال التوثيق الذي توفره AH وموقع ترويسة التوثيق في كلا الأسلوبين، علماً بأن الاعتبارات مختلفةٌ بعض الشيء بالنسبة لـ IPv4 وIPv6. ويبيّن الشكل 6-6 (a) رزم IPv4 وIPv6 النمطية. في هذه الحالة، يكون حمل IP الآجر هو قطعة TCP، كما يمكن أيضا أن تكون وحدة بيانات لأي بروتوكول آخر يستخدم بروتوكول IP، كبروتوكول ICMP أو UDP.



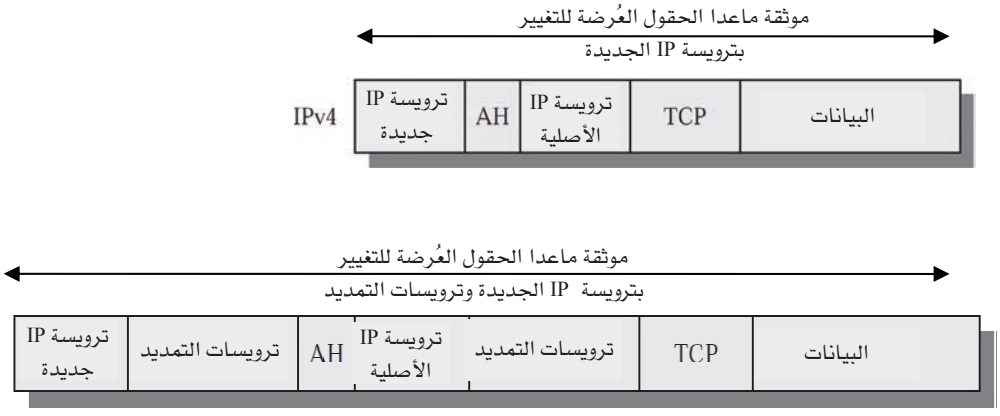
الشكل 6-5: التوثيق من طرف إلى طرف في مقابل التوثيق من طرف إلى وسيط.



(a) قبل إدراج ترويسة التوثيق



(b) نمط النقل



(c) نمط النفق

الشكل 6-6: مجالات التوثيق بترويسة AH.

بالنسبة لترويسة التوثيق بنمط النقل باستخدام IPv4، يتم إدراج ترويسة التوثيق بعد ترويسة بروتوكول الإنترنت الأصلية وقبل حمل IP الآجر (مثلاً قطعة TCP)، كما هو مُبيّن في الجزء العلوي من الشكل 6-6 (b). يغطي التوثيق الرزمة بأكملها، باستثناء الحقول القابلة للتغيير في ترويسة IPv4 والتي يتم تصفيرها لحساب كود التحقق من الرسالة (ماك).

وفي سياق IPv6، يُنظر إلى ترويسة التوثيق (AH) على أنها حمولة من طرفٍ إلى طرف، بمعنى أنها لا تُفحص أو تُعالج من قِبَل الموجّهات الوسيطة. وعليه تظهر ترويسة التوثيق بعد ترويسة الأساس لـ IPv6 وترويسات القفزة - بقفزة، والتوجيه، والترويسات الجزئية الملحقة. ويمكن أن تظهر ترويسة خيارات الوِجهة الملحقة قبل أو بعد ترويسة AH تبعاً للدلالات المرغوبة. ومرةً أخرى يشمل التوثيق الرزمة بأكملها، باستثناء القابلة للتغيُّر والتي يتم تصفيرها لحساب كود التحقق من الرسالة (ماك).

بالنسبة لترويسة التوثيق بنمط النفق، يتم توثيق رزمة بروتوكول IP الأصلية بأكملها، ويتم إدراج ترويسة التوثيق بين ترويسة IP الأصلية وترويسة IP خارجية جديدة (الشكل 6-6 (c)). وتحمل ترويسة IP الداخلية عناوين المصدر الأصلي الأساسي والوِجهة النهائية، بينما قد تتضمّن ترويسة IP الخارجية عناوين IP مختلفة (على سبيل المثال، عناوين جدران نارية أو بوابات أمنية أخرى).

في نمط النفق يتم حماية رزمة IP الداخلية بأكملها، بما في ذلك ترويسة IP الداخلية كلها، بواسطة ترويسة التوثيق. ويتم حماية ترويسة IP الخارجية (وفي حالة IPv6، الترويسات الملحقة الخارجية) باستثناء الحقول القابلة للتغيير، والتي لا يمكن التنبؤ بها.

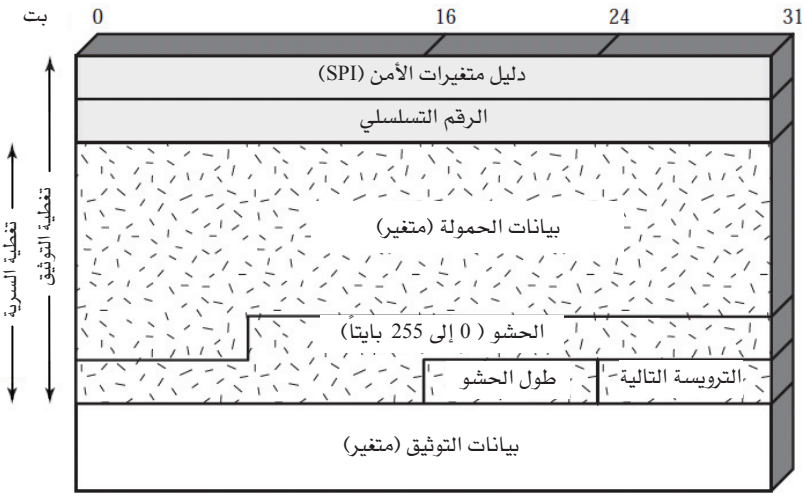
4-6 التغليف الأمني للحمولة (Encapsulating Security Payload (ESP))

يوفّر التغليف الأمني للحمولة (ESP) خدماتٍ للسرية، بما في ذلك سرية محتويات الرسالة وسرية محدودة لتدفق حركة المرور. كما يُمكن أيضاً للتغليف الأمني للحمولة توفير خدمة التوثيق كميزة اختيارية.

6-4-1 صيغة التغليف الأمني للحمولة

يوضّح الشكل 6-7 صيغة رزمة التغليف الأمني للحمولة (ESP)، والتي تتضمّن الحقول الآتية:

- مؤشر مُعامِلات الأمن (Security Parameters Index) (32 بتاً): يُعرِّف ارتباط أمن.
- الرقم التسلسلي (Sequence Number) (32 بتاً): قيمة متزايدة لعدّاد، ويستغل لاكتشاف إعادة الإرسال كما نوقش في ترويسة التوثيق.
- بيانات الحمولة (Payload Data) (بطول متغيِّر): وهي قطعة بمستوى النقل (في نمط النقل) أو رزمة IP (في نمط النفق) محميّة بالتشفير.
- الحشوة (Padding) (من 0 إلى 255 بايت): سنتناول الغرض من هذا الحقل لاحقاً.
- طول الحشوة (Pad Length) (8 بتات): ويُمثل عدد بايتات الحشوة.
- الترويسة التالية (Next Header) (8 بتات): وتحدّد نوع البيانات الواردة في حقل بيانات الحمولة عن طريق تحديد الترويسة الأولى في ذلك الحِمل (على سبيل المثال، ترويسة مُلحقة في IPv6 أو بروتوكول طبقة أعلى كـ TCP).
- بيانات التوثيق (Authentication Data) (بطول متغير): حقل بطول متغيّر (يجب أن يكون عدداً صحيحاً من كلمات 32-بت) يتضمّن قيمة تحقق السلامة (ICV) محسوبةً على رزمة التغليف الأمني للحمولة (ESP) ناقصةً حقل بيانات التوثيق.



الشكل 6-7: صيغة التغليف الأمني للحمولةِ في IPSec.

6-4-2 خوارزميات التشفير والتوثيق

تقوم خدمة التغليف الأمني للحمولةِ بتشفير الحقول الآتية: بيانات الحمل الآجر، والحشوة، وطول الحشوة، والترويسة التالية. إذا كانت الخوارزمية المستخدمة لتشفير الحمل الآجر تتطلب بيانات تزامن للتشفير، كمتجه التهيئة (IV)، فإنه يمكن إدراج تلك البيانات صراحةً في بداية حقل بيانات الحمولة. وإذا أُدرجت، فإن متجه التهيئة عادةً ما يكون غير مُشفَّر، رغم أنه كثيراً ما يُذكر على أنه جزء من النص المُشفَّر.

تتطلّب المواصفات الحالية أن تلتزم التنفيذات الممتثلة بدعم معيار تشفير البيانات (DES) في نمط تسلسل كتلة الشفرة (CBC) (نوقش في الفصل الثاني). وهناك عدد من الخوارزميات الأخرى التي تم تخصيص محدّدات لها في وثيقة نطاق التفسير (Domain of Interpretation (DOI)) ولذا يمكن استخدامها للتشفير بسهولة، بما في ذلك:

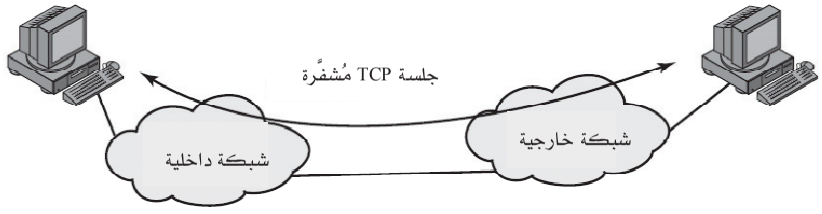
- معيار DES ثلاثي المفاتيح.
- RC5.
- IDEA.
- معيار IDEA ثلاثي المفاتيح.
- CAST.
- Blowfish.

وكما هوالحال مع ترويسة التوثيق، يدعم التغليف الأمني للحمولة استخدام كود التحقّق من الرسالة (ماك) بطول مُفترض يبلغ 96 بتاً. وأيضا كما هوالحال مع ترويسة التوثيق، تنص المواصفات الحالية على أن التنفيذات الممتثلة لابد أن تدعم HMAC-MD5-96 وHMAC-SHA-1-96.

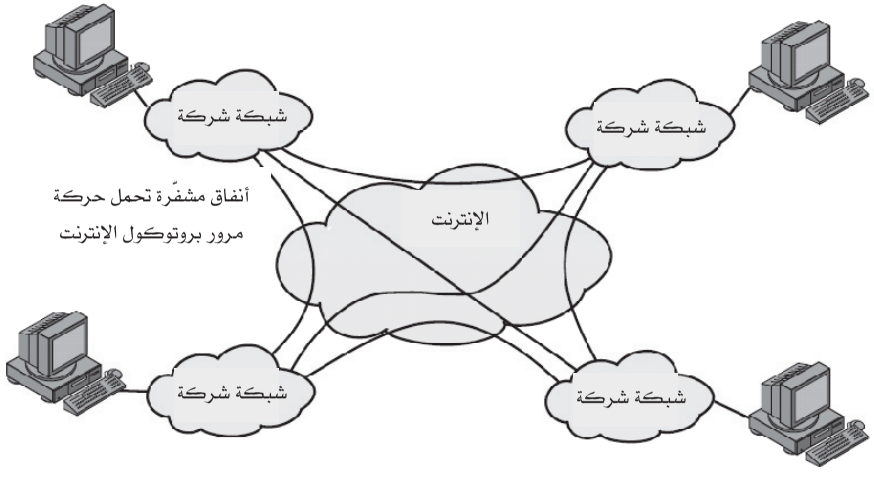
6-4-3 الحشو

يخدم حقل الحشوة عدة أغراض:

- إذا كانت خوارزمية التشفير تتطلب أن يكون النص الأصلي عدداً معيّناً من البايتات (على سبيل المثال، عدداً صحيحاً من الكتل لأحد شفرات الكتلة) يُستخدَم حقل الحشو لتوسيع النص الأصلي (والذي يتألف من حقول بيانات الحمولة، والحشوة، وطول الحشوة، والترويسة التالية) إلى الطول المطلوب.
- تتطلب صيغةُ التغليف الأمني للحمولة أن تكون حقول طول الحشوة والترويسة التالية منحازة يميناً داخل كلمة مؤلَّفة من 32 بتاً. وفي المقابل، يجب أن يكون النص المُشفَّر عدداً صحيحاً من الكلمات. ويُستخدَم حقل الحشو لتحقيق تلك المحاذاة.
- قد يضاف المزيد من الحشو لتوفير سريّة جزئيةٍ لتدفق حركة المرور عن طريق إخفاء الطول الفعلي للحمل الآجر.



(a) أمن مستوى النقل



(b) شبكة خاصة افتراضية بنمط النفق

الشكل 6-8: التشفير بنمط النقل مقابل التشفير بنمط النفق.

6-4-4 نمطا النقل والنفق

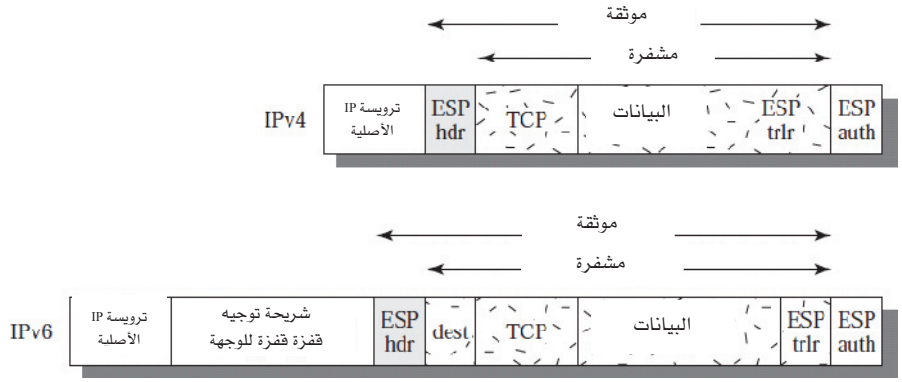
يبيّن الشكل 6-8 أسلوبين لاستخدام خدمة التغليف الأمني للحمولة في بروتوكول IPSec. في الجزء العلوي من الشكل (6-8 (a))، يتم توفير التشفير (والتوثيق اختيارياً) بين المضيفين مباشرة. ويبيّن الشكل 6-8 (b) كيفية استخدام عملية نمط النفق لإنشاء شبكة خاصة افتراضية. وفي هذا المثال، تمتلك منظمة أربع شبكات خاصة مرتبطة عبر الإنترنت. ويستخدم المضيفون على الشبكات الداخلية شبكة الإنترنت لنقل البيانات ولكنهم لا يتفاعلون مع غيرهم من مضيفي الإنترنت. وبإنهاء الأنفاق عند بوابة الأمن لكل شبكة داخلية، يسمح هذا الترتيب للمضيفين بتجنب تنفيذ الإمكانات الأمنية. ويتم دعم الأسلوب السابق بواسطة ارتباط أمني بنمط النقل، في حين أن الأسلوب الأخير يستخدم ارتباطاً أمنياً بنمط النفق.

في هذا الجزء، سنلقي نظرةً على مجالات التغليف الأمني للحمولة لكلا النمطين، مع ملاحظة أن الاعتبارات مختلفة بعض الشيء بالنسبة لـ IPv4 وIPv6. وكما كان الحال في مناقشتنا عن مجالات ترويسة التوثيق، سوف نستخدم صيغة الرزمة بالشكل 6-6 (a) كنقطة انطلاق.

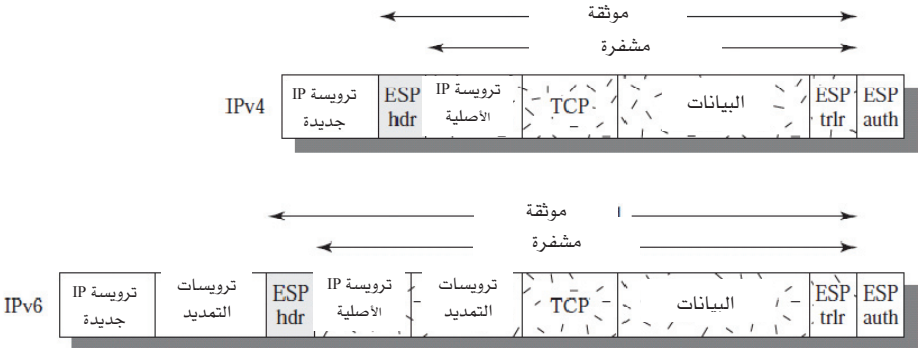
❖ التغليف الأمني للحمولة بنمط النقل:

يستخدم التغليف الأمني للحمولة بنمط النقل للتشفير والتوثيق الاختياري للبيانات التي يحملها بروتوكول الإنترنت (على سبيل المثال، قطعة TCP) كما هو مبيّن في الشكل 6-9 (a). وبالنسبة لـ IPv4 يتم في هذا النمط إدخال ترويسة التغليف الأمني للحمولة في رزمة IP مباشرةً قبل ترويسة طبقة النقل (على سبيل المثال: UDP، TCP، ICMP) ويتم وضع تذييل للتغليف الأمني للحمولة (حقول الحشوة، طول الحشوة، والترويسة التالية) بعد رزمة IP، في حال اختيار التوثيق، يتم إضافة حقل بيانات توثيق التغليف الأمني للحمولة بعد تذييل التغليف الأمني للحمولة. ثم يتم

تشفير القطعة بمستوى النقل كاملةً بالإضافة إلى التذييل. ويشمل التوثيق كل النص المُشفَّر بالإضافة إلى ترويسة التغليف الأمني للحمولة.



(a) نمط النقل



(b) نمط النفق

الشكل 6-9: مجال التشفير والتوثيق للتَّغليف الأمني للحمولة.

في سياق IPv6 يُنظر للتغليف الأمني للحمولة على أنه حمولة من طرفٍ إلى طرف، بمعنى أنه لا يُفحص أو يُعالج من قِبَل الموجّهات الوسيطة. وعليه تظهر

ترويسة التوثيق بعد ترويسة الأساس الـ IPv6 وترويسات القفزة - بقفزة، والتوجيه، والترويسات الجزئية الملحقة. ويمكن أن تظهر ترويسة خيارات الوِجهة الملحقة قبل ترويسة التغليف الأمني للحمولة أو بعدها تبعاً للدلالات المرغوبة. وبالنسبة لـ IPv6، يغطي التشفير قطعة مستوى النقل كاملةً بالإضافة إلى تذييل التغليف الأمني للحمولةِ وترويسة خيارات الوِجهة الملحقة في حال وقوعها بعد ترويسة التغليف الأمني للحمولة. ومرةً أخرى، يشمل التوثيق النص المُشفَّر كاملا بالإضافة إلى ترويسة التغليف الأمني للحمولة.

يمكن تلخيص عمل نمط النقل على النحو الآتي:

1. عند المصدر، يتم تشفير كتلة البيانات التي تتكون من تذييل التغليف الأمني للحمولة بالإضافة إلى كل قطعة طبقة النقل، كما يتم استبدال النص الأصلي لتلك الكتلة بالنص المُشفَّر لتشكيل رزمة IP بغرض النقل. ويتم إضافة التوثيق إذا استخدم هذا الخيار.
2. يتم بعد ذلك توجيه الرزمة للوِجهة المطلوبة. ويقوم كل موجّه وسيط بفحص ومعالجة ترويسة IP بالإضافة إلى أي ترويسات IP ملحقة بالنص الأصلي، ولكنه لا يحتاج إلى فحص النص المُشفَّر.
3. تقوم عقدة الوِجهة بفحص ومعالجة ترويسة IP، بالإضافة إلى أي ترويسات IP ملحقة بالنص الأصلي. ثم تقوم بفك تشفير ما تبقى من الرزمة لاسترداد النص الأصلي بقطعة طبقة النقل، وذلك بناءً على مؤشر المُعامِلات الأمنية (SPI).

يوفر نمط النقل السرية لأي تطبيق يستخدم هذا النمط، ومن ثَمَّ لا يحتاج إلى تنفيذ السرية في كل تطبيق على حدة. كما يتميز أيضاً بدرجة معقولة من الكفاءة، حيث إنه لا يؤدي إلا إلى زيادة طفيفة في الطول الإجمالي لرزمة IP. ولكن من عيوب هذا النمط إمكانية القيام بتحليل لحركة مرور الرزم المُرسَلة بواسطته.

❖ التغليف الأمني للحمولة بنمط النفق:

يُستخدم التغليف الأمني للحمولة بنمط النفق لتشفير رزم IP بكاملها (الشكل 9-6 (b)). في هذا النمط، تُثبَّت ترويسة التغليف الأمني للحمولة في مُقدِّمة الرزمة ثم يتم تشفير الرزمة بالإضافة إلى تذييل التغليف الأمني للحمولة. ويمكن استخدام هذه الطريقة لإحباط محاولات تحليل حركة المرور.

نظراً لأن ترويسة IP تتضمّن عنوان الوِجهة وربما أيضاً تعليمات توجيه المصدر ومعلومات خيار القفزة - بقفزة، فلن يكون بالإمكان نقل رزم IP المُشفَّرة مع ترويسة التغليف الأمني للحمولةِ المثبتة في مقدمتها– حيث لن تستطيع الموجّهات الوسيطة معالجة تلك الرزم. ولذا فمن الضروري تغليف الكتلة بأكملها (ترويسة التغليف الأمني للحمولة، بالإضافة إلى النص المُشفَّر، وبيانات التوثيق إن وجدت) مع ترويسة IP جديدة والتي سوف تتضمّن معلومات كافية للتوجيه ولكن ليس لتحليل حركة المرور.

في حين يُعدُّ نمط النقل مناسباً لحماية الاتصالات بين الأنظمة المضيفة التي تدعم ميزة التغليف الأمني للحمولة، يُعدُّ نمط النفق مفيداً في التشكيلات التي بها جدار حماية أو أي نوع آخر من البوابات الأمنية التي تحمي شبكة موثوق بها من شبكات خارجية. وفي تلك الحالة الأخيرة يحدث التشفير فقط بين مضيف خارجي وبوابة أمنية أو بين اثنتين من البوابات الأمنية. وهذا يعفي الأنظمة المضيفة على الشبكة الداخلية من عبء معالجة التشفير ويبسِّط عملية توزيع المفاتيح عن طريق خفض عدد المفاتيح المطلوبة. بالإضافة إلى ذلك فإنه يحبط محاولات تحليل حركة المرور على أساس الوِجهة النهائية.

لنأخذ بعين الاعتبار حالة مضيف خارجي يرغب في الاتصال مع مضيف على شبكة داخلية محميّة بواسطة جدار حماية، علماً بأن التغليف الأمني للحمولة مُنفَّذ بالمضيف الخارجي وبجدران الحماية. ويتم اتباع الخطوات الآتية لنقل قطعة من طبقة النقل من المضيف الخارجي إلى المضيف الداخلي:

1. يقوم المصدر بإعداد رزمة IP داخلية عنوان الوِجهة فيها هو عنوان المضيف الداخلي المستهدف. تُثبَّت ترويسة التغليف الأمني للحمولة في مقدمة تلك الرزمة، ثم يتم تشفير الرزمة مع تذييل التغليف الأمني للحمولة، كما

يمكن إضافة بيانات التوثيق. يتم تغليف الكتلة الناتجة بترويسة IP جديدة (ترويسة الأساس مع ملحقات اختيارية كخيارات التوجيه والقفزة - بقفزة لـ IPv6) والتي يكون عنوان الوِجهة فيها هو عنوان جدار الحماية، وبهذا يتم تشكيل رزمة IP الخارجية.

2. يتم توجيه الرزمة الخارجية إلى جدار الحماية. يقوم كل موجّه وسيط بفحص ومعالجة ترويسة IP الخارجية بالإضافة إلى أيٍّ من ترويسات IP الملحقة الخارجية، وبدون حاجة لفحص النص المُشفَّر.
3. يقوم جدار الحماية المقصود بفحص ومعالجة ترويسة IP الخارجية بالإضافة إلى أيٍّ من ترويسات IP الملحقة الخارجية. ثم تعمل عقدة الوِجهة على فك تشفير ما تبقى من الرزمة على أساس مؤشر مُعامِلات الأمن (SPI) بترويسة التغليف الأمني للحمولة لاسترداد النص الأصلي لرزمة IP الداخلية. بعد ذلك تُنقل هذه الرزمة عبر الشبكة الداخلية.
4. يتم توجيه الرزمة الداخلية بعد ذلك عبر عدد من الموجّهات بالشبكة الداخلية لتصل إلى المضيف المقصود.

5-6 الجمع بين الارتباطات الأمنية

يمكن لارتباط أمني واحد استخدام ترويسة التوثيق (AH) أو التغليف الأمني للحمولةِ (ESP) ولكن ليس كلاهما معاً. وفي بعض الأحيان يتطلب تدفق معيّن لحركة المرور استخدام الخدمات التي تقدمها ترويسة التوثيق والتغليف الأمني للحمولة. كما قد يتطلب تدفق معيّن خدمات أمن IPSec بين الأنظمة المضيفة، وفي نفس الوقت قد يتطلب هذا التدفق نفسه خدمات أخرى بين البوابات الأمنية، كجدران الحماية. في جميع هذه الحالات ينبغي استخدام عدة ارتباطات أمنية لتحقيق خدمات IPSec المنشودة لنفس تدفق حركة المرور. يُمثل مصطلح "رزمة الارتباطات الأمنية (security association bundle) سلسلة من الارتباطات الأمنية التي يتم معالجة حركة المرور من خلالها لتوفير المجموعة المطلوبة من خدمات أمن IPSec. هذا وقد تنتهي الارتباطات الأمنية برزمةٍ ما في نقاط نهاية مختلفة أو في نفس النقطة النهاية.

ويمكن الجمع بين الارتباطات الأمنية في رزم بطريقتين:

- جوار النقل: ويشير إلى تطبيق أكثر من بروتوكول أمني على نفس رزمة IP، دون الاستعانة بنفق. تسمح هذه الطريقة للجمع ما بين ترويسة التوثيق والتغليف الأمني للحمولة بمستوى واحد فقط من هذا الجمع؛ لأن تراكبها (nesting) لا يُؤدي أي فائدة إضافية لكون المعالجة تتم في موقع أمن IPSec واحد، ألا وهي الوِجهة النهائية.
- النفق المتكرر: ويشير إلى تطبيق طبقات متعددة من البروتوكولات الأمنية تنفّذ من خلال نفق بروتوكول IP. وتتيح هذه الطريقة استخدام مستويات متعددة من التراكب (nesting)، لأن كل نفق يمكن إنشاؤه أو إنهاؤه في مواقع IPSec مختلفة على طول المسار.

ويمكن الجمع بين النهجين، على سبيل المثال بجعل ارتباط أمني بنمط النقل يربط بين أنظمة مضيفة وينتقل في جزء من مساره بين البوابات الأمنية خلال ارتباط أمني بنمط النفق.

ومن قضايا رزم الارتباط الأمني الشائقة قضية الترتيب الذي تتم به عمليات التوثيق والتشفير بين زوج معيّن من النقاط الطرفية وسبل تحقيق ذلك. وسوف نتناول هذه المسألة فيما يلي، ثم ننظر إلى ترتيبات الارتباطات الأمنية التي تتضمّن نفقاً واحداً على الأقل.

1-5-6 التوثيق بالإضافة إلى السرية

يمكن الجمع بين التشفير والتوثيق بغية نقل رزمة IP بخاصيتي التوثيق والسرية بين أنظمة مضيفة. وسوف ننظر إلى عدة سبل لتحقيق ذلك.

❖ خيار التوثيق مع التغليف الأمني للحمولة:

يوضّح الشكل 6-9 هذه الطريقة، حيث يُطبّق المستخدم أولاً التغليف الأمني للحمولة على البيانات التي يتعين حمايتها ثم يلحق بها حقل بيانات التوثيق. وهناك في الواقع حالتان فرعيتان:

- التغليف الأمني للحمولة بنمط النقل: يُطبّق كلٌّ من التوثيق والتشفير على حمولة IP التي يتم تسليمها إلى المضيف، أما الترويسة (IP) فلا يتم حمايتها.
- التغليف الأمني للحمولةِ بنمط النفق: يُطبّق التوثيق على رزمة IP بالكامل والتي يتم تسليمها إلى عنوان IP للوِجهة الخارجية (على سبيل المثال، جدار الحماية)، ويتم التوثيق عند تلك الوِجهة. كما يتم حماية رزمة IP الداخلية بكاملها بموجب آليات الخصوصية، وذلك لتسليمها إلى عنوان IP للوِجهة الداخلية.

وفي كلتا الحالتين، يُطبّق التوثيق على النص المُشفَّر وليس النص الأصلي.

❖ تجاور النقل:

هناك طريقة أخرى لتطبيق التوثيق بعد التشفير باستخدام رزمة مكوَّنة من ارتباطين أمنيين بنمط النقل، حيث يكون الداخلي منهما ارتباطاً أمنياً مع تغليف أمني للحمولة والخارجي ارتباطاً أمنياً مع ترويسة التوثيق. وفي هذه الحالة يتم استخدام التغليف الأمني للحمولة بدون خيار التوثيق. ولأن الارتباط الأمني الداخلي بنمط النقل، فإنه يتم تشفير حمولة IP. وتتألف الرزمة الناتجة من ترويسة IP (وربما ترويسات IPv6 ملحقة) يليها التغليف الأمني للحمولة. بعدها يتم تطبيق ترويسة التوثيق بنمط النقل، بحيث يغطي التوثيق التغليف الأمني للحمولة مع ترويسة IP الأصلية (وملحقاتها) باستثناء الحقول القابلة للتغيير. وتمتاز هذه الطريقة على مجرد استخدام ارتباط أمني واحد مع التغليف الأمني للحمولةِ بخيار التوثيق بأن التوثيق يشمل المزيد من الحقول، بما في ذلك عناوين IP للمصدر والوِجهة. أما عيب هذه الطريقة فهو زيادة العبء الإضافي نتيجة استخدام ارتباطين أمنيّين بدلاً من ارتباطٍ واحد.

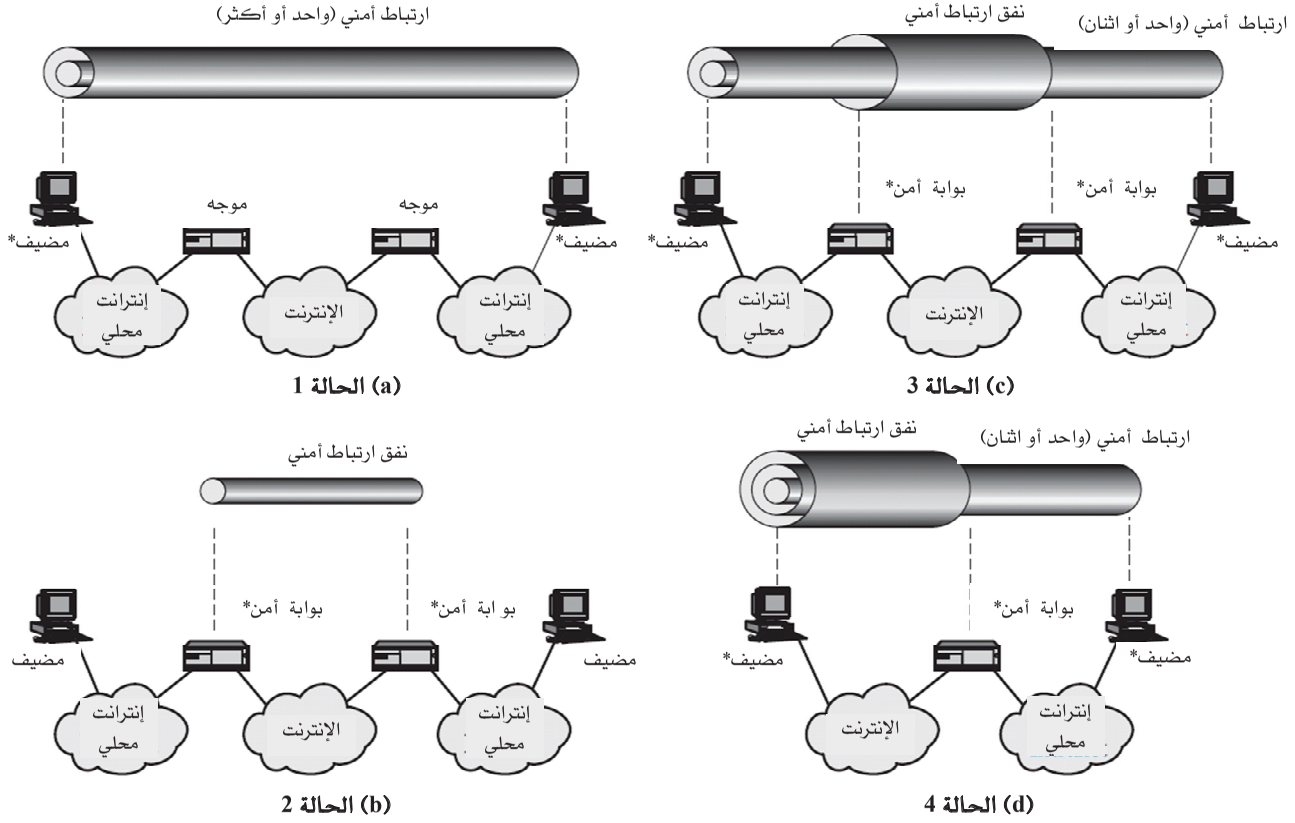
❖ رزمة النقل - النفق:

قد يكون من الأفضل استخدام التوثيق قبل التشفير لعدة أسباب. أولها: أن بيانات التوثيق تكون محمية بواسطة التشفير، ومن ثَمَّ سيكون من المستحيل على أي شخص اعتراض الرسالة وتغيير بيانات التوثيق بها دون أن يتم اكتشاف ذلك، وثانيها، قد يكون من المستحسن تخزين معلومات التوثيق مع الرسالة عند المُستلِم للرجوع إليها إذا ما دعت الحاجة لذلك لاحقاً. ويُعدُّ ذلك أكثر ملاءمة إذا كانت معلومات التوثيق تُطبّق على الرسالة غير المُشفَّرة؛ وإلا فإن الرسالة يجب أن يتم إعادة تشفيرها للتحقق من معلومات التوثيق.

تتلخص إحدى طرق تطبيق التوثيق قبل التشفير بين نظامين مضيفين في استخدام رزمة تتألف من ارتباطين، داخلي بترويسة توثيق بنمط النقل وخارجي بالتغليف الأمني للحمولةِ بنمط النفق. وفي هذه الحالة، يتم تطبيق التوثيق على حمولة IP بالإضافة إلى ترويسة IP (وملحقاتها) باستثناء الحقول القابلة للتغيير. ويتم بعد ذلك معالجة رزمة IP الناتجة في نمط النفق بواسطة التغليف الأمني للحمولة، والنتيجة هي تشفير الرزمة الداخلية الموثقة كاملةً وإضافة ترويسة IP خارجية جديدة (وملحقاتها).

2-5-6 التوليفات الأساسية للارتباطات الأمنية

تتضمن وثيقة بنية IPSec أربعة أمثلة على توليفات للارتباطات الأمنية التي يتعيّن على الأنظمة المضيفة المتوافقة مع IPSec (كمحطات العمل، والخادمات) أو بوابات الأمن (كجدران الحماية، والموجّهات) دعمها، كما هو موضح بالشكل 6-10. ويمثل الجزء السفلي من الشكل التوصيل المادي للعناصر ؛ بينما يمثل الجزء العلوي التوصيل المنطقي عن طريق ارتباط أمني واحد أو عدة ارتباطات أمنية متراكبة (nested). يمكن أن يكون كل ارتباط أمني ترويسة توثيق أو تغليفاً أمنياً للحمولة. ويمكن أن تكون الارتباطات الأمنية من مضيف إلى مضيف إما بنمط النقل أو نمط النفق، وإلا فإنه يتعيّن أن يكون الارتباط بنمط النفق.



الشكل 6-10: التوليفات الأساسية للارتباطات الأمنية (* = تطبق بروتوكول أمن الانترنت).

في الحالة 1، يتم توفير كل إجراءات الأمن بين الأنظمة الطرفية التي تنفذ أمن بروتوكول IPSec. للاتصال بين نظامين طرفيّين عبر الارتباط الأمني لابد أن يشتركا في المفاتيح السرية المناسبة. ومن بين التوليفات الممكنة:

1. ترويسة التوثيق بنمط النقل.
2. التغليف الأمني للحمولة بنمط النقل.
3. التغليف الأمني للحمولة متبوعاً بترويسة توثيق بنمط النقل (ارتباط أمني بالتغليف الأمني للحمولة داخل ارتباط أمني بترويسة التوثيق).
4. أيٍّ من الحالات السابقة (1، أو 2، أو 3) داخل ترويسة التوثيق أو داخل التغليف الأمني للحمولة بنمط النفق.

ناقشنا حتى الآن كيف يمكن استخدام تلك التوليفات المختلفة لدعم التوثيق، والتشفير، والتوثيق قبل التشفير، والتوثيق بعد التشفير.

في الحالة 2، يتم توفير الأمن بين البوابات الأمنية فقط (الموجّهات، جدران الحماية، الخ) ولا تقوم الأنظمة المضيفة بتنفيذ بروتوكول IPSec. وتوضح هذه الحالة الدعم للشبكات الخاصة الافتراضية (VPN) البسيطة. وتنص وثيقة بنية الأمن على أن المطلوب لهذه الحالة هو ارتباط أمني بنفق واحد فقط. ويمكن للنفق دعم ترويسة توثيق أو التغليف الأمني للحمولة أو التغليف الأمني للحمولة مع خيار التوثيق. لا يحتاج الأمر هنا إلى أنفاق متراكبة لأن خدمات IPSec تُطبّق على الرزمة الداخلية برمتها.

الحالة 3 مبنية على الحالة 2 وذلك بإضافة نظام أمن من طرف إلى طرف. يُسمح هنا بنفس التوليفات التي تمت مناقشتها في الحالتين 1 و2. ويوفّر نفق البوابة إلى بوابة التوثيق أو السرية أو كليهما لكامل حركة المرور بين الأنظمة الطرفية. عندما يتبع نفق البوابة إلى بوابة نظام التغليف الأمني للحمولة، فإنه يوفر أيضاً شكلاً محدوداً من سرية حركة المرور. ويمكن للمضيفين كلٍ على حدة تنفيذ أي خدمات IPSec إضافية تكون مطلوبة لتطبيقات معينة، أو لمستخدمين معيّنين عن طريق ارتباطات أمنية من طرفٍ إلى طرف.

توفر الحالة 4 الدعم لمضيف بعيد يستخدم الإنترنت للوصول إلى جدار الحماية للمؤسسة ومن ثم الوصول إلى خادم أو محطة العمل خلف جدار الحماية. المطلوب هنا فقط هو نمط النفق بين المضيف البعيد وجدار الحماية. وكما في الحالة رقم 1، يمكن استخدام ارتباط أمني واحد أو ارتباطين بين المضيف البعيد والمضيف المحلي.

6-6 إدارة المفاتيح

يتضمّن الجزء الخاص بإدارة المفاتيح في IPSec تحديد المفاتيح السرية وتوزيعها. ويحتاج الأمر عادةً إلى أربعة مفاتيح للاتصال بين تطبيقين: زوج مفاتيح (مفتاح للإرسال وآخر للاستقبال) لكلٍّ من ترويسة التوثيق والتغليف الأمني للحمولة. وتتطلب وثيقة بنية IPSec دعم نوعين من أساليب إدارة المفاتيح:

- يدوي: يقوم مسئول النظام يدوياً بتهيئة كل نظام مع المفاتيح الخاصة به والمفاتيح الخاصة بالأنظمة الأخرى المُتصلة. ويُعدُّ هذا الأسلوب عملياً أكثر في البيئات الصغيرة الثابتة نسبيا.
- آلي: حيث يمكن استحداث مفاتيح جديدة للارتباطات الأمنية آلياً حسب الطلب، ويسهِّل عملية استخدام المفاتيح في نظام مُوزَّع كبير دائم التطور.

يُعرف بروتوكول إدارة المفاتيح الآلي المفترض في بروتوكول IPSec بـ (ISAKMP/Oakley)، ويتألف من العناصر الآتية:

- بروتوكول أوكلي لتعيين المفاتيح (Oakley Key Determination Protocol): أوكلي هو بروتوكول لتبادل المفاتيح يستند إلى خوارزمية ديفي- هيلمان مع توفير أمن إضافي. يُعدُّ أوكلي بروتوكولاً عاما لا يتطلب صيغاً بعينها.
- بروتوكول ارتباط أمن الإنترنت وإدارة المفاتيح (Internet Security Association and Key Management Protocol (ISAKMP)): يوفر بروتوكول ارتباط أمن الإنترنت وإدارة المفاتيح (ISAKMP) إطاراً لإدارة مفاتيح الإنترنت، كما يوفر الدعم المحدّد للبروتوكول، بما في ذلك الصيغ، للتفاوض على السمات الأمنية.

ولا يُملي ISAKMP خوارزمية محدّدة لتبادل المفاتيح؛ ولكنه يتكون من مجموعة من أنواع الرسائل تسمح باستخدام مجموعة متنوعة من خوارزميات تبادل المفاتيح. حدّد الإصدار الأوليّ من بروتوكول ISAKMP أوكلي كخوارزمية تبادل المفاتيح المعتمدة.

نبدأ هذا الجزء بلمحة عامة عن بروتوكول أوكلي، ثم نتناول بعد ذلك بروتوكول ISAKMP.

1-6-6 بروتوكول أوكلي لتعيين المفاتيح

يمكن اعتبار بروتوكول أوكلي تنقيحاً لخوارزمية ديفي- هيلمان لتبادل المفاتيح. تذكّر أن خوارزمية ديفي- هيلمان تتضمّن التبادلات التالية والتي تتم بين المستخدمين A و B. ولإتمام تلك التبادلات لابد من وجود اتفاق مسبق على متغيرين عامَّين: q وهوعدد أوَّلي كبير؛ وα وهو جذر بدائي للعدد الأوَّلي q. حيث يختار A عدداً صحيحاً عشوائياً (X_A) كمفتاحٍ خاصٍ به، ويرسل إلى B مفتاحه العام ($Y_A = \alpha^{X_A} \bmod q$). وبنفس الطريقة، يختار B عدداً صحيحاً عشوائياً X_B كمفتاحٍ خاصٍ به، ويرسل إلى A مفتاحه العام ($Y_B = \alpha^{X_B} \bmod q$). عندها يمكن لكل جانب حساب مفتاح الجلسة السري من:

وتتمتع خوارزمية ديفي - هيلمان بميزتين مهمتين:

- يتم إنشاء المفاتيح السرية عند الحاجة فقط. ولذا فلا حاجة لتخزين المفاتيح السرية لفترة طويلة من الزمن والذي قد يشكّل تهديداً أمنياً.
- لا يتطلب التبادل وجود بنية تحتيَّة فيما عدا الاتفاق على المتغيرين العامَّين.

ومع ذلك، فهناك عدد من نقاط الضعف في خوارزمية ديفي - هيلمان كما ورد في [HUIT98]:

- أنها لا تقدم أي معلومات عن هوية الأطراف.

- أنها عُرضةٌ لهجوم "رجل في الوسط"، والذي يقوم فيه طرف ثالث C بالاتصال مع الطرف A متقمصاً شخصية الطرف B وكذلك الاتصال مع الطرف B متقمصاً شخصية الطرف A. يؤدي ذلك في النهاية إلى أنَّ كلاً من A وB يقومان بالفعل بالتفاوض على المفتاح مع الطرف C وليس مع بعضهما بعضاً، مما يمكّن C بعد ذلك من التنصت على حركة المرور وتمريرها. يمضي هجوم "رجلٌ في الوسط" على النحو الآتي:

1. يرسل الطرف B مفتاحه العام Y_B في رسالة موجهة إلى الطرف A (انظر الشكل 3-11).
2. يقوم الخَصْم E باعتراض تلك الرسالة، ويحفظ المفتاح العام للطرف B ثم يقوم بإرسال رسالةٍ إلى الطرف A بهوية (User ID) الخاصة بالطرف B ولكن مع المفتاح العام للخَصْم E. تُرسَل هذه الرسالة بهذه الطريقة حتى تبدو كما لوكانت قد أُرسلت من النظام المضيف للطرف B. باستلام هذه الرسالة من الخَصْم E يقوم الطرف A بحفظ المفتاح العام للخَصْم E مع هوية الطرف B. وبالمثل، يرسِل الخَصْم E رسالة إلى الطرف B مع المفتاح العام للخَصْم E، والتي تزعم أنها من الطرف A.
3. يقوم B بحساب مفتاح سري K_1 بناءً على المفتاح الخاص للطرف B وY_E. ويقوم A بحساب مفتاح سري K_2 بناءً على المفتاح الخاص للطرف A وY_E. يقوم الخَصْم E بحساب K_1 مستخدماً مفتاح E السري X_E وY_B وحساب K_2 مستخدماً X_E وY_A.
4. من الآن فصاعداً يصبح الخَصْم E قادراً على توصيل الرسائل من الطرف A إلى الطرف B ومن الطرف B إلى الطرف A، وتغيير تشفيرهما كما يريد دون أن يدري الطرف A ولا الطرف B أن الخَصْم E يشاركهما اتصالاتهما.

- تُعدُّ الخوارزمية عالية الكلفة حسابياً حيث تتطلب عمليات حسابية مُكثّفة وتستغرق وقتاً طويلاً، مما يعرضها لهجمات العرقلة (clogging attacks)، حيث يطلب الخَصْم عدداً كبيراً من المفاتيح، ويستهلك الضحية لذلك موارد معالجة ضخمة للقيام بعمليات الرفع الأسّي المقاسي (modular exponentiation) بدون طائل بدلاً من القيام بعمل مفيد.

$I \rightarrow R$: $CKY_I, OK_KEYX, GRP, g^x, EHAO, NIDP, ID_I, ID_R, N_I, S_{KI}[ID_I \| ID_R \| N_I \| GRP \| g^x \| EHAO]$

$R \rightarrow I$: $CKY_R, CKY_I, OK_KEYX, GRP, g^y, EHAS, NIDP, ID_R, ID_I, N_R, N_I, S_{KR}[ID_R \| ID_I \| N_R \| N_I \| GRP \| g^y \| g^x \| EHAS]$

$I \rightarrow R$: $CKY_I, CKY_R, OK_KEYX, GRP, g^x, EHAS, NIDP, ID_I, ID_R, N_I, N_R, S_{KI}[ID_I \| ID_R \| N_I \| N_R \| GRP \| g^x \| g^y \| EHAS]$

الرموز:

I = المنشئ (البادئ)

R = المستجيب

CKY_I, CKY_R = كوكيز البادئ والمستجيب

OK_KEYX = نوع رسالة تبادل المفاتيح

GRP = اسم مجموعة ديفي – هيلمان لهذا التبادل

g^x, g^y = المفتاحان العامان للبادئ والمستجيب؛ g^{xy} = مفتاح الجلسة نتيجة هذا التبادل

EHAO, EHAS = وظيفة التشفير والتوثيق بالهاش التي تم عرضها واختيارها

NIDP = تدل على عدم استخدام التشفير لبقية هذه الرسالة

ID_I, ID_R = مُعرِّفا البادئ والمستجيب

N_I, N_R = عددان عشوائيان يستخدمان لمرة واحدة (nonce) يوفرهما البادئ والمستجيب لهذا التبادل

$S_{KI}[X], S_{KR}[X]$ = ترمزان للتوقيعين على X باستخدام المفتاح الخاص (مفتاح التوقيع) للبادئ والمستجيب

الشكل 6-11: مثال أوكلي للتبادل الاعتسافي للمفاتيح.

يهدف بروتوكول أوكلي إلى الحفاظ على مزايا خوارزمية ديفي- هيلمان وفي نفس الوقت تلافي نقاط الضعف فيها.

❖ خصائص بروتوكول أوكلي:

يتميز بروتوكول أوكلي بخمس خصائص مهمة:

1. تستخدم آلية تُعرف باسم الكوكيز (cookies) لإحباط هجمات العرقلة.
2. تمكِّن الطرفين من التفاوض على مجموعة (group)؛ ومن ثم تحديد المتغيرين العامَّين لخوارزمية ديفي - هيلمان لتبادل المفاتيح.
3. تستعمل أعدد الاستخدام لمرة واحدة فقط (nonces) للحماية ضد هجمات إعادة الإرسال.
4. تُتيح تبادل قيم مفاتيح ديفي - هيلمان العامة.
5. تقوم بتوثيق تبادلات ديفي - هيلمان لإحباط هجمات "رجل في الوسط".

بعد أن ناقشنا خوارزمية ديفي - هيلمان، لنلقِ نظرةَ على ما تبقى من تلك الخصائص بالترتيب. ننظر أولاً إلى مشكلة هجمات العرقلة. في هذا الهجوم يزيِّف الخَصْم عنوان المصدر لمستخدم شرعي ويرسل مفتاح ديفي - هيلمان عمومي إلى الضحية. ينفِّذ الضحية عمليات الرفع الأسّي المقاسي (modular exponentiation) لحساب المفتاح السري. وبتكرار رسائل من هذا النوع يمكن عرقلة نظام الضحية بحسابات وعمليات غير مُجدية. ويتطلب تبادل الكوكي أن يرسل كل جانب عدداً شِبه عشوائي، أي الكوكي، في الرسالة الأوليّة، والتي يقر الجانب الآخر بتلقّيها. وينبغي تكرار هذا الإقرار في الرسالة الأولى لتبادل مفاتيح ديفي - هيلمان، وإذا كان عنوان المصدر مزوراً لا يحصل الخَصْم على أي جواب. ومن ثَمَّ، يمكن للخَصْم جعل المستخدم يولّد إشعارات استلام فقط وليس القيام بحسابات ديفي - هيلمان المعقّدة.

يتطلب بروتوكول ISAKMP أن يلبي توليد الكوكيز المتطلبات الثلاث الأساسية الآتية:

1. يجب أن يعتمد الكوكي على الأطراف المعنية. وبهذا يُمنَع المهاجِم من الحصول على الكوكي باستخدام عنوان IP ومنفذ UDP حقيقين ثم استعمال ذلك الكوكي لإغراق الضحية بالطلبات المقدّمة من عناوين IP أو منافذ تم اختيارها عشوائياً.
2. يجب ألا يكون بإمكان أي شخص آخر غير الجهة المُصدرة توليد كوكيز تكون مقبولة لدى تلك الجهة. ويعني ذلك ضمناً أن الجهة المُصدرة ستَستخدم معلومات سرية محلية في توليد الكوكي والتحقق منه لاحقاً. ويجب أن لا يكون بالإمكان استخلاص تلك المعلومات السرية من أي كوكي. والهدف من هذا الشرط هو أن الجهة المُصدرة لن تحتاج إلى حفظ نسخ من الكوكيز الخاصة بها مما قد يجعلها أكثر عرضة للانكشاف، ولكن سيمكنها التحقق من أي إشعار استلام لكوكي عندما تحتاج إلى ذلك.
3. يجب أن تكون أساليب توليد الكوكي والتحقق منه سريعة لإحباط الهجمات التي تهدف إلى استهلال موارد المعالج.

يتلخص الأسلوب الموصى به لإنشاء كوكي في استخدام دالة من دوال التحوير (hashing) السريعة (على سبيل المثال: MD5) على عناوين IP للمصدر والوِجهة، ومنافذ UDP للمصدر والوِجهة، وقيمة سريّة مولّدة محليا.

تدعم خوارزمية أوكلي استخدام مجموعات مختلفة من مفاتيح تبادل ديفي-هيلمان، حيث تضم كل مجموعة تعريف المتغيرين العامّين وهوية الخوارزمية. وتشمل المواصفات الحالية المجموعات الآتية:

- الرفع الأسّي المقاسي مع مُعامِل من 768 بتاً:

$q = 2^{768} - 2^{704} - 1 + 2^{64} \times (\lfloor 2^{638} \times \pi \rfloor + 149686)$

$\alpha = 2$

- الرفع الأسّي المقاسي مع مُعامِل من 1024 بتاً:

$q = 2^{1024} - 2^{960} - 1 + 2^{64} \times (\lfloor 2^{894} \times \pi \rfloor + 129093)$

$\alpha = 2$

- الرفع الأسّي المقاسي مع مُعامِل من 1536 بتاً:
  - سيتم تحديد المتغيرات.
- مجموعة المنحنى الإهليلجي (Elliptic curve group) على 2^{155}:
  - المولد (الست عشري): X=7B, Y=1C8.
  - متغيرات المنحنى الإهليلجي (الست عشرية): A=0, Y= 7338F.
- مجموعة المنحنى الإهليلجي (Elliptic curve group) على 2^{185}:
  - المولد (الست عشري): X=18, Y=D.
  - متغيرات المنحنى الإهليلجي (الست عشرية): A=0, Y= 1EE9.

المجموعات الثلاث الأُوَل هي خوارزمية ديفي - هيلمان التقليدية باستخدام الرفع الأسّي المقاسي. تستخدم المجموعتان الأخريان المنحنى الإهليلجي المناظر لديفي- هيلمان.

تستعمل خوارزمية أوكلي الأعداد التي تُستخدم لمرة واحدة فقط للحماية ضد هجمات إعادة الإرسال. ويتم توليد كل عدد من تلك الأعداد محلياً كعدد شبه عشوائي. تظهر تلك الأعداد في الردود ويتم تشفيرها أثناء بعض الإجراءات في عملية التبادل لأغراض الحماية.

يمكن استخدام ثلاثة أساليب مختلفة للتوثيق مع خوارزمية أوكلي:

- التواقيع الرقمية: يتم توثيق عملية التبادل عبر توقيع ناتج دالة تحوير (هاش) يمكن الحصول عليه بصورة متبادلة؛ ويقوم كل طرف بتشفير الهاش مع المفتاح الخاص به. ويتم توليد الهاش على المتغيرات المهمة؛ كهويات المستخدمين والأعداد التي تُستخدم لمرة واحدة فقط.
- التشفير بالمفاتيح العامة: يتم توثيق عملية التبادل عبر تشفير المتغيرات كهوية المُستخدِمين والأعداد التي تُستخدم لمرة واحدة فقط، وذلك بواسطة المفتاح الخاص بالمرسل.
- التشفير بالمفاتيح المتماثلة: يمكن استخدام مفتاح مستمد من آلية خارج النطاق لتوثيق التبادل بواسطة التشفير المتماثل لمتغيرات عملية التبادل.

❖ مثال لتبادل أوكلي:

تتضمن مواصفات بروتوكول أوكلي عدداً من الأمثلة لعمليات التبادل التي يُسمح بها بموجب البروتوكول. ولإعطاء نكهة لحديثنا عن أوكلي، نورد هنا مثالاً واحداً، يطلق عليه في المواصفات عملية اعتساف تبادل المفاتيح، نظراً لأن التبادل يتم باستخدام ثلاثة رسائل فقط.

يوضّح الشكل 6-11 بروتوكول اعتساف تبادل المفاتيح (aggressive key exchange protocol). في الخطوة الأولى، ينقل البادئ I كوكي، والمجموعة التي ستُستخدَم، ومفتاح ديفي - هيلمان الخاص بـ I لهذا التبادل. كما يشير I أيضاً إلى الخوارزميات التي ستستخدم في هذا التبادل للتشفير بالمفاتيح العامة، وتوليد دالة التحوير (هاش)، والتوثيق. كما تتضمن الرسالة أيضا معرِّفي البادئ I والمستجيب R والعدد الذي سيستخدمه I لمرة واحدة فقط في هذا التبادل. وأخيراً، وباستخدام

المفتاح الخاص بـ I، يُذيِّل I الرسالة بتوقيعه على هويتي I وR، والعدد المستخدم لمرة واحدة، والمجموعة، والمفتاح العام لديفي - هيلمان، والخوارزميات المعروضة للاستخدام.

عندما يتلقى R الرسالة يتحقق من التوقيع باستخدام مفتاح توقيع I العام. ويرسل R إشعاراً بالاستلام وذلك بإعادة إرسال الكوكي الخاص بـ I، ومعرِّفه، وعدده المستخدم مرة واحدة، وأيضاً المجموعة. كما يُدرج R أيضا في الرسالة كوكي، ومفتاحه العام لديفي - هيلمان، والخوارزميات المختارة (والتي يجب أن تكون ضمن الخوارزميات المعروضة)، ومعرِّف R، والعدد الذي يستخدمه R لمرة واحدة في هذا التبادل. وأخيراً، يُذيِّل R الرسالة بتوقيعه باستخدام المفتاح الخاص به على المعرِّفين، والعددين المستخدمين لمرة واحدة، والمجموعة، والمفتاحين العامين لديفي - هيلمان، والخوارزميات المختارة.

عندما يتلقى I الرسالة الثانية يتحقق من التوقيع باستخدام المفتاح العام لـ R. تضمن قيم الأعداد المستخدمة لمرة واحدة أن هذه الرسالة ليست تكراراً لرسالة قديمة. ولإتمام عملية التبادل، لا بد أن يرسل I رسالة إلى R ليؤكد لـ R أنه قد استلم المفتاح العام لـ R.

2-6-6 بروتوكول أمن الإنترنت وإدارة المفاتيح (ISAKMP)

يحدّد بروتوكول أمن الإنترنت وإدارة المفاتيح (ISAKMP) الإجراءات وصيغ الرزم اللازمة لإنشاء الارتباطات الأمنية والتفاوض عليها وتعديلها وإلغائها. وكجزء من عملية إنشاء ارتباط أمني جديد، يحدّد ISAKMP الأحمال الآجرة اللازمة لتبادل بيانات توليد المفاتيح وبيانات التوثيق. وتوفر صيغ الحمولة هذه إطاراً متوافقاً ومستقلاً لبروتوكول تبادل المفاتيح، وخوارزميات التشفير، وآليات التوثيق.

❖ صيغة ترويسة بروتوكول ISAKMP:

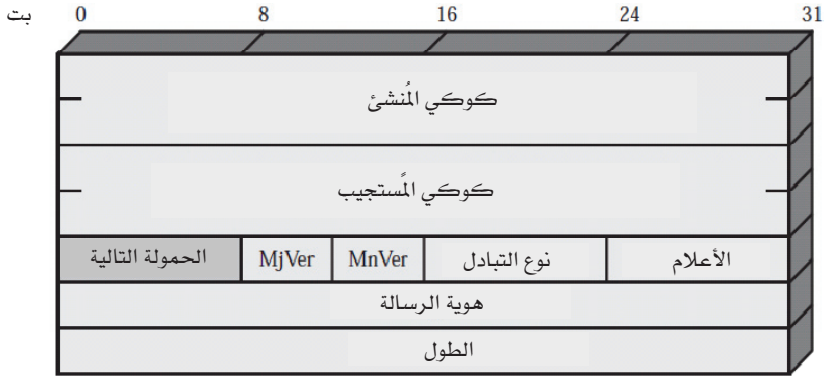
تتكون رسالة ISAKMP من ترويسة ISAKMP يليها حمل آجر واحد أو أكثر، وتنقل كلها ضمن بروتوكول النقل. وتتطلب المواصفات أن تدعم التنفيذات استخدام UDP كبروتوكول للنقل.

يوضّح الشكل 6-12 (a) صيغة الترويسة لرسالة ISAKMP، والتي تتكون من الحقول الآتية:

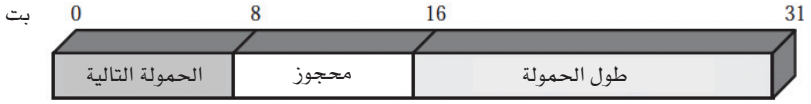
- كوكي البادئ (Initiator Cookie) (64 بتاً): كوكي الجهة التي بادرت بإنشاء الارتباط الأمني، أو الإخطار بالارتباط الأمني، أو إلغاء الارتباط الأمني.
- كوكي المجيب (Responder Cookie) (64 بتاً): كوكي الجهة المجيبة؛ يكون هذا الحقل خالياً في الرسالة الأولى من البادئ.
- الحمولة التالية (Next Payload) (8 بتات): يبيّن نوع الحمولة الأولى في الرسالة؛ سيتم مناقشة الحمولات في الجزء التالي.
- رقم الإصدار الرئيس (Major Version) (4 بتات): يبيّن رقم الإصدار الرئيس لبروتوكول ISAKMP المُستخدَم.
- رقم الإصدار الثانوي (Minor Version) (4 بتات): يبيّن رقم الإصدار الثانوي لبروتوكول ISAKMP المُستخدَم.
- نوع التبادل (Exchange Type) (8 بتات): يبيّن نوع التبادل، سيناقش لاحقاً في هذا الجزء.
- الأعلام (Flags) (8 بتات): تُبين الخيارات المحدّدة لتبادل بروتوكول ISAKMP. وقد تم تحديد اثنين من تلك البتات حتى الآن هما بت التشفير (Encryption) وبت الارتباط (Commit). وتأخذ بت التشفير القيمة 1 إذا كانت جميع الأحمال الآجرة التالية للترويسة قد تم تشفيرها باستخدام خوارزمية التشفير الخاصة بهذا الارتباط الأمني. أما بت الارتباط فيستخدم

لضمان أن المادة المُشفَّرة لا يتم استلامها قبل الانتهاء من إنشاء الارتباط الأمني.

- هوية الرسالة (Message ID) (32 بتاً): هوية فريدة لهذه الرسالة.
- الطول (Length) (32 بتاً): الطول الكلي للرسالة (الترويسة بالإضافة إلى الحمولات) بالبايتات.



(a) ترويسة ISAKMP



(b) ترويسة حمولة عامة

الشكل 6-12: صيغة ISAKMP.

❖ أنواع الحمولة في بروتوكول ISAKMP:

تبدأ جميع أنواع الحمولة في بروتوكول ISAKMP بنفس ترويسة الحمولة العامة المبينة في الشكل 6-12 (b). وتكون قيمة حقل الحمولة التالية صفراً فقط إذا كانت تلك هي الحمولة الأخيرة في الرسالة؛ وإلا فإن قيمتها تحدّد نوع الحمولة

المقبلة. ويدل حقل طول الحمولة على حجم هذه الحمولة بالبايتات بما في ذلك ترويسة الحمولة العامة.

يلخص الجدول 6-3 أنواع الحمولة المحدّدة لبروتوكول ISAKMP، كما يسرد الحقول أو المتغيرات التي تشكّل جزءاً من كل حمولة. وتُستخدم حمولة الارتباط الأمني (SA payload) للبدء في إنشاء الارتباط الأمني. في هذه الحمولة، قيمة متغير نطاق التفسير (DOI) تُحدّد نطاق التفسير الذي تتم المفاوضات في إطاره. ويُعدُّ IPSec أحد الأمثلة لقيم DOI، ولكن يمكن استخدام ISAKMP في سياقات أخرى. أما مُعامِل الحالة (Situation parameter) فيحدّد السياسة الأمنية لتلك المفاوضات، وبشكلٍ أساسي يتم من خلاله تحديد المستويات الأمنية اللازمة للتشفير والسرية (على سبيل المثال، مستوى الحساسية، وحجرة الأمن (security compartment)).

تتضمّن حمولة الاقتراح (Proposal payload) معلومات تُستخدم أثناء التفاوض بخصوص الارتباط الأمني. وتُبيّن الحمولة بروتوكول هذا الارتباط الأمني (إما ترويسة التوثيق أو التغليف الأمني للحمولة) والذي من أجله يتم التفاوض على الخدمات والآليات. كما تتضمّن الحمولة أيضاً المُعامِل الأمني (SPI) للجهة المُرسِلة وعدد التحويلات (transforms). وجديرٌ بالذكر أن كلّ تحويل يوضع في حمولة من حمولات التحويل. يوفّر استخدام عدة حمولات تحويل للبادئ عدة بدائل، والتي يتعيّن على المجيب اختيار أحدها أو رفض العرض كله.

يُحدّد حقل حمولة التحويل (Transform payload) أحد التحويلات الأمنية لاستخدامه في تأمين قناة الاتصالات للبروتوكول المحدّد. ويُستخدم متغيّر رقم التحويل لتحديد تلك الحمولة بحيث يمكن للمجيب استخدامه كإشارة إلى قبوله لذلك التحويل. ويحدّد حقل هوية التحويل (Transform-ID) وحقل الخصائص (Attributes) تحويلاً بعينه (مثلاً، 3DES للتغليف الأمني للحمولة، و-HMAC-SHA-96-1 لترويسة التوثيق) مع الخصائص الخاصة به (مثلاً، طول كود التحوير (الهاش)).

الجدول 6-3 أنواع حمولات ISAKMP.

النوع	المتغيرات	الوصف
الارتباط الأمني (SA)	مجال التفسير (DOI)، الحالة	يُستخدم للتفاوض حول الخواص الأمنية، يحدّد DOI والحالة التي ينعقد التفاوض في ظلها
المقترح (P)	رقم المقترح، هوية البروتوكول، حجم مؤشر المُعامِلات الأمنية، عدد التحويلات، مؤشر المُعامِلات الأمنية.	يُستخدم أثناء التفاوض حول ارتباط أمني؛ يُحدّد البروتوكول المُستخدم وعدد التحويلات
التحويل (T)	رقم التحويل، هوية التحويل، خواص الارتباط الأمني	يُستخدم أثناء التفاوض حول ارتباط أمني، يُحدّد التحويل وخواص الارتباط الأمني ذات الصلة
تبادل المفاتيح (KE)	بيانات تبادل المفاتيح	يدعم عدة أساليب لتبادل المفاتيح
التعريف (ID)	نوع التعريف، بيانات التعريف	يُستخدم لتبادل معلومات التعريف
الشهادة (CERT)	تكويد الشهادة، بيانات الشهادة	يُستخدم لنقل الشهادات والبيانات الأخرى المتعلقة بها
طلب الشهادة (CR)	أرقام أنواع الشهادات، أنواع الشهادات، أرقام جهات إصدار الشهادات، جهات إصدار الشهادات.	يُستخدم لطلب الشهادات؛ يُحدّد أنواع الشهادات المطلوبة والجهات المقبولة لإصدار الشهادات
الهاش (HASH)	بيانات الهاش	يتضمّن بيانات مولّدة من وظيفة الهاش
التوقيع (SIG)	بيانات التوقيع	يتضمّن بيانات مولّدة من وظيفة التوقيع الرقمي
العدد الذي يستخدم لمرة واحدة فقط (NONCE)	بيانات العدد الذي يستخدم لمرة واحدة فقط	يتضمّن عدداً يستخدم لمرة واحدة فقط
الإخطار (N)	DOI، هوية البروتوكول، حجم مؤشر المُعامِلات الأمنية، نوع رسالة الإخطار، مؤشر المُعامِلات الأمنية، بيانات الإخطار	يُستخدم لنقل بيانات الإخطار، كحالة خطأ ما
الحذف (D)	DOI، هوية البروتوكول، حجم مؤشر المُعامِلات الأمنية، عدد مؤشرات المُعامِلات الأمنية، مؤشرات المُعامِلات الأمنية (واحد أو أكثر)	يُحدّد ارتباطاً أمنياً لم يعد صالحاً

يمكن استخدام حمولة تبادل المفاتيح (Key Exchange payload) لمجموعة متنوعة من أساليب تبادل المفاتيح، بما في ذلك أوكلي، وديفي - هيلمان، وتبادل المفاتيح المبني على RSA المُستخدَم من قِبَل PGP. ويتضمّن حقل بيانات تبادل المفاتيح البيانات المطلوبة لتوليد مفتاح الجلسة ويعتمد على الخوارزمية المُستخدَمة لتبادل المفاتيح.

يُستخدم حقل حمولة التعريف (Identification payload) لتحديد هوية النظراء المتواصلين، ويمكن استخدامه أيضاً لتحديد موثوقية المعلومات. وعادةً ما يحتوي حقل بيانات الهوية (ID Data field) على عنوان IPv4 أو IPv6.

تنقل حمولة الشهادة (Certificate payload) شهادة مفتاح عام. ويبيّن حقل ترميز الشهادة (Certificate Encoding field) نوع الشهادة أو المعلومات ذات الصلة بالشهادة، والتي قد تشمل ما يأتي:

- شهادة X.509 المُغلَّفة بـ PKCS #7.
- شهادة PGP.
- مفتاح DNS المُوقَّع.
- شهادة X.509 للتوقيع.
- شهادة X.509 لتبادل مفاتيح.
- علامات كيربيروس (Kerberos tokens).
- قوائم إلغاء الشهادات (CRL).
- قوائم إلغاء الهيئات (ARL).
- شهادة SPKI.

في أي مرحلة من مراحل تبادلات بروتوكول ISAKMP يستطيع المرسل أن يُرفِق حمولة طلب شهادة (Certificate Request) لطلب شهادة الطرف الآخر المُتّصَل به. وقد تسرد الحمولة أكثر من نوع من الشهادات المقبولة، وأيضاً أكثر من جهة مقبولة لإصدار الشهادات.

يحتوي حقل حمولة الهاش (Hash payload) على البيانات المولّدة بواسطة دوال التحوير المطبَّقة على جزء من الرسالة و/أو حالة بروتوكول ISAKMP. وقد تُستخدم هذه الحمولة للتحقق من سلامة البيانات في رسالةٍ ما أو لتوثيق الجهات المتفاوضة.

يحتوي حقل حمولة التوقيع (Signature payload) على البيانات المولدة بواسطة التوقيع الرقمي على جزءٍ من الرسالة و/أو حالة بروتوكول ISAKMP. وتستخدم هذه الحمولة للتحقق من سلامة البيانات في الرسالة، كما يمكن استخدامها أيضاً لخدمات دحض الإنكار.

يحتوي حقل حمولة العدد الذي يستخدم لمرة واحدة فقط (Nonce payload) على بيانات عشوائية تُستخدم لضمان الحيوية (liveness) خلال عملية التبادل وتوفير الحماية ضد هجمات إعادة الإرسال.

يحتوي حقل حمولة الإخطار (Notification payload) على معلومات عن الخطأ أو معلومات عن حالة ذلك الارتباط الأمني أو حالة مفاوضاته. وقد تم تعريف رسائل أخطاء بروتوكول ISAKMP كما يأتي:

نوع الحمولة غير صحيح	لم يتم اختيار أي مقترح
هوية البروتوكول غير صحيحة	بيانات الهاش غير صحيحة
تكويد الشهادة غير صحيح	الإصدار الثانوي غير صحيح
DOI غير مدعوم	تركيب المقترح غير صالح
مؤشر المُعامِلات الأمنية (SPI) غير صحيح	فشل التوثيق
الشهادة غير صحيحة	نوع التبادل غير صحيح
الحالة غير مدعومة	الحمولة مشوَّهة
هوية التحويل غير صحيحة	التوقيع غير صحيح
تركيب طلب الشهادة غير صالح	الأعلام (Flags) غير صحيحة
الكوكي غير صحيح	بيانات المفتاح غير صحيحة
الخواص غير مدعومة	إخطار بالعنوان
جهة إصدار الشهادة غير صحيحة	هوية الرسالة غير صحيحة
الإصدار الرئيسي غير صحيح	

حتى الآن تم تعريف رسالة حالة بروتوكول ISAKMP واحدة فقط وهي "متصلة". وبالإضافة إلى إخطارات بروتوكول ISAKMP تلك، تُستخدم أيضاً إخطارات خاصة بـ DOI. وفيما يتعلق ببروتوكول IPSec فقد تم تعريف الرسائل الإضافية الآتية للحالة:

- فترة العمر للمُجيب (Responder-Lifetime): لبيان فترة العمر للارتباط الأمني الذي تم اختياره من قِبَل المُجيب.
- حالة إعادة الإرسال (Replay-Status): وتُستخدم للتأكيد الإيجابي على رغبة (أو عدم رغبة) المُجيب في تفعيل إمكانية اكتشاف الخدمة المضادة لإعادة الإرسال.
- الاتصال الأوّليّ (Initial-Contact): لإبلاغ الطرف الآخر أن هذا هو أول ارتباط أمني يجري إنشاؤه مع النظام البعيد. ويمكن لمتلقي هذا الإشعار بعد ذلك حذف أي ارتباط أمني لديه للنظام المُرسِل مُفترضاً أن النظام المُرسِل قد تم بدء إعادة تشغيله من جديد، ولذا لم يعد لديه إمكانية الوصول إلى تلك الارتباطات الأمنية.

ويشير حقل حمولة الحذف (Delete payload) إلى الارتباطات الأمنية التي قام المُرسِل بحذفها من قاعدة بياناته، ومن ثَمَّ لم تعد سارية المفعول.

❖ تبادلات بروتوكول ISAKMP:

يوفِّر بروتوكول ISAKMP إطاراً لتبادل الرسائل، حيث تُستخدم أنواع الحمولات كلبنات بناء. وتُحدّد المواصفات خمسة أنواع للتبادلات ينبغي دعمها، والتي يلخِّصها الجدول 6-4. تشير SA في الجدول إلى حمولة ارتباط أمني مع ما يرتبط بها من حمولات البروتوكول وحمولات التحويل.

يسمح التبادل الأساسي (Base Exchange) بنقل مواد تبادل المفاتيح وبيانات التوثيق معاً، مما يقلل من عدد التبادلات على حساب حماية الهوية. وتقوم أول رسالتين بتوفير الكوكيز وإنشاء ارتباط أمني باستخدام البروتوكول والتحويلات

المتفق عليها، ويستخدم كلا الجانبين عدداً يُستعمل لمرة واحدة فقط للحماية ضد هجمات إعادة الإرسال. وتقوم الرسالتان الأخيرتان بتبادل بيانات المفتاح وهوية المُستخدِم بآلية التوثيق المُستخدمة لتوثيق المفاتيح والهويات والأعداد التي تُستعمل لمرة واحدة فقط من أول رسالتين.

يقوم تبادل حماية الهوية (Identity Protection Exchange) بتوسيع التبادل الأساسي لحماية هويات المُستخدِمين، حيث تقوم أول رسالتين بإنشاء الارتباط الأمني. وتقوم الرسالتان التاليتان بعملية تبادل المفاتيح، مع الأعداد التي تُستخدَم لمرة واحدة للحماية ضد هجمات إعادة الإرسال. وبمجرد حساب مفتاح الجلسة، يتبادل الطرفان الرسائل المُشفَّرة التي تحتوي على بيانات التوثيق كالتوقيعات الرقمية وكذلك شهادات توثيق المفاتيح العامة بشكلٍ اختياري.

ويُستخدم تبادل التوثيق فقط (Authentication Only Exchange) للقيام بتوثيقٍ متبادَل بدون تبادل للمفاتيح، حيث تقوم أول رسالتين بإنشاء الارتباط الأمني. بالإضافة إلى ذلك يستخدم المُجيب الرسالة الثانية لبيان هويته مستخدماً التوثيق لحماية الرسالة. ويقوم البادئ بإرسال رسالة ثالثة للتعريف بهويته الموثّقة.

يُقلل التبادل الاعتسافي (Aggressive Exchange) عدد التبادلات من ناحية ولكنه - من ناحية أخرى - لا يوفر حمايةً للهوية. في الرسالة الأولى يقترح البادئ ارتباطاً أمنياً مع البروتوكول المصاحب له بالإضافة إلى خيارات التحويل. كما يقوم أيضاً ببدء عملية تبادل المفاتيح ويرسل هويته. في الرسالة الثانية يشير المُجيب إلى قبوله لذلك الارتباط الأمني مع بروتوكول وتحويل معينين، كما يقوم باستكمال عملية تبادل المفاتيح وتوثيق المعلومات المُرسَلة. وفي الرسالة الثالثة ينقل البادئ نتيجة توثيق تشمل البيانات السابقة مُشفَّرةً بواسطة المفتاح االسري المُشترك للجلسة.

يُستخدم التبادل الإخطاري (Informational Exchange) لنقل المعلومات باتجاه واحد بهدف إدارة الارتباط الأمني.

الجدول 6-4 أنواع تبادلات ISAKMP.

التبادل	ملاحظات
(a) التبادل الأساسي	
(1) **I → R:** SA; NONCE	بدأ التفاوض على ارتباط أمني- بروتوكول ISAKMP
(2) **R → I:** SA; NONCE	الاتفاق على الارتباط الأمني الأساسي
(3) **I → R:** KE; ID_I; AUTH	توليد المفتاح، يتحقق المستجيب من هويّة البادئ
(4) **R → I:** KE; ID_R; AUTH	يتحقق البادئ من هويّة المستجيب، توليد المفاتيح، يتم إنشاء الارتباط الأمني
(b) تبادل حماية الهوية	
(1) **I → R:** SA	بدأ التفاوض على ارتباط أمني- بروتوكول ISAKMP
(2) **R → I:** SA	الاتفاق على الارتباط الأمني الأساسي
(3) **I → R:** KE; NONCE	توليد المفتاح
(4) **R → I:** KE; NONCE	توليد المفتاح
(5)* **I → R:** ID_I; AUTH	يتحقق المستجيب من هويّة البادئ
(6)* **R → I:** ID_R; AUTH	يتحقق البادئ من هويّة المستجيب، يتم إنشاء الارتباط الأمني
(c) تبادل التوثيق فقط	
(1) **I → R:** SA; NONCE	بدأ التفاوض على ارتباط أمني- بروتوكول ISAKMP
(2) **R → I:** SA; NONCE; ID_R; AUTH	الاتفاق على الارتباط الأمني الأساسي، يتحقق المستجيب من هويّة البادئ
(3) **I → R:** ID_I; AUTH	يتحقق البادئ من هويّة المستجيب، يتم إنشاء الارتباط الأمني
(d) تبادل اعتسافي	
(1) **I → R:** SA; KE; NONCE; ID_I	بدأ التفاوض على ارتباط أمني- بروتوكول ISAKMP وتبادل المفاتيح
(2) **R → I:** SA; KE; NONCE; ID_R; AUTH	يتحقق المستجيب من هويّة البادئ، توليد المفتاح، الاتفاق على الارتباط الأمني الأساسي
(3)* **I → R:** AUTH	يتحقق البادئ من هويّة المستجيب، يتم إنشاء الارتباط الأمني
(e) تبادل إخطاري	
(1)* **I → R:** N/D	إخطار بخطأ أو بحالة، أو إلغاء

مفتاح:

I = البادئ

R = المستجيب

* = تمثل تشفير حمولة بعد ترويسة بروتوكول ISAKMP

AUTH = آلية التوثيق المستخدمة

7-6 توصيات للمطالعة

تم تغطية IPv6 وIPv4 بتفصيل أكبر في [STAL04]، ويتضمن [CHEN98] مناقشةً جيدة لتصميم بروتوكول IPSec. أما [FRAN01] و[DORA99] فيقدمان معالجة شاملة لبروتوكول IPSec.

[CHEN98] Cheng, P., et al. "A Security Architecture for the Internet Protocol," *IBM Systems Journal,* Number 1, 1998.

[DORA03] Doraswamy, N., and Harkins, D. *IPSec.* Upper Saddle River, NJ: Prentice Hall, 2003.

[FRAN01] Frankel, S. *Demystifying the IPSec Puzzle.* Boston: Artech House, 2001.

[STAL04] Stallings, W. *Computer Networking with Internet Protocols and Technology.* Upper Saddle River, NJ: Prentice Hall, 2004.

8-6 مصادر للمعلومات على الويب

مشروع المعهد الوطني للمعايير والتكنولوجيا (NIST) الخاص ببروتوكول IPSec: يتضمن أوراقاً بحثية وعروضاً وتنفيذات مرجعية.

9-6 مصطلحات رئيسة

Anti-replay service	خدمة مضادة لإعادة الإرسال
Authentication header (AH)	ترويسة التوثيق
Encapsulating security payload (ESP)	التغليف الأمني للحمولة
Internet Security Association and Key Management Protocol (ISAKMP)	جمعية أمن الإنترنت وبروتوكول إدارةِ المفاتيح
IP Security (IPSec)	بروتوكول أمن الإنترنت
Oakley key determination protocol	بروتوكول أوكلي لتعيين المفاتيح
Replay attack	هجوم إعادة الإرسال
Security Association (SA)	الارتباط الأمني
Transport mode	نمط النقل
Tunnel mode	نمط النفق

10-6 أسئلة للمراجعة ومسائل

1-10-6 أسئلة للمراجعة

1-6 أعط أمثلة لتطبيقات بروتوكول IPSec.

2-6 ما الخدمات التي يقدمها بروتوكول IPSec؟

3-6 ما المتغيرات التي تُعرّف الارتباط الأمني وما المتغيرات التي تصف طبيعة ارتباطٍ أمنيٍّ معين؟

4-6 ما الفرق بين نمط النقل ونمط النفق؟

5-6 ما المقصود بهجوم إعادة الإرسال؟

6-6 لماذا يشتمل التغليف الأمني للحمولة على حقل الحشو؟

7-6 ما الطرق الأساسية لتجميع الارتباطات الأمنية؟

8-6 ما دور بروتوكول أوكلي لتعيين المفاتيح ودور بروتوكول أمن الإنترنت وإدارة المفاتيح ISAKMP في بروتوكول أمن الإنترنت IPSec؟

2-10-6 مسائل

1-6 أثناء مناقشة معالجة ترويسة التوثيق، أشير إلى أن حساب كود التحقق من الرسالة (MAC) لا يشمل كل الحقول في ترويسة التوثيق:

a. لكل واحد من الحقول في ترويسة IPv4، بيِّن ما إذا كان الحقل غيرَ قابل للتغيير، أو قابلاً للتغيير ولكنه تغييرٌ مُتوقع، أو قابلاً للتغيير (يُصفّر قبل حساب ICV).

b. قم بنفس الشيء بالنسبة لترويسة IPv6.

c. قم بنفس الشيء بالنسبة لترويسات IPv6 الملحقة. وفي كل حالة على حدة، برر قرارك لكل حقل.

2-6 عند استخدام نمط النفق، يتم إنشاء ترويسة بروتوكول إنترنت خارجية. لكلٍّ من IPv4 وIPv6، وَضِّح العلاقة بين كل حقل في ترويسة بروتوكول IP الخارجية وكل ترويسة مُلحقة في الرزمة الخارجية مع الحقل المناظر أو الترويسة المُلحقة المناظرة لرزمة بروتوكول IP الداخلية. أي وَضِّح أيّ قيم الترويسة الخارجية يتم استنتاجها من قيم الترويسة الداخلية وأيّ منها يتم تحديده بشكلٍ مستقل عن قيم الترويسة الداخلية.

3-6 مطلوب القيام بالتوثيق والتشفير من طرفٍ إلى طرف بين مضيفين. ارسم أشكالاً مشابهة للشكلين 6-6 و6-9. تُوضح:

a. جوار النقل، مع تطبيق التشفير قبل التوثيق.

b. ارتباطاً أمنياً بنمط النقل مجمّعاً داخل ارتباط أمني بنمط النفق، مع تطبيق التشفير قبل التوثيق.

c. ارتباطاً أمنياً بنمط النقل مجمّعاً داخل ارتباط أمني بنمط النفق، مع تطبيق التوثيق قبل التشفير.

4-6 تنص بنية أمن بروتوكول الإنترنت IPSec على أنه عند تجميع ارتباطين أمنيين بنمط النقل ليتسنى استخدام كلٍّ من بروتوكول ترويسة التوثيق وبروتوكول التغليف الأمني للحمولة من العمل على نفس التدفق من طرف إلى طرف، فإن ترتيباً واحداً فقط للبروتوكولات الأمنية يكون مناسباً: القيام ببروتوكول التغليف الأمني للحمولة قبل بروتوكول ترويسة التوثيق. وضّح أسباب كون ذلك هو النهج الموصى به بدلاً من التوثيق قبل التشفير.

5-6 a. أيّ أنواع تبادل بروتوكول ISAKMP (الجدول 6-4) يُناظر تبادل أوكلي الاعتسافي للمفاتيح (الشكل 6-11) ؟

b. في تبادل أوكلي الاعتسافي للمفاتيح، وضِّح نوع حمولة بروتوكول ISAKMP المناظر لكل متغير بكل رسالة.

الملحق A-6

التشبيك البيني وبروتوكول الإنترنت

يعطي هذا الملحق لمحةً عامّةً عن بروتوكولات الإنترنت، حيث نبدأ بنبذةٍ مختصرةٍ عن دور بروتوكول الإنترنتِ في توفير التشبيك البيني (internetworking) ونتبع ذلك بعرض بروتوكولي الإنترنت الرئيسين IPv4 وIPv6 .

دور بروتوكول الإنترنت

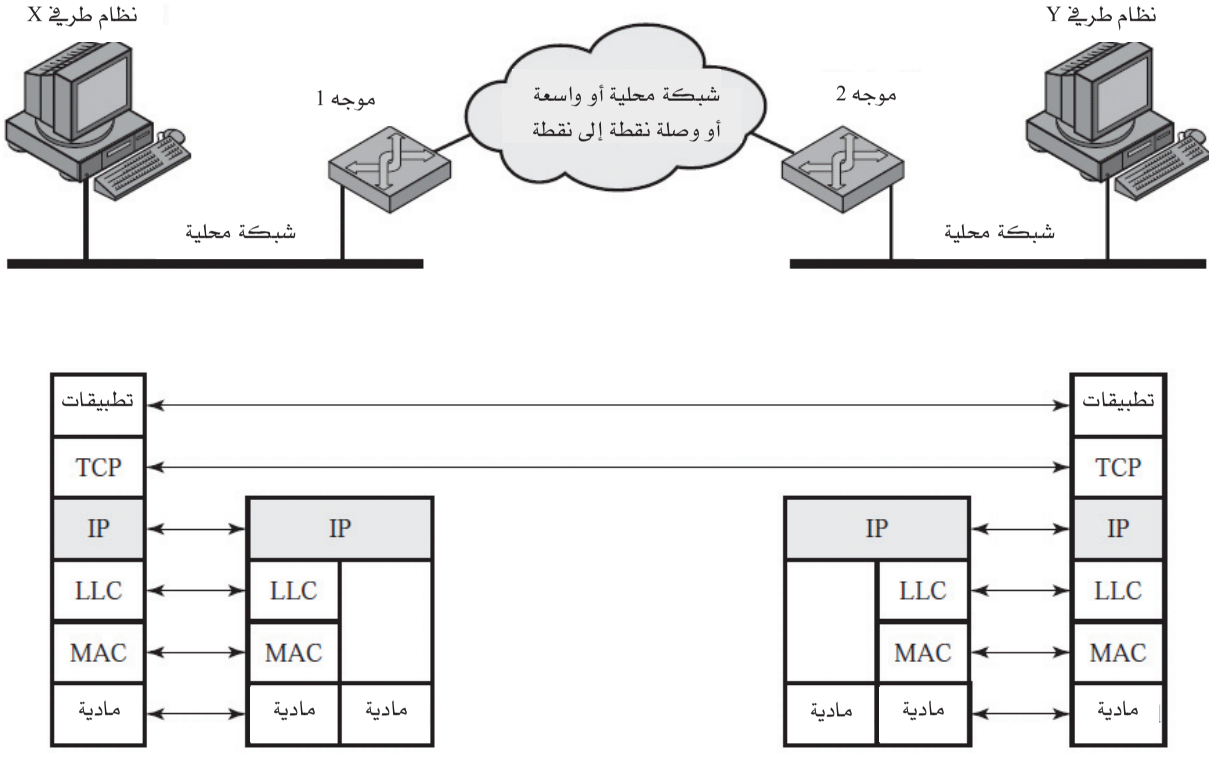
يوفر بروتوكول الإنترنت (IP) الوظائف المطلوبة لربط أنظمةٍ طرفية عبر عدة شبكات. لهذا الغرض فإن IP يُستخدم في كل الأنظمة الطرفيةِ وفي الموجّهات، أي الأجهزة التي تُوفر الاتصالات بين الشبكات. وتُغَلَّف البيانات عالية المستوى في نظام طرفي مصدري في وحدة بيانات بروتوكول IP ((Protocol Data Unit (PDU) تمهيداً لإرسالها. وتُمرّر PDU عبر شبكة أو أكثر من خلال الموجّهات الوسيطة حتى تصل إلى النظام الطرفي المُستهدف.

يجبُ على الموجّه أن يكون قادراً على العمل مع وجود عدة اختلافات بين الشبكات، منها:

- أساليب العنونة: قد تستخدم الشبكات أساليب مختلفةً لتخصيص العناوين للأجهزة على الشبكة. فعلى سبيل المثال؛ تستخدم شبكة اتصالات محلية (LAN) من طراز IEEE 802 عناوين بطول 16 بتاً أو بطول 48 بتاً لكل جهاز متصل بالشبكة. أمّا شبكة تحويل الرزم العامة من طراز X.25 فتستخدم عناوين بطول 12 خانة عشرية (Decimal Digit) بواقع 4 بتّات لكل خانة بمجموع 48 بتاً. وعليه فلابد من توفير أسلوب عالمي لعنونة الشبكات، وكذلك توفير خدمة للدليل (Directory Service).

- الحجم الأقصى للرزم: قد يتطلب الأمر تقسيم الرزم المرسلة من شبكةٍ معينة إلى قطعٍ أصغر لإرسالها إلى شبكةٍ أخرى، وتُعرف هذه العملية بالتجزيء (fragmentation). فعلى سبيل المثال، الحجم الأقصى للرزمة في شبكة إيثرنت هو 15,000 بايت بينما الحجم الشائع في شبكات X.25 هو 1000 بايت فقط. فعلى الموجّه الذي يستلم رزمةً مُرسَلةً من نظام إيثرنت لإرسالها عبر شبكة X.25 تجزئة الرزمة القادمة إلى رزمتين أصغر حجماً.
- الواجهات (Interfaces): تختلفُ وجهات البرمجيات (software) والأجهزة (hardware) للشبكات المختلفة. لذا فإن مفهوم عمل الموجّه ينبغي ألا يعتمد على تلك الاختلافات.
- الاعتمادية (Reliability): بوسع الخدمات المختلفة المتوفرة على الشبكة توفير أيّ شيءٍ بدءاً من دائرة افتراضية موثوقة من طرف إلى طرف (end-to-end) وانتهاءً بخدمة عديمة الموثوقية. لذا فلا ينبغي أن يعتمد الموجّه في عمله على فرضيّة موثوقية الشبكة.

كما يوضّح الشكل 6-13، يعتمد الموجّه في عمله على بروتوكول الإنترنت، في هذا المثال، يقوم بروتوكول الإنترنت (IP) ضمن مجموعة بروتوكولات TCP/IP بتأدية تلك المهمة. ولا بد من تنفيذ بروتوكول IP في جميع الأنظمة الطرفية، وعلى جميع الشبكات، وكذلك في الموجّهات. وعلاوة على ذلك ينبغي أن تكون البروتوكولات الأعلى من بروتوكول IP على كل نظام طرفي متوافقة لنجاح عمليات الاتصال. تحتاج الموجّهات الوسيطة لبروتوكولات حتى IP فقط.



الشكل 6-13: مثال لتهيئة نظام TCP/IP.

خذ في الاعتبار نقل كتلة بيانات من النظام الطرفي X إلى النظامِ الطرفي Y في الشكل 6-13. تستقبل طبقة IP على X كتل البيانات المطلوب إرسالها إلى Y من بروتوكول TCP على X، وتقوم طبقة IP بإلحاق ترويسة تحدّد عنوان الإنترنت العالمي لنظام الوِجهة الطرفي Y. يتكون هذا العنوان من جزأين: معرِّف شبكة ومعرّف نظامٍ طرفي، وسنشير هنا إلى هذه الكتلةِ كرزمة IP. بعد ذلك يُدرك بروتوكول IP أن مضيف الوِجهة Y يوجد على شبكةٍ فرعيةٍ أخرى. ولذا تكون الخطوة الأولى هي إرسال تلك الرزمة إلى موجّه، في هذا المثال الموجّه 1. للقيام بذلك، يُسلِّم IP وحدة بياناته إلى طبقة LLC مرفقةً بمعلومات العنونة الملائمة. وتقوم LLC بتكوين وحدة بيانات LLC تُسلِّمها إلى طبقة الماك أسفلها والتي تقوم بدورها بتكوين رزمة ماك تتضمّن ترويستها عنوان الموجّه 1.

بعد ذلك تنتقل الرزمة على الشبكة المحلية (LAN) إلى الموجّه 1 والذي يقوم بنزع ترويسات الرزمة وترويسة LLC والتذييلات ثم تحليل ترويسة IP لتحديد العنوان النهائيّ للبيانات (في هذه الحالة Y) حيث ينبغي على الموجّه الآن اتخاذ قرارٍ يتعلق بتوجيه مسار البيانات. هناك احتمالان:

1. النظام الطرفي المستهدف Y متصلٌ مباشرةً بإحدى الشبكات الفرعية المتصلة بالموجّه.
2. للوصول للنظام الطرفي المستهدف Y ينبغي المرور عبر موجّه آخر أو أكثر.

في هذا المثال لا بد للرزمة من المرور عبر موجّه 2 لبلوغ غايتها النهائية. وعليه يقوم موجّه 1 بتمرير رزمة IP إلى موجّه 2 عن طريق شبكات وسيطة، وفي هذا يستخدم الموجِّه بروتوكولات تلك الشبكة الوسيطة. فمثلاً، إذا كانت الشبكة الوسيطة من نوع X.25 فإن وحدة بيانات IP تُغلَّف مع معلومات العنونة الملائمة ضمن رزمة X.25 حتى تصل إلى موجّه 2. عندما تصل تلك الرزمة إلى موجّه 2 يقوم الموجّه بنزع ترويسة الرزمة ويكتشف الموجّه أنَّ الوِجهة النهائية

لرزمة IP هي النظام الطرفي Y والموصّل مباشرةً على شبكة فرعية متصلة بالموجّه. ومن ثَمَّ يُنشئ الموجّه رزمةً معنونةً إلى Y ويُرسلها على الشبكة المحلية (LAN). وأخيراً تصل البيانات إلى Y حيث تُزال ترويسات الرزمة وLLC والإنترنت وكذلك التذييلات.

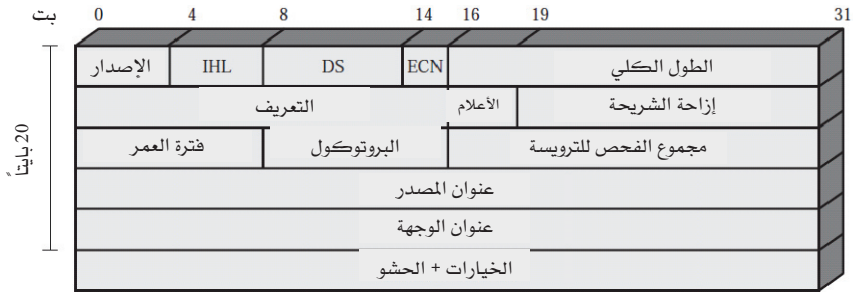
هذه الخدمة المقدّمة من IP غير موثوقة، حيث إن بروتوكول IP لا يضمن وصول البيانات كلّها ولا وصولها بالترتيب السليم إلى وِجهتها. وتناط بالطبقة في المستوى الأعلى (هنا بروتوكول TCP) مسئولية التعافي من أية أخطاء قد تحدُث. ويوفر هذا النهج قدراً كبيراً من المرونة. نظراً لأن تسليم البيانات غير مضمون، فلا توجد أيَّة متطلبات معيّنة للموثوقية على أيٍّ من الشبكات الفرعية. وعليه، فإن البروتوكول يمكنه العمل مع أي توليفة من الأنواع المختلفة من الشبكات الفرعية. وبما أن ضمان وصول البيانات بنفس ترتيب إرسالها غير مطلوب، فإنَّ الرزم المتتالية يمكنها أن تتبع مساراتٍ مختلفة عبر الإنترنت. ويسمح ذلك للبروتوكول بالاستجابة لحالات الازدحام والتعطل في الشبكة وذلك بتغيير المسارات.

الإصدار الرابع لبروتوكول الإنترنت (IPv4)

لعقودٍ مضت، بقى الإصدار الرابع من بروتوكول الإنترنت (IPv4) بمثابة حجر الزاوية لبنية بروتوكول TCP/IP.

يوضّح الشكل 6-14 (a) صيغة ترويسة IP والتي تتكون من 20 بايتاً أو 160 بتاً. وتتضمن الحقول الآتية:

- الإصدار (Version) (4 بتات): ويحدّد رقم الإصدار للسماح بتطور البروتوكول؛ القيمة 4.
- طول ترويسة الإنترنت ((Internet Header Length (IHL)) (4 بتات): طول الترويسة مقاساً بكلمات مؤلَّفة من 32 بتاً، الحد الأدنى 5 كلمات أي 20 بايتاً.



(a) ترويسة IPv4.



(b) ترويسة IPv6.

الشكل 6-14: ترويسات IP.

- DS/ECN (8 بتات): حيث تعني DS الخدمات التفاضلية (Differentiated Services)، وتعني ECN الإخطار الصريح بالازدحام (Explicit Congestion Notification). قبل استحداث الخدمات التفاضلية، كان يُطلق على هذا الحقل "نوع الخدمة" (Type of Service (TOS))

ويُستخدم لتحديد قيم متغيرات الموثوقية والأولوية والتأخير والطاقة الإنتاجية لنقل البيانات. حالياً يقُسّم الحقلُ إلى 6 بتّات للخدمات التفاضلية (DS) وبتّين للإخطار الصريح بالازدحام (ECN).

- الطول الكلي (Total Length) (16 بتاً): طول رزمة IP الكلي بالبايتات.
- المعرِّف (Identification) (16 بتاً): رقم تسلسلي يُكوِّن، مع عنواني المصدر والوِجهة ومع بروتوكول المُستخدِم، تعريفاً حصرياً لرزمةٍ ما. لذا يجب أن يكون هذا الرقم فريداً للرزمة لتمييزها عن أخرى تشاركها نفس عنواني المصدر والوِجهة ونفس بروتوكول المُستخدِم، وذلك طوال فترة بقاء الرزمة في الإنترنت.
- أعلام (Flags) (3 بتات): تم تعريف اثنين فقط من البتات الثلاث. في حالة تجزيء الرزمة، تُبيّن البت الأولى (المزيد More) ما إذا كان هذا الجزء هو الجزء الأخير من الرزمة الأصلية أو لا. أما البت الثانية (لا تجزّيء Don't Fragment) فتمنع التجزيء عندما تكون قيمتها "1". وتُستخدم تلك البت عندما لاتتوافر لدى النظام الطرفي المستهدف إمكانية إعادة تجميع الرزم المجزّأة. حيث يتم في هذه الحالة إهمال تلك الرزمة إذا تجاوز حجمها الحجم الأقصى للرزمة على أي شبكة فرعية على المسار. لذا فعندما تكون تلك البت لها القيمة 1 يستحسن استخدام توجيه مصدري (source routing) لتفادي الشبكات الفرعية التي لها قيم صغيرة للحجم الأقصى للرزمة.
- قيمة الإزاحة لهذا الجزء (Fragment Offset) (13 بتاً): يحدّد موضع هذا الجزء في الرزمة الأصلية، والتي تقاس بوحدة طولها 64 بتاً. وهذا يعني

أن أي جزء من أجزاء الرزمة – عدا الجزء الأخير – لا بد أن يكون طول حقل بياناته من مضاعفات 64 بتاً.

- العُمرُ الافتراضي ((Time to Live (TTL) (8 بتّات): يحدّد أقصى فترة بالثواني يُسمح لرزمة بأن تظل بالإنترنت. ويجب أن يقوم كل موجّه بإنقاص العمر الافتراضي لكل الرزم التي تمر به بواحد على الأقل. لذا فإن العمر الافتراضي يشبه إلى حدٍّ ما عدد القفزات.

- البروتوكول (8 بتات): يحدّد هذا الحقل بروتوكول الطبقة الأعلى مباشرةً التي ستتسلّم حقل البيانات عند الوِجهة، وعليه فإن هذا الحقل يحدّد نوع الترويسة التالية في الرزمة بعد ترويسة IP.

- المجموع التدقيقي للترويسة (Header Checksum) (16 بتاً): يتم تطبيق كود اكتشاف الأخطاء (error-detection code) على الترويسة فقط. ونظراً لأن بعض حقول الترويسة يمكن أن تتغير أثناء النقل (كحقل العمر الافتراضي أو الحقول المتعلقة بعملية تجزيء البيانات (segmentation)) فإن هذه العملية تُكرر ويُتحقق منها عند كل موجّه. قيمة حقل المجموع التدقيقي هي مكمل الواحد (1's complement) للمجموع بطول 16 بتاً لمكمل الواحد لكل كلمة بطول 16 بتاً من كلمات الترويسة. ولأغراض الحساب يتم تصفير حقل المجموع التدقيقي نفسه في البداية.

- عنوان المصدر (Source Address) (32 بتاً): يُكوّد هذا الحقل ليسمح بتخصيص عدد متغير من البتات لتحديد الشبكة والنظام الطرفي الموصّل بها (7 و24 بتاً، أو 14 و16 بتاً، أو 21 و8 بتات).

- عنوان الوِجهة (Destination Address) (32 بتاً): وله نفس خصائص عنوان المصدر.
- الخيارات (Options) (متغيرة): يُكوّد هذا الحقل لتمثيل الخيارات المطلوبة من المُرسِل والتي قد تشمل: وسمة الأمن (security label)، أو توجيه المصدر (source routing)، أو توجيه السِّجِل (record routing)، أو خاتم الوقت (time-stamping).
- الحشو (Padding) (متغير): يُستخدم لجعل طول ترويسة الرزمة من مضاعفات 32 بتاً.

الإصدار السادس لبروتوكول الإنترنت (IPv6)

في عام 1995 أصدر فريق مهام هندسة الإنترنت (IETF) – وهو الفريق الذي يضع معايير بروتوكولات الإنترنت – مواصفات الجيل التالي لبروتوكول الإنترنت، والذي عرف آنذاك بـ IPng. تحولت تلك المواصفات إلى مواصفات قياسية في عام 1996 وعُرِفت باسم IPv6.

يوفّر IPv6 عدداً من التحسينات الوظيفية مقارنةً بـ IPv4، حيث صُمِّم لاستيعاب السرعات العالية المتوفرة للشبكات الحديثة وكذلك الأنواع المختلفة من البيانات المُتدفقة عبر تلك الشبكات، بما في ذلك الصور والصوت والفيديو التي أصبحت شائعة الانتشار. ولكن القوة الدافعة وراء تطوير البروتوكول الجديد كانت الحاجة لمزيد من العناوين. يستخدم IPv4 عناوين بطول 32 بتاً لتحديد المصدر والوِجهة. مع النمو الهائل لشبكة الإنترنت والشبكات الخاصة المتصلة بها، أصبح طول هذا العنوان غير كاف لتعريف جميع الأنظمة التي تحتاج إلى عناوين. ويوضّح الشكل 6-14(b) أن حقل عناوين المصدر والوِجهة في IPv6 يتكون من 128 بتاً. في نهاية المطاف، من المتوقع ترحيل جميع المنشآت

التي تستخدم TCP/IP من بروتوكول الإنترنت الحالي إلى IPv6، ولكن هذه العملية سوف تستغرق عدة سنوات، أو ربما عقود.

❖ ترويسة IPv6

لترويسة IPv6 طولٌ ثابتٌ يبلغ 40 بتاً وتتضمن الحقول الآتية (الشكل 6-14(b)):

- الإصدار (Version) (4 بتات): ويحدّد رقم الإصدار لبروتوكول الإنترنت؛ القيمة 6.
- DS/ECN (8 بتات): حيث DS تعني الخدمات التفاضلية (Differentiated Services)، وتعني ECN الإخطار الصريح بالازدحام (Explicit Congestion Notification). قبل استحداث الخدمات التفاضلية، كان يُطلق على هذا الحقل "فئة المرور" (Traffic Class)، وتُرك محجوزاً للاستخدام من قِبَل الأطراف البادئة و/أو الموجّهات المُمَرِّرة لتحديد الفئات والأوليات المختلفة لرزم IPv6 والتمييز بينها. يشار حالياً لأول 6 بتات من حقل فئة المرور بالخدمات التفاضلية (Differentiated Services). أما البتان المتبقيتان فمحجوزتان لحقل الإخطار الصريح بالازدحام (Explicit Congestion Notification).
- وسمة التدفق (Flow Label) (20 بتاً) يمكن استخدام هذا الحقل من قِبَل المُضيف لبيان تلك الرزم التي تتطلب معاملة خاصة من قِبَل الموجّهات داخل الشبكة، حيث يمكن لوسمة التدفق المساعدة في عملية حجز الموارد والمعالجة الآنية لحركة مرور البيانات.
- طول الحمولة (Payload Length) (16 بتاً) طول الجزء المتبقى من رزمة IPv6 بعد الترويسة بالبايتات. وبعبارة أخرى، هو الطول الإجمالي لجميع

الترويسات الملحقة بالإضافة إلى طول وحدة بيانات بروتوكول النقل (Transport PDU).

- الترويسة التالية (Next Header) (8 بتات) يحدّد نوعية الترويسة التي تلي ترويسة IPv6 مباشرةً ؛ وهذه إما أن تكون ترويسة IPv6 ملحقة أو ترويسة طبقة أعلى، مثل TCP أو UDP.
- حد عدد القفزات (Hop Limit) (8 بتات) العدد المتبقي من القفزات المسموح به لهذه الرزمة. يقوم المصدر بتحديد هذه القيمة. وتقوم كل عقدة تمرّر هذه الرزمة بطرح 1 من قيمة هذا الحقل. تُهمل هذه الرزمة عندما تصل هذه القيمة إلى الصفر.
- عنوان المصدر (Source Address) (128 بتاً): عنوان مُنشئ الرزمة.
- عنوان الوِجهة (Destination Address) (128 بتاً): عنوان المستلِم المقصود لهذه الرزمة. في الواقع قد لا يكون هذا هو عنوان الوِجهة النهائية في حالة وجود ترويسة تمرير كما سنوضّح لاحقاً.

رغم أن ترويسة IPv6 أطول من الجزء الإلزامي لترويسة IPv4 (40 بايتاً مقابل 20 بايتاً)،فإنها تتضمن عدداً أقل من الحقول (8 حقول مقابل 12). ولهذا فإن العمليات التي تقوم بها الموجّهات أقل لكل ترويسة، مما يُسرع من عملية التوجيه.

❖ ترويسات IPv6 الملحقة

تشمل رزمة IPv6 ترويسة IPv6 التي نوقشت أعلاه، كما قد تشمل عدداً من الترويسات الملحقة. تم تعريف الترويسات الملحقة الآتية خارج نطاق IPSec:

- ترويسة خيارات القفزة - بقفزة (Hop-by-Hop Options Header): تُعرِّف الخيارات الخاصة التي تتطلب معالجة بكل قفزة.

- ترويسة التوجيه (Routing Header): توفّر توجيهاً ممتداً يماثل التوجيه المصدري في IPv4.
- ترويسة التجزيء: تتضمّن معلومات التجزيء وإعادة التجميع.
- ترويسة التوثيق: توفر سلامة الرزمة وتوثيقها.
- ترويسة التغليف الأمني للحمولة: توفر الخصوصية.
- ترويسة خيارات الوِجهة: تتضمن معلومات اختيارية تُفحص من قِبَل عقدة الوِجهة.

عند استخدام عدة ترويسات ملحقة، يوصي معيار IPv6 بترتيب ترويسات IPV6 على النحو الآتي:

1. ترويسة IPv6: يتعيّن إلزاميا أن تكون دائماً في المقدمة
2. ترويسة خيارات القفزة – بقفزة
3. ترويسة خيارات الوِجهة: للخيارات التي سيتم معالجتها من قِبَل أول وِجهة والتي تظهر في حقل عنوان IPv6 للوِجهة ـ بالإضافة إلى الوجهات التالية المتضمّنة في ترويسة التوجيه
4. ترويسة التوجيه
5. ترويسة التجزيء
6. ترويسة التوثيق
7. ترويسة التغليف الأمني للحمولة
8. ترويسة خيارات الوِجهة: للخيارات التي ينبغي معالجتها فقط من قِبَل الوِجهة النهائية للرزمة.

يوضّح الشكل 6-15 مثالاً لرزمة IPv6 التي تتضمن أمثلة لكل الترويسات غير الأمنية. لاحظ أن ترويسة IPv6 وكل ترويسة ملحقة تتضمن حقل الترويسة

التالية والذي يعرِّف نوعية الترويسة التالية مباشرةً. إذا كانت الترويسة التالية ترويسةً ملحقة فعندئذٍ يتضمّن ذلك الحقل مُعرِّف النوعية (type identifier) لتلك الترويسة. وإلا فإن ذلك الحقل يتضمّن معرِّف بروتوكول الطبقة الأعلى التي تستخدم IPv6 (عادةً ما يكون بروتوكول نقل)، بنفس القيم المستعملة بحقل البروتوكول في IPv4. في الشكل 6-15، طبقة البروتوكول الأعلى هي TCP، لذا فإن بيانات الطبقة الأعلى التي تحملها رزمة IPv6 تتكون من ترويسة TCP تليها كتلة بيانات التطبيق.



الشكل 6-15: رزمة IPv6 مع ترويسات ملحقة (تحتوي على قطعة TCP).

تحمل ترويسة خيارات القفزة - بقفزة معلومات اختيارية والتي، إن وُجدت، فإنه ينبغي فحصها بكل موجّه على المسار. تتألف الترويسة من الحقول الآتية:

- الترويسة التالية (Next Header) (8 بتات): يعرِّف نوع الترويسة التي تلي هذه الترويسة مباشرةً
- طول امتداد الترويسة (Header Extension Length) (8 بتات): طول هذه الترويسة بوحدات طولها 64 بتاً، ولا يشمل ذلك أول 64 بت.
- الخيارات (Options): تتضمّن خياراً واحداً أو أكثر. يحتوي كل خيار على ثلاثة حقول فرعية هي: مؤشر لبيان نوع الخيار، والطول، والقيمة.

تم حتى الآن تعريف خيار واحد هو خيار الحمولة الفائقة (Jumbo Payload Option) والذي يُستخدم لإرسال رزم IPv6 بحمولات تزيد على 2^{16}-1 65535 =بايتاً. حقل خيار البيانات لهذا الخيار طوله 32 بتاً ويعطي طول الرزمة بالبايت باستثناء ترويسة IPv6.

لمثل تلك الرزم، ينبغي أن يكون حقل طول الحمولة في ترويسة IPv6 صفراً، ويجب أن لا يكون هناك ترويسة تجزيء. مع هذا الخيار، يستطيع IPv6 دعم رزم يصل طولها إلى أكثر من 4 بلايين بايت، مما يسهل نقل رزم الفيديو الكبيرة ويمكّن IPv6 من تحقيق الاستفادة القصوى من سعات الإرسال المتاحة على أي وسائط إرسال.

تتضمّن ترويسة التوجيه قائمة العُقَد الوسيطة (عقدة واحدة أو أكثر) والتي ينبغي أن تُزار خلال المسار إلى وِجهة الرزمة. تبدأ جميع ترويسات التوجيه بكتلة طولها 32 بتاً تحتوي على أربعة حقول طول كلٍّ منها 8 بتات، ويتبع هذه الكتلة بيانات التوجيه المتعلقة بنوعٍ معينٍ من التوجيه. الحقول الأربعة بطول 8 بتات لكلٍّ منها هي: الترويسة التالية، وطول امتداد الترويسة بالإضافة إلى:

- نوع التوجيه (Routing Type): ويُعَرِّف نوعية معيّنة من ترويسات التوجيه. إذا لم يتعرف الموجّه على قيمة نوع التوجيه، فيتعيّن عليه إهمال الرزمة.
- قطع المسار المتبقية (Segments Left): عدد العقد الوسيطة المنصوص عليها صراحةً والتي يجب زيارتها قبل الوصول إلى الوِجهة النهائية.

بالإضافة إلى هذا التعريف العام لترويسة التوجيه، تعرّف مواصفات IPv6 ترويسة التوجيه من النوع الصفري (Type 0). عند استخدام ترويسة التوجيه من النوع الصفري فإن عقدة المصدر لا تضع عنوان الوِجهة النهائية في ترويسة IPv6. وبدلا من ذلك، يوضع هذا العنوان كآخر عنوان مسجل بترويسة التوجيه، أما ترويسة IPv6 فيكون عنوان الوِجهة بها هو أول موجّه مطلوب على المسار. ولا يتم فحص ترويسة التوجيه حتى تصل الرزمة إلى العقدة المعرّفة بترويسة IPv6، وعندها يتم تحديث محتويات ترويسة IPv6 وترويسة التوجيه ثم تمرّر الرزمة. وتشمل عملية التحديث وضع العنوان التالي الذي سوف تتم زيارته في ترويسة IPv6 وإنقاص حقل "قطع المسار المتبقية" في ترويسة التوجيه.

يتطلب بروتوكول IPv6 أن تقوم عقدة IPv6 بعكس المسارات المبينة في رزمة تحتوي على ترويسة توجيه، كي تعيد الرزمة إلى المُرسِل.

تستخدم عقدة المصدر ترويسة التجزيء (Fragment Header) عند الحاجة إلى التجزيء. وفقاً لـ IPv6، يُسمَح للعقد المصدرية فقط القيام بعملية التجزيء، ولا يجوز ذلك للموجّهات الواقعة على مسار انتقال الرزمة. ولتحقيق الاستفادة القصوى من بيئة تربيط الشبكات، لا بد للعقدة من استخدام خوارزمية لاكتشاف المسار (path discovery algorithm) لتحديد أصغر قيمة لوحدات الإرسال القصوى (MTU) التي تدعمها الشبكات الفرعية على المسار. بعبارة أخرى، تحدّد خوارزمية اكتشاف المسار قيمة (MTU) للشبكة الفرعية على المسار والتي تمثل عنق الزجاجة. بناءً على ذلك،تقوم عقدة المصدر بعملية

التجزيء حسب المطلوب لكلَ من عناوين الوِجهة، وإلا فإنه سيتعيّن على المصدر تحديد حجم كل الرزم بـ 1280 بايتاً، وهو الحد الأدنى لوحدة الإرسال القصوى (MTU) التي يجب أن تدعمها أي شبكة فرعية.

بالإضافة إلى حقل الترويسة التالية، تتضمّن ترويسة التجزيء الحقول الآتية:

- قيمة الإزاحة لهذا الجزء (Fragment Offset) (13 بتاً): يحدّد موضع هذا الجزء في الرزمة الأصلية، ويُقاس بوحدات طول كلٍّ منها 64 بتاً. وهذا يعني أن أي جزء من أجزاء الرزمة – عدا الجزء الأخير – لابد أن يكون طول حقل بياناته من مضاعفات 64 بتاً.
- محجوز (Res) (بتان) مخصّص للاستخدام في المستقبل.
- علامة المزيد (More (M) Flag) (بت واحد): 1= مزيد من الأجزاء (Fragments)، 0 = الجزء الأخير.
- المعرِّف (Identification) (32 بتاً): للتعريف الحصري بالرزمة الأصلية. ويجب أن يكون المُعرِّف فريداً لعنوان مصدر الرزمة وعنوان وِجهة الرزمة طوال فترة تواجد الرزمة بالإنترنت. يتم تجميع كل الأجزاء التي تحمل نفس المُعَرِّف ونفس عنوان المصدر ونفس عنوان الوِجهة لإعادة تكوين الرزمة الأصلية.

تحمل ترويسة خيارات الوِجهة (Destinations Options Header) معلومات اختيارية، في حال وجودها يتم فحصها من قِبَل عقدة وِجهة الرزمة فقط. صيغة هذه الترويسة هي نفس صيغة ترويسة خيارات القفزة – بقفزة.

7

الفصل السابع

أمن شبكة الويب

محتويات الفصل:

7-1 متطلبات أمن الويب

7-2 طبقة المقابس الآمنة وأمن طبقة النقل

7-3 المُعامَلات الإلكترونية الآمنة (SET)

7-4 توصيات للمطالعة

7-5 مصادر للمعلومات على الويب

7-6 مصطلحات رئيسة

7-7 أسئلة للمراجعة ومسائل

"استخدم عقلك ...واستيقظ على الحقيقة" ـ من أغنية "استحوذت عليك تحت جلدي" ، لكول بورتر

النقاط الرئيسة

- يوفر بروتوكول طبقة المقابس الآمنة (SSL) خدمات الأمن للتطبيقات التي تستخدم بروتوكول TCP. أما النسخة القياسية الخاصة بالإنترنت فيطلق عليها "أمن طبقة النقل" (TLS). ونظراً لتقاربهما ، فسنشير إليهما على أنهما بروتوكول واحد SSL/TLS إلا في حالة وجود اختلاف.
- يوفر SSL/TLS السرية عن طريق استخدام التشفير المتماثل، ويوفر سلامة الرسائل باستخدام كود توثيق الرسالة (MAC).
- يتضمّن SSL/TLS آليات لتمكين مستخدمي TCP من تحديد طرق الأمن وخدماته التي سيستخدمونها.
- بروتوكول المُعامَلات الإلكترونية الآمنة (SET) هو معيار تشفير وأمن مفتوح صُمِّم لحماية استخدام بطاقات الائتمان على الإنترنت.

توجد مواقع ويب لكل الشركات التجارية تقريباً الآن وكذلك لأغلب الأجهزة الحكومية والعديد من الأفراد. يتزايد عدد الأفراد والشركات المتصلين بالإنترنت بسرعة، ولدى كلٍّ منهم متصفح ويب (Web browser) بواجهة مستخدم رسومية (GUI). مما جعل عدداً كبيراً من الشركات تتحمس لوضع الوسائل اللازمة على شبكة الويب لتسهيل التجارة الإلكترونية عبر الإنترنت. لكن تبقى الحقيقة الماثلة أن الإنترنت والويب يتعرضان بشكلٍ قوي لتشكيلة كبيرة من الهجمات.

ومع تنبه الشركات لهذه الحقيقة، ازداد الطلب على توفير خدمات ويب آمنة. وهذا الموضوع – أمن الويب – موضوع واسع، ويمكن أن يُشكِّل بسهولة كتاباً بكامله (للمزيد يمكنك الاطلاع على الكتب الموصَّى بها في نهاية هذا الفصل).

سنبدأ هذا الفصل بمناقشة المتطلبات العامة لأمن الويب، وبعد ذلك سنركز على معيارين ازدادت أهميتهما بشكلٍ كبير كجزء مهم من بنية نظم التجارة عبر الإنترنت: SSL/TLS، وSET.

1-7 متطلبات أمن الويب

إن شبكة المعلومات العالمية (والمعروفة بالويب) ليست في جوهرها إلا تطبيق "زبون/خادم" يعمل على شبكات الإنترنت والإنترانت (أي شبكات TCP/IP الداخلية). ولذا، فإن أدوات الأمن وأساليبه المختلفة التي ناقشناها حتى الآن في هذا الكتاب تصلح لتأمين الويب. لكن، كما أشار [GARF 97]، هناك مجموعة من التحديات الجديدة التي يقدمها الويب ولم تعالج بشكلٍ عام في سياق أمن الشبكة والحاسب؛ منها:

- يُعدُّ الويب ــ على خلاف بيئات النشر التقليدية أو حتى أنظمة النشر الإلكترونية الأخرى (كقنوات المعلومات النصية، وأنظمة الردّ الصوتي، ونظام إرسال الفاكسات واستقبالها عن طريق الحاسب) ــ عرضةً للكثير من الهجمات على خادمات الويب من خلال الإنترنت.
- يعمل الويب على نحو متزايد كواجهة مرئية للشركة ومنتجاتها، وأيضاً كمنصة لإجراء مُعامَلات تجارية. لكن يمكن أن تدمّر السمعة وتضيع الأموال إذا تعرضت خادمات الويب للتخريب.
- رغم سهولة استخدام متصفحات الويب وإعداد خادمات الويب وإدارتها وتطوير محتوى الويب، فإن برمجيات البنية التحتية معقدةٌ جداً. وقد يؤدى هذا التعقيد في البرمجيات إلى إخفاء كثير من النقاط المحتمل ضعفها أمنياً. إن تاريخ الويب القصير مليء بالأمثلة لأنظمة جديدة وأنظمة مُرقَّاة تم إعدادها بشكلٍ شرعي لكنها تتضمن نقاطاً عرضةً لكثير من الهجمات.
- يمكن أن يُستغَل خادم الويب كقاعدة لشن هجوم على جميع أجهزة الحاسب بالشركة أو المنظمة. عندما يتعرض خادم الويب للتخريب، يمكن

أن يصبح المهاجم قادراً على الدخول إلى أنظمة البيانات والأنظمة التي ليست جزءاً من الويب نفسه لكنها مرتبطة بالموقع المحلي للخادم.

- أكثر مستخدمي خدمات الويب أشخاصٌ عاديون وغير مدرَّبين على القضايا الأمنية، وهؤلاء لا يُدركون بالضرورة الأخطار الأمنية المُحدقة، كما أنهم لا يمتلكون الأدوات أو المعرفة لمواجهة تلك الأخطار بشكلٍ فعالٍ.

1-1-7 تهديدات أمن الويب

يتضمن الجدول 7-1 ملخصاً بالتهديدات الأمنية المختلفة التي يمكن التعرض لها عند استخدام الويب. يمكن تصنيف تلك التهديدات من حيث كونها هجمات سلبية أو هجمات نشطة. وتتضمّن الهجمات السلبية التنصت على حركة مرور البيانات على الشبكة بين المتصفح والخادم، والتمكّن من الوصول لمعلومات موجودة على موقع الويب يفترض أنها مقيّدة الاستخدام. أما الهجمات النشطة فتتضمّن تقمص شخصية مستخدم آخر، وتعديل الرسائل أثناء انتقالها بين الزبون والخادم، وتعديل المعلومات على موقع الويب.

من الطرق الأخرى لتصنيف تهديدات أمن الويب تصنيفها حسب موقع مصدر التهديد: هل هو خادم الويب، أم متصفح الويب، أم حركة مرور بيانات الشبكة بين المتصفح والخادم؟. تقع قضايا أمن الخادم والمتصفح ضمن أمن نظام الحاسب التي يناقشها بشكلٍ عام الباب الثالث من هذا الكتاب. أما قضايا أمن حركة مرور البيانات فتُصنَّف ضمن أمن الشبكة وسنناقشها في هذا الفصل.

2-1-7 طرق تأمين حركة مرور بيانات الويب

يمكن استخدام عدة طرق لتوفير أمن الويب. تتشابه الطرق المختلفة التي اقتُرحَت بدرجة كبيرة في الخدمات التي توفرها وإلى حدٍّ ما في الآليات التي

تستخدمها، لكنها تختلف من حيث نطاق التطبيق وموقعها ضمن رصَّة بروتوكولات TCP/IP.

يبيّن الشكل 7-1 هذا الاختلاف. يمكن توفير أمن الويب من خلال تأمين بروتوكول IP باستخدام IPSec (الشكل 7-1 (a)). وتمتاز هذه الطريقة بأنها تتم دون الحاجة إلى تدخلٍ من مستخدمي الأنظمة الطرفية على الشبكة أو تطبيقاتها، كما أنها توفر حلاً عاماً. يتضمّن IPSec أيضاً إمكانيةً للترشيح (filtering) كي تقتصر المعاناة من عبء المعالجة الإضافي على نوعية محددة فقط من حركة مرور البيانات.

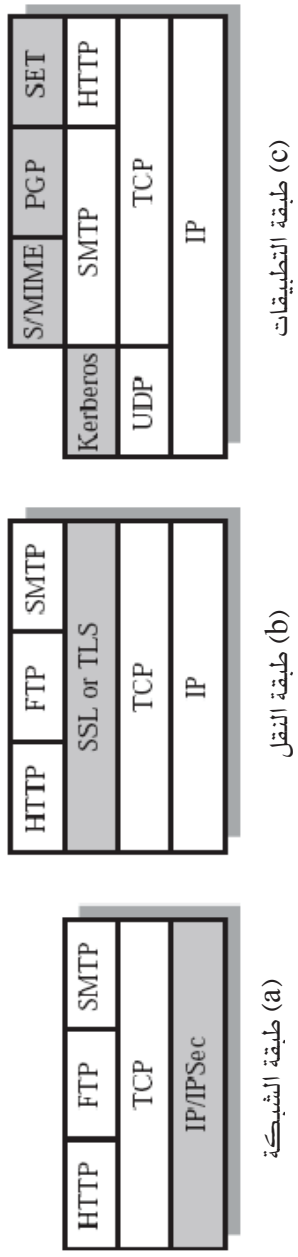
هناك حلٌّ آخرٌ عامٌ نسبياً وهو وضع إجراءات الأمن فوق TCP مباشرةً (انظر الشكل 7-1 (b)). وإن أفضل مثال لهذه الطريقة هو بروتوكول طبقة المقابس الآمنة (Secure Socket Layer (SSL)) ومعيار الإنترنت الذي تلاه والمعروف باسم أمن طبقة النقل (Transport Layer Security (TLS)). ويمكن تحقيق خدمات الأمن في هذا المستوى بأحد أسلوبين: توفير بروتوكول SSL (أو TLS) كجزء من رصَّة البروتوكولات التحتية وبالتالي لا تشعر التطبيقات بوجوده، البديل الآخر هو تضمين SSL في حزم برمجيات معينة.

على سبيل المثال، يأتي متصفح نيتسكيب ومتصفح مستكشف مايكروسوفت مزوداً ببروتوكول SSL، وكذلك تتضمَّن أكثر خادمات الويب ذلك البروتوكول. لكن خدمات الأمن الخاصة بتطبيق معين تكون متضمَّنة مع ذلك التطبيق نفسه. ويبيّن الشكل 7-1 (c) أمثلة لتلك البنية المعمارية. وتكمن ميزة هذه الطريقة في القدرة على تفصيل الخدمة لتلبي احتياجات معيّنة للتطبيق. ومن الأمثلة المهمة لتلك الطريقة - ضمن سياق أمن الويب - بروتوكول المُعامَلات الإلكترونية الآمنة (SET).

سنكرس بقية هذا الفصل لمناقشة SSL/TLS وSET.

الجدول 7-1: مقارنة التهديدات الأمنية على الويب [RUBI97].

	التهديدات	العواقب	الأساليب المضادة
سلامة البيانات	- تعديل بيانات المستخدم - متصفح حصان الطروادة - تعديل محتويات الذاكرة - تعديل حركة مرور الرسائل العابرة	- فقد معلومات - اختراق الجهاز - التعرض لجميع التهديدات الأخرى	المجموع التدقيقي المُشفَّر
السرية	- التجسس على الشبكة - سرقة معلومات من الخادم - سرقة بيانات من الزبون - تسرب معلومات إعدادات الشبكة - تسرب معلومات عن الزبائن الذين يتصلون بالخادم	- فقد معلومات - فقد الخصوصية	التشفير، خادمات وكيل الويب
حجب الخدمة	- إنهاء عمليات المستخدم قبل الأوان - غمر الجهاز بالطلبات المزيَّفة - ملء الذاكرة أو القرص الصلب (إهدارهما فيما لا طائل منه) - عزل الجهاز بمهاجمة DNS	- عرقلة الخدمة - مضايقة المستخدمين - منع المستخدم من أداء عمله (ومن ثَمَّ تؤثر على الإنتاجية)	من الصعب منعها
التوثيق	- تقمص شخصيات أخرى - تزييف البيانات	- إساءة تمثيل المستخدم - اعتقاد أن المعلومات المزيفة صحيحة	أساليب التشفير



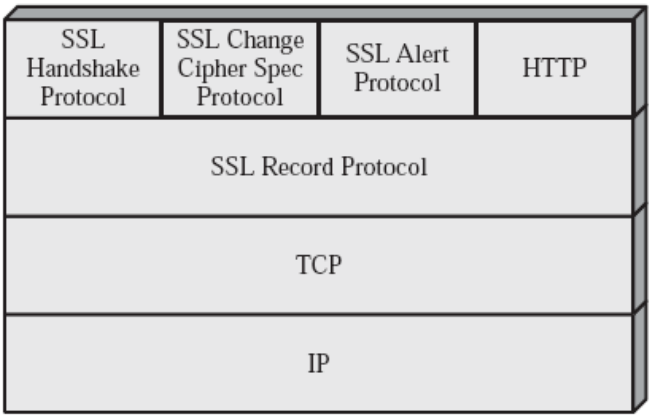
الشكل 7-1: مواقع وسائل الأمن ضمن رصَّة بروتوكولات TCP/IP.

2-7 طبقة المقابس الآمنة وأمن طبقة النقل

قامت شركة نيتسكيب بتطوير بروتوكول SSL. صُمِّمت النسخة 3 من هذا البروتوكول بعد مراجعة من مجتمع الإنترنت وملاحظات الصناعة ونُشِرت كمسوّدة إنترنت أولية. بعد ذلك، وعندما تم التوصل إلى إجماع لإصدار معيار موحّد للإنترنت، شُكِّلت مجموعة عمل خاصة ضمن فريق عمل هندسة الإنترنت (IETF) لتطوير معيار موحّد. أُطلق على هذه النسخة TLS، ويمكن اعتبارها في جوهرها النسخة 3.1 من بروتوكول SSL (SSLv3.1) وهي قريبة جداً ومتوافقة مع SSLv3. سنكرِّس معظم هذا الجزء لمناقشة SSLv3. وفي نهاية الجزء سنوضح الاختلافات الرئيسة بين SSLv3 وTLS.

1-2-7 بنية SSL

صُمِّم SSL بحيث يستخدم TCP لتقديم خدمة آمنة موثوقة من طرف إلى طرف. جديرٌ بالذكر أن SSL ليس بروتوكولاً واحداً وإنما يتألف في الواقع من طبقتين من البروتوكولات كما هو مبين في الشكل 7-2. يوفر بروتوكول سجل SSL (SSL Record Protocol) خدمات الأمن الأساسية لبروتوكولات الطبقات الأعلى المختلفة. وبشكلٍ خاص، يعمل بروتوكول HTTP ــ الذي يوفر خدمة النقل بين زبون وخادم الويب ــ فوق بروتوكول SSL. توجد ثلاثة بروتوكولات للطبقات العليا مُعرَّفة كجزءٍ من SSL: بروتوكول المصافحة (Handshake Protocol)، وبروتوكول تغيير مواصفات التشفير (Change Cipher Spec Protocol)، وبروتوكول التنبيه (Alert Protocol). وتُستخدَم تلك البروتوكولات الثلاث في إدارة تبادلات SSL (SSL exchanges)، وسنناقشها لاحقاً في هذا الجزء.



الشكل 7-2: رصّة بروتوكولات SSL.

من مفاهيم SSL المهمة: مفهوم توصيلة SSL ومفهوم جلسة SSL؛ وهما مُعرَّفان في مواصفات البروتوكول كالتّالي:

- التوصيلة (Connection): هي وسيلة نقل (في مصطلحات نموذج OSI) توفر نوعاً مناسباً من الخدمة. تمثل توصيلات SSL علاقات نظير لنظير، وهذه التوصيلات مؤقتة. حيث ترتبط كل توصيلة بجلسة واحدة.
- الجلسة (Session): تمثل جلسة SSL ارتباطاً بين زبون وخادم. ويتم إنشاء الجلسات عن طريق بروتوكول المصافحة. وتُعرِّف الجلسات مجموعة مُعامِلات الأمن الخاصة بالتشفير التي يمكن أن تشترك فيها عدة توصيلات. تُستخدم الجلسات لتفادي التفاوضات المكلِّفة حول مُعامِلات أمن جديدة لكل توصيلة.

قد يوجد بين كل زوج من التطبيقات (مثلاً HTTP على الزبون والخادم) عدة توصيلات آمنة. نظرياً، قد يكون هناك أيضاً عدة جلسات بين تلك الأطراف في نفس الوقت، ولكن هذه الخاصية لا تُستخدم عملياً. ويرتبط بكل جلسة في الواقع عددٌ من أوضاع (حالات) التشغيل. وبمجرد إنشاء الجلسة، تكون في حالة تشغيل

حالية تسمح بالقراءة والكتابة (أي الإرسال والاستقبال). كما أنه أثناء عملية المصافحة، يتم إنشاء حالات معلَّقة للقراءة والكتابة. وعند الانتهاء من بروتوكول المصافحة بنجاح، تصبح الحالات المعلَّقة حالات حالية. وتُعرَّف حالة الجلسة بالمُعامِلات الآتية (التعريفات مقتبسة من مواصفات SSL):

- مُعرِّف الجلسة: سلسلة من البايتات الاعتباطية يختارها الخادم لتعريف حالة جلسة نشطة أو قابلة للاستئناف.
- شهادة النظير: وهي شهادة X509.v3. ويمكن أن يأخذ القيمة null (لا يوجد).
- أسلوب ضغط البيانات: يحدد الخوارزمية المستخدمة لضغط البيانات قبل التشفير.
- مواصفات التشفير: يحدد خوارزمية تشفير كتل البيانات (مثلاً AES أو لا يوجد) وخوارزمية التحوير (مثلاً MD5 أو SHA-1) المستخدمة لحساب كود توثيق الرسالة (الماك). كما أنه يُعرِّف أيضاً خواص التشفير مثل طول كود التحوير.
- السرّ الرئيس: وهو سرٌ مشترك بين الزبون والخادم يتكون من 48 بايتاً.
- عَلَم يشير إلى مدى إمكانية استخدام الجلسة لبدء توصيلات جديدة.

تُعرَّف حالة التوصيلة بالمُعامِلات الآتية:

- رقم عشوائي لكلٍّ من الخادم والزبون: وهو سلسلة بايتات يختارها كلٌّ من الخادم والزبون لكل توصيلة.
- سرّ كود توثيق الرسالة للكتابة لدى الخادم: المفتاح السرّي المستخدم في عمليات الماك على البيانات التي يرسلها الخادم.
- سرّ كود توثيق الرسالة للكتابة لدى الزبون: المفتاح السرّي المستخدم في عمليات الماك على البيانات التي يرسلها الزبون.

- مفتاح الخادم للكتابة: مفتاح التشفير التقليدي للبيانات التي يشفّرها الخادم ويقوم الزبون بإزالة تشفيرها.
- مفتاح الزبون للكتابة: مفتاح التشفير التقليدي للبيانات التي يشفّرها الزبون ويقوم الخادم بإزالة تشفيرها.
- متجهات التهيئة (IV): عند استخدام تشفير كتلة في نمط CBC، يتم الاحتفاظ بمتجه تهيئة (IV) لكل مفتاح. يأخذ هذا الحقل قيمته الأولية من قِبَل بروتوكول المصافحة، ومن ثم يتم الاحتفاظ بكتلة الشفرة النهائية لكل سجل لاستخدامها كمتجه تهيئة للسجل الذي يليه.
- أرقام التسلسل: يحتفظ كل طرف بأرقام تسلسل مستقلة للرسائل المرسلة والمستلمة لكل توصيلة. عندما يرسل طرف أو يتلقى رسالة تغيير مواصفات التشفير (change cipher spec)، يقوم بتصفير رقم التسلسل المعني. يجب ألا تتجاوز قيمة رقم التسلسل $2^{64}-1$.

7-2-2 بروتوكول سجل SSL

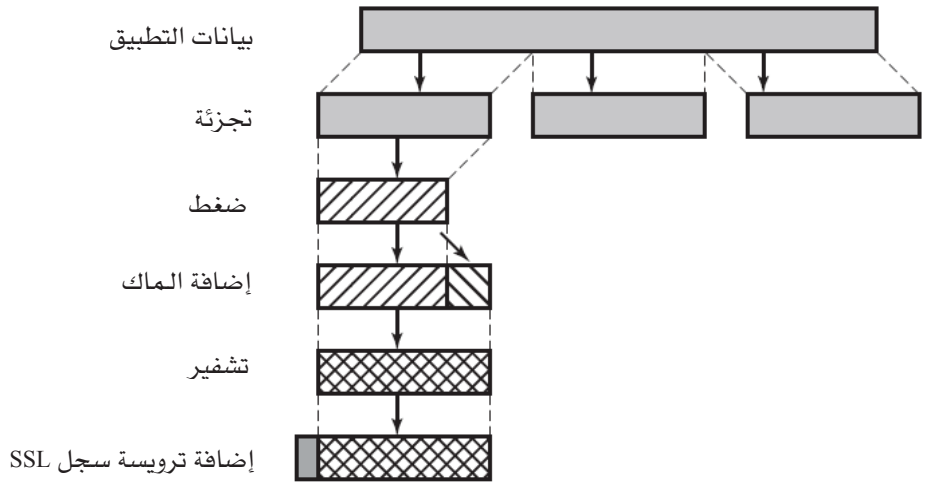
يوفر بروتوكول سجل SSL خدمتين لتوصيلات SSL:

- السرّية: يعرّف بروتوكول المصافحة مفتاحاً سرياً مشتركاً يُستخدَم للتشفير التقليدي للحمل الآجر لبروتوكول SSL.
- سلامة الرسالة: يُعرِّف بروتوكول المصافحة أيضاً مفتاحاً سرّياً مشتركاً يُستخدَم لتكوين كود توثيق الرسالة (الماك).

يبيّن الشكل 7-3 طريقة عمل بروتوكول سجل SSL. يأخذ البروتوكول رسالة التطبيق المراد إرسالها ويجزئ البيانات إلى كتلٍ يسهل التعامل معها، وقد يضغط البيانات (وهي عملية اختيارية)، ثم يطبّق الماك. يقوم بعد ذلك بتشفير الناتج، وإضافة ترويسة، ثم إرسال وحدة البيانات الناتجة في قطعة TCP. يتم إزالة تشفير

البيانات المُستلَمة وتوثيقها، وإزالة الضغط (إن وجد)، ثم يعاد تجميع الأجزاء وتسليمها للمستخدمين في الطبقات العليا.

إن أول خطوة هي **التجزئة**. تُجزأ كل رسالة واردة من الطبقة العليا إلى كتل مؤلَّفة من 2^{14} بايتاً (أي 16384 بايتاً) أو أقل. بعد ذلك تُضغط الكتل الناتجة عند الرغبة في ذلك؛ ويجب أن يكون الضغط بدون فقد (lossless) وألا يزيد طول المحتوى عن 1024 بايتاً. واضح أن الهدف من الضغط هو تقليص حجم البيانات. ومع ذلك، عندما تكون الكتل قصيرة جداً فقد تؤدي خوارزمية الضغط إلى زيادة الطول عن الطول الأصلي؛ بسبب مقتضيات الصياغة.



الشكل 7-3: طريقة عمل بروتوكول سجل SSL.

في بروتوكول SSLv3 (وأيضاً في النسخة الحالية من TLS)، لم يتم تحديد خوارزمية ضغط ومن ثمَّ تكون القيمة الاعتيادية لخوارزمية الضغط "لا يوجد" (null).

الخطوة التالية هي حساب كود توثيق الرسالة (MAC) للرسالة من البيانات المضغوطة. لهذا الغرض يُستخدَم مفتاح سرّي مشترك؛ وتتم الحسابات كما يأتي:

```
Hash(MAC_write_secret || pad_2 ||
        Hash(MAC_write_secret || pad_1 || seq_num ||
        SSLCompressed.type ||
        SSLCompressed.length || SSLCompressed.fragment))|
```

حيث:

|| = عملية الوصل (الإلحاق).

MAC_write_secret = مفتاح سري مشترك.

Hash = خوارزمية التحوير MD5 أو SHA-1.

pad_1 = القيمة 0x36 (أي 0110 0011) مكررة 48 مرة (أي بطول كلي 384 بتاً) في حالة MD5 ومكررة 40 مرة (أي بطول كلي 320 بتاً) في حالة SHA-1.

pad_2 = القيمة 0x5C (أي 1100 0101) مكررة 48 مرة في حالة MD5 ومكررة 40 مرة في حالة SHA-1.

seq_num = الرقم التسلسلي لتلك الرسالة.

SSLCompressed.type = بروتوكول الطبقة العليا المستخدم لمعالجة هذا الجزء (segment).

SSLCompressed.length = طول الجزء بعد الضغط.

SSLCompressed.fragment = الجزء المضغوط (في حالة عدم استخدام الضغط يوضع جزء النص الأصلي).

لاحظ أن هذه العملية تشبه بشكلٍ كبير جداً خوارزمية HMAC التي عرَّفناها في الفصل الثالث. الاختلاف هو أنه يتم وصل pad_1 مع pad_2 في SSLv3، لكن في HMAC يتم حساب العملية المنطقية XOR عليهما. وتعتمد خوارزمية ماك لبروتوكول SSLv3 على مسوّدة الإنترنت الأصلية لـ HMAC التي تستخدم عملية الوصل. أما النسخة النهائية لـ HMAC، والمُعرَّفة في RFC 2104 فتستخدم XOR. بعد ذلك تُشفَّر الرسالة المضغوطة مضافاً إليها الماك بالتشفير المتماثل. ولا يزيد التشفير طول المحتوى بأكثر من 1024 بايتاً، ومن ثمَّ لا يزيد الطول الكلي عن 2048+2^{14}.

يمكن استخدام أيٍّ من خوارزميات التشفير الآتية:

تشفير التدفق		تشفير الكتل	
الخوارزمية	طول المفتاح	الخوارزمية	طول المفتاح
RC4-40	40	AES	256 ،128
RC4-128	128	IDEA	128
		RC2-40	40
		DES-40	40
		DES	56
		3DES	168
		Fortezza	80

يمكن استخدام طريقة فورتيزا (Fortezza) في نظامٍ لتشفير البطاقات الذكية. في تشفير التدفق، تُشفَّر الرسالة المضغوطة مضافاً إليها الماك. لاحظ أن الماك يُحسَب قبل القيام بالتشفير وأن الماك يُشفَّر سويةً مع النص الأصلي أو النص الأصلي بعد الضغط.

في تشفير الكتلة، قد تُضاف بعد الماك وقبل التشفير حشوة (padding). تتكون الحشوة من سلسلة من البايتات يليها بايت يبيّن طول الحشوة. الطول الكلي للحشوة هو أقصر طول يجعل الطول الكلي للبيانات المراد تشفيرها (النص الأصلي + الماك + الحشوة) من مضاعفات طول كتلة الشفرة. وكمثال على ذلك،

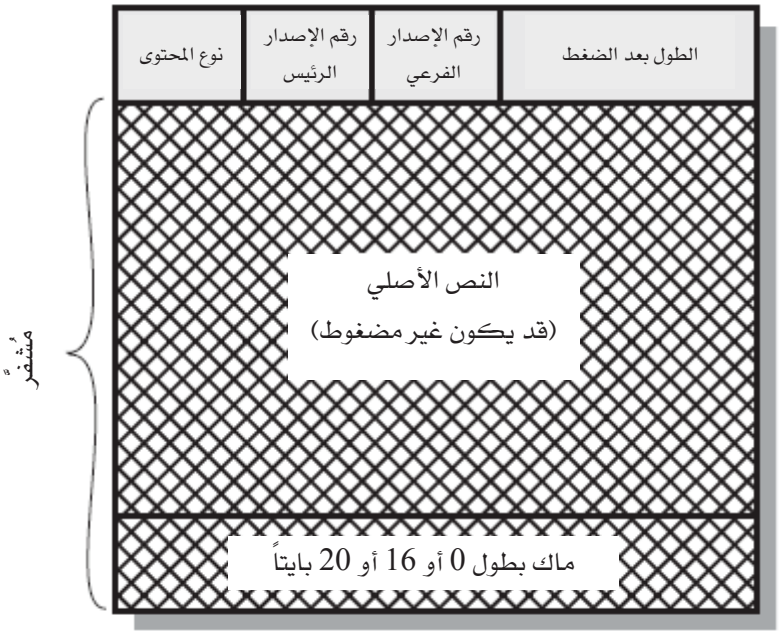
خذ في الاعتبار نصاً أصلياً (أو نصاً مضغوطاً إذا كان الضغط مستخدماً) طوله 58 بايتاً، وطول الماك 20 بايتاً (باستخدام SHA-1)، ويتم تشفيره باستخدام كتلة بطول 8 بايتات (مثلاً باستخدام DES). مع بايت طول الحشوة (padding.length byte) يكون المجموع 79 بايتاً. ولجعل المجموع الكلي من مضاعفات العدد 8، تضاف حشوة من بايت واحد.

تتضمن خطوة المعالجة النهائية في بروتوكول سجل SSL إضافة ترويسة تشمل الحقول الآتية:

- نوع المحتوى (8 بتات): بروتوكول الطبقة العليا المُستخدَم لمعالجة جزء الرزمة (fragment) المتضمَّن.
- الإصدار الرئيس (8 بتات): يحدد إصدار SSL الرئيس المُستخدَم. في حالة SSLv3 تكون القيمة 3.
- الإصدار الثانوي (8 بتات). يحدد إصدار SSL الثانوي المُستخدَم. في حالة SSLv3 تكون القيمة 0.
- الطول بعد الضغط (16 بتاً): طول جزء النص الأصلي بالبايتات (أو طول الجزء المضغوط إذا كان الضغط مستخدماً). القيمة القصوى هي $2^{14}+2048$.

أنواع المحتوى المُعرَّفة هي: change_cipher_spec (تغيير مواصفات الشفرة)، وalert (تنبيه)، وhandshake (مصافحة)، وapplication_data (بيانات تطبيق). الأنواع الثلاثة الأولى هى بروتوكولات خاصة بـ SSL، وسنناقشها فيما بعد. لاحظ عدم وجود تمييز بين التطبيقات المختلفة التي قد تستخدم SSL (كـ HTTP مثلاً)؛ فمحتوى البيانات من مثل هذه التطبيقات يبقى غير مميّز لـ SSL.

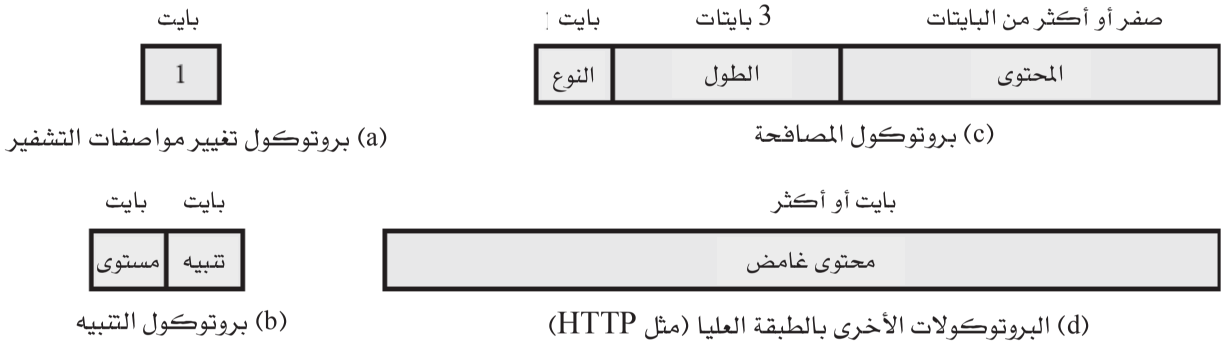
يبيّن الشكل 7-4 صيغة بروتوكول سجل SSL.



الشكل 7-4: صيغة سجل SSL.

3-2-7 بروتوكول تغيير مواصفات الشفرة

بروتوكول تغيير مواصفات الشفرة (change cipher spec protocol) هو أحد البروتوكولات الثلاثة للبرتوكولات المحددة بـ SSL التي تستخدم بروتوكول سجل SSL؛ وهو أبسطها. ويشمل هذا البروتوكول رسالة واحدة (الشكل 7-5 (a)) تتكون من بايت واحد قيمته 1. الغرض الوحيد من هذه الرسالة هو التسبب في نسخ الحالة المُعلَّقة (pending state) إلى الحالة الحالية (current state)، والتي تُحدِّث مجموعة برمجيات التشفير التي ستستخدم على هذه التوصيلة.



الشكل 7-5: حمولة بروتوكول سجل SSL.

4-2-7 بروتوكول التنبيه

بروتوكول التنبيه هو البروتوكول المسؤول عن رسائل التنبيه في SSL للكيان النظير. كما هو الحال مع التطبيقات الأخرى التي تستخدم SSL، تُضغط الرسائل وتُشفَّر كما هو محدد بالحالة الحالية.

تتكون كل رسالة في هذا البروتوكول من بايتين (الشكل 7-5 (b)). يأخذ البايت الأول القيمة 1 (تعني تحذير) أو 2 (تعني فادح) لبيان مستوى خطورة الرسالة. وإذا كان مستوى الخطورة "فادح"، فسيُغلق SSL تلك التوصيلة فوراً. بينما قد تستمر التوصيلات الأخرى ضمن نفس الجلسة، لن يتم إنشاء توصيلات جديدة ضمن تلك الجلسة. ويتضمّن البايت الثاني كوداً يبيّن رسالة التنبيه المحددة. وسنسرد هنا أولاً تلك التنبيهات التي تُعدُّ دائماً فادحة (التعريفات مُقتبسة من مواصفات SSL):

- unexpected_message (رسالة غير متوقعة): للإشارة إلى استلام رسالة غير متوقعة.
- bad_record_mac (ماك سجل خطأ): للإشارة إلى استلام ماك خطأ.
- decompression_failure (فشل إزالة الضغط): للإشارة إلى أن المُدخَل لوظيفة إزالة الضغط غير صحيح (مثلاً عدم القدرة على إزالة الضغط، أو إزالة الضغط لأكثر من الطول المسموح به).
- handshake_failure (فشل المصافحة): للإشارة إلى عدم قدرة المُرسِل على التوصل إلى مجموعة مقبولة من مُعامِلات الأمن في ظل الخيارات المتاحة للتفاوض.
- illegal_parameter (معامِل غير صحيح): للإشارة إلى وجود حقل في رسالة المصافحة قيمته خارج المدى أو متناقضة مع الحقول الأخرى.

أما بقيّة التنبيهات فهي على النحو الآتي:

- close_notify (إشعار غلق): لإشعار المُستلِم بأن المُرسِل لن يرسل رسائل أخرى على هذه التوصيلة. على كل طرف إرسال تنبيه close_notify قبل إغلاق جهة الكتابة للتوصيلة.
- no_certificate (لا توجد شهادة): قد يُرسَل هذا التنبيه رداً على طلب شهادة عند عدم توفر شهادة ملائمة.
- bad_certificate (شهادة غير صالحة): في حالة وجود خلل في الشهادة المُستلَمة (مثلاً قد تحتوي على توقيع لا يمكن تحقيقه).
- unsupported_certificate (شهادة غير مدعومة): نوع الشهادة المُستلَمة غير مدعوم.
- certificate_revoked (شهادة ملغاة): تم إلغاء الشهادة من قِبَل مُوقِّعها.
- certificate_expired (شهادة منتهية الصلاحية).
- certificate_unknown (شهادة غير معروفة): ظهرت مشكلة أخرى غير محدّدة في معالجة الشهادة جعلتها غير مقبولة.

7-2-5 بروتوكول المصافحة

يُعدُّ بروتوكول المصافحة الجزء الأكثر تعقيداً في SSL. ويسمح هذا البروتوكول للخادم والزبون بالتحقق من هوية بعضهما بعضاً وبالتفاوض على خوارزمية التشفير والماك ومفاتيح التشفير التي ستُستخدم لحماية البيانات المُرسَلة في سجل SSL. ويُستخدم بروتوكول المصافحة قبل إرسال أي بيانات من التطبيق.

يشمل بروتوكول المصافحة سلسلةً من الرسائل المُتبادلة بين الزبون والخادم، لكلٍّ منها الصيغة المبينة بالشكل 7-5 (c). وتتألف كل رسالة من ثلاثة حقول:

- النوع (بايت واحد): يدُل هذا الحقل على إحدى الرسائل العشر والمبينة بالجدول 7-2.

- الطول (3 بايتات): طول الرسالة بالبايتات.
- المحتوى (تبادل الرسائل ≤ 0 بايت): المُعامِلات المرتبطة بهذه الرسالة؛ وهي معروضة بالجدول 7-2.

يبيّن الشكل 7-6 التبادل المطلوب لإنشاء اتصال منطقي بين الزبون والخادم، وينقسم إلى أربع مراحل.

❖ المرحلة 1. تأسيس إمكانيات الأمن:

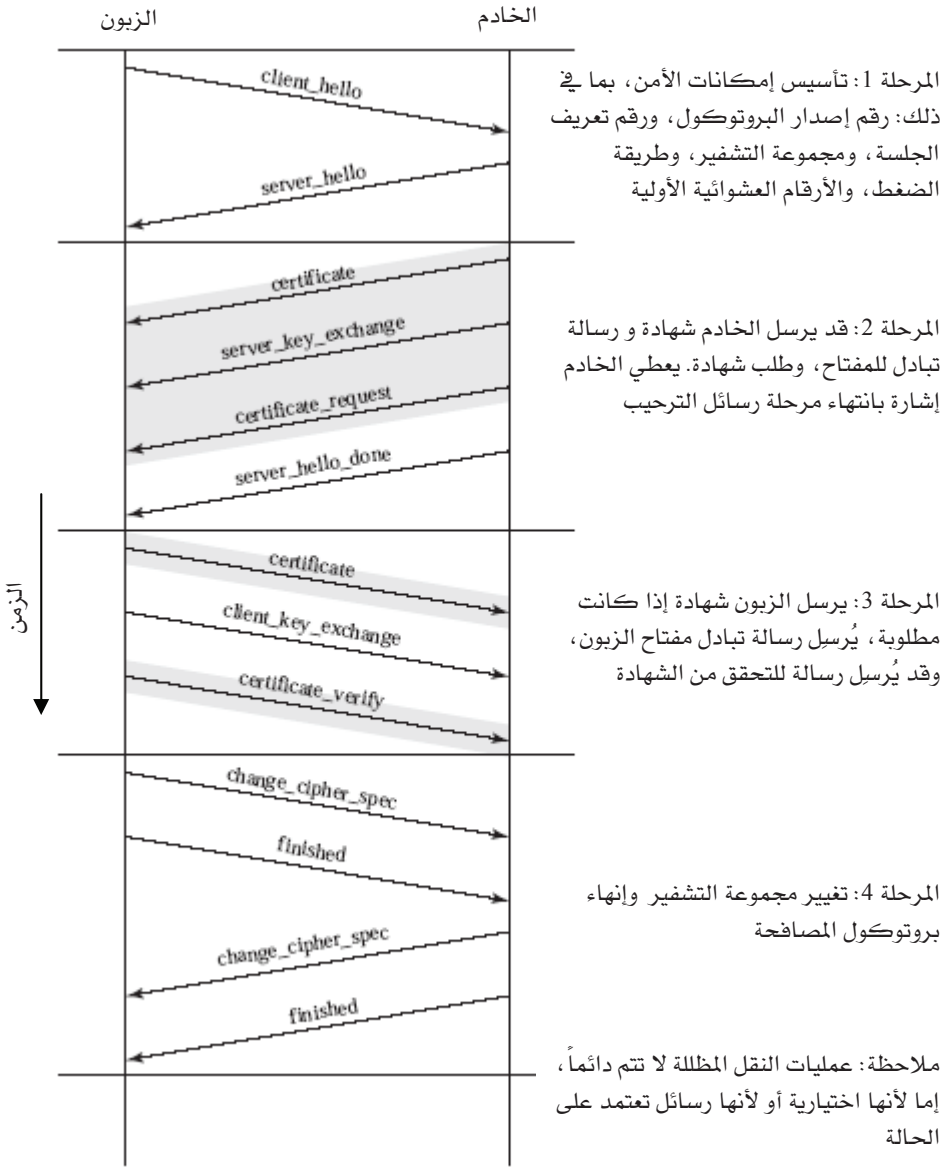
تُستخدَم هذه المرحلة لبدء توصيلة منطقية وتأسيس إمكانيات الأمن المرتبطة بها. يبدأ الزبون بإرسال رسالة ترحيب (client_hello_message) تشمل المُعامِلات الآتية:

- Version (الإصدار): أعلى رقم إصدار لـ SSL يمكن للزبون فهمه.
- Random (تركيبة عشوائية): يولدها الزبون وتشمل 32 بتاً عبارة عن خاتم وقت و28 بايتاً يولدها مُولِّد أرقام عشوائية. وتستخدم هذه القيم لمرة واحدة (nonces) أثناء عملية تبادل المفاتيح لمنع هجمات إعادة التشغيل.
- SessionID (مُعرِّف الجلسة): بطول متغيّر، وتشير قيمته غير الصفرية إلى الزبون الذي يرغب في تحديث مُعامِلات توصيلة حالية أو يريد بدء توصيلة جديدة ضمن هذه الجلسة. أما القيمة الصفرية فتشير إلى أن الزبون يرغب في تأسيس توصيلة جديدة على جلسة جديدة.
- CipherSuite (مجموعة خوارزميات التشفير): هذه قائمة تحتوي على مجموعة من خوارزميات التشفير التي يدعمها الزبون مرتبة حسب الأفضلية من الأعلى إلى الأدنى. يُعرِّف كل عنصر في تلك القائمة خوارزميةً لتبادل المفاتيح ومواصفات الشفرة (CipherSpec)؛ وسنناقشها لاحقاً.
- Compression Method (طريقة الضغط): هذه قائمة بطرق ضغط البيانات التي يدعمها الزبون.

الجدول 7-2: أنواع رسائل بروتوكول مصافحة SSL.

نوع الرسالة	المُعامِلات
طلب ترحيب (hello_request)	لايوجد (null)
ترحيب من الزبون (client_hello)	رقم الإصدار، رقم عشوائي، رقم تعريف الجلسة، مجموعة التشفير، طريقة ضغط البيانات
ترحيب من الخادم (server_hello)	رقم الإصدار، رقم عشوائي، رقم تعريف الجلسة، مجموعة التشفير، طريقة ضغط البيانات
شهادة (certificate)	سلسلة من شهادات X.509v3
تبادل مفتاح الخادم (server_key_exchange)	مُعامِلات، توقيع
طلب شهادة (certificate_request)	النوع، سلطات التصديق
اكتمال الخدمة (server_done)	لا يوجد (null)
التحقق من الشهادة (certificate_verify)	توقيع
تبادل مفتاح الزبون (client_key_exchange)	مُعامِلات، توقيع
الانتهاء (finished)	قيمة كود التحوير

بعد أن يرسل الزبون رسالة الترحيب، ينتظر رسالة ترحيب من الخادم (server_hello message) تتضمّن نفس المُعامِلات المتضمنة في رسالة الترحيب التي أرسلها. تنطبق الاصطلاحات التالية على رسالة الترحيب من الخادم. ويحتوي حقل الإصدار (Version) على أعلى قيمة إصدار يدعمها الخادم من بين قيم الإصدارات المُقترحة من الزبون. ويُولَّد الحقل العشوائي (Random) لدى الخادم بشكلٍ مستقل عن الحقل العشوائي لدى الزبون. وإذا كانت قيمة حقل مُعرِّف الجلسة (SessionID) لدى الزبون لا تساوي صفراً، يستخدم الخادم نفس القيمة؛ وإلا فسيتضمّن هذا الحقل لدى الخادم القيمة الخاصة بجلسة جديدة.



الشكل 7-6: طريقة عمل بروتوكول المصافحة.

يتضمّن حقل مجموعة التشفير (CipherSuite) مجموعة التشفير التي اختارها الخادم من بين المجموعات المقترحة من قِبَل الزبون. كما يتضمّن حقل الضغط (Compression) طريقة الضغط التي اختارها الخادم من بين الطرق المقترحة من قِبَل الزبون.

يتضمن العنصر الأول من مُعامِل مجموعة التشفير طريقة تبادل المفاتيح (الطريقة التي يتم بها تبادل مفاتيح التشفير للتشفير التقليدي والماك). يوفر SSL دعماً للطرق الآتية لتبادل المفاتيح:

- طريقة RSA: يُشفَّر المفتاح السري بمفتاح RSA العام للمُستلِم. ويجب توفُّر شهادة مفتاح عام لمفتاح المُستلِم.
- طريقة ديفي - هيلمان الثابتة (Fixed Deffie-Hellman): هي طريقة ديفي - هيلمان لتبادل المفاتيح وفيها تتضمّن شهادة الخادم مُعامِلات ديفي - هيلمان العامة موقَّعة من سلطة الشهادات (CA). يوفر الزبون مُعامِلات ديفي - هيلمان للمفتاح العام الخاص به إما في شهادة (إذا كان توثيق الزبون مطلوباً)، أو في رسالة تبادل مفاتيح. وتؤدي هذه الطريقة إلى توفير مفتاح سري ثابت بين الطرفين على أساس حساب ديفي - هيلمان باستخدام المفاتيح العامة الثابتة.
- طريقة ديفي - هيلمان العابرة (Ephemeral Diffie-Hellman): تُستخدَم هذه الطريقة لتكوين مفاتيح سرية عابرة (مؤقتة، تُستخدم لمرة واحدة). في هذه الحالة، يتم تبادل مفاتيح ديفي - هيلمان العامة، مُوقَّعةً باستخدام المفتاح الخاص للمُرسِل في أسلوب RSA أو DSS. ويمكن أن يستخدم المُستلِم المفتاح العام المُناظر للتحقق من التوقيع. وتُستخدم الشهادات للتحقق من المفاتيح العامة. وتبدو هذه الطريقة الأكثر أمناً بين خيارات ديفي - هيلمان الثلاثة؛ حيث تؤدّي إلى مفتاح مؤقت موثَّق.
- طريقة ديفي - هيلمان مجهولة المصدر (Anonymous Diffie-Hellman): تُستخدم خوارزمية ديفي - هيلمان الأساسية بدون توثيق. أي يرسل كل

جانب مُعامِلات ديفي - هيلمان العامة إلى الآخرين بدون توثيق. هذه الطريقة عرضة لهجمات "رجل في الوسط"، حيث يستخدم المهاجم طريقة ديفي - هيلمان مجهولة المصدر مع كلا الطرفين المتصلين (وذلك بالتعامل مع كلٍّ منهما على أنه الطرف الآخر).

- طريقة Fortezza: وهي الطريقة المُعرَّفة لأسلوب Fortezza.

بعد تعريف طريقة تبادل المفاتيح يأتي مُعامِل CipherSpec، ويتضمّن الحقول الآتية:

- CipherAlgorithm (خوارزمية التشفير): أيٌّ من الخوارزميات التي ذكرت سابقاً: RC4، RC2، DES، 3DES، DES40، IDEA، Fortezza.
- MACAlgorithm (خوارزمية كود التحوير (الماك)): MD5 أو SHA-1.
- CipherType (نوع التشفير): تشفير كتل أو تشفير تدفق.
- IsExportable (إمكانية التصدير): نعم (true) أو لا (false).
- HashSize (طول كود التحوير): بقيمة 0 أو 16 بايتاً (لـ MD5)، أو 20 بايتاً (لـ SHA-1).
- key material (مادّة المفتاح): سلسلة من البايتات تتضمّن البيانات المُستخدَمة في توليد مفاتيح الكتابة.
- IV Size (طول متجه التهيئة): طول متجه التهيئة لأسلوب التشفير "سَلسَلة الكتل المُشفَّرة" (CBC).

❖ المرحلة 2. توثيق الخادم وتبادل المفاتيح:

يبدأ الخادم هذه المرحلة بإرسال الشهادة الخاصة به، إذا كان هناك حاجة للتحقق من هويته؛ وتتضمّن الرسالة شهادة أو سلسلة من شهادات X.509. يحتاج الأمر إلى رسالة الشهادة عند استخدام أي طريقة متفق عليها لتبادل المفاتيح باستثناء طريقة ديفي - هيلمان مجهولة المصدر. لاحظ أنه في حالة استخدام طريقة ديفي -

هيلمان الثابتة، فإن رسالة الشهادة هذه تُستغَل كرسالة تبادل لمفتاح الخادم؛ لأنها تتضمّن مُعامِلات ديفي - هيلمان العامة للخادم.

بعد ذلك قد تُرسَل رسالة تبادل مفتاح الخادم (server_key_exchange) إذا تطلب الأمر ذلك. ولا يلزم إرسال تلك الرسالة في إحدى حالتين: (1) إذا كان الخادم قد أرسل شهادة مع مُعامِلات ديفي - هيلمان الثابتة، (2) في حالة استخدام طريقة RSA لتبادل المفاتيح. ويلزم إرسال تلك الرسالة في الحالات الآتية:

- طريقة ديفي - هيلمان مجهولة المصدر: يتضمن محتوى الرسالة قيمتي ديفي -هيلمان العامتين (عدد أولي prime number وجذر بدائي primitive root لذلك العدد)، بالإضافة إلى مفتاح ديفي - هيلمان العام للخادم (انظر الشكل 3-10).
- طريقة ديفي - هيلمان العابرة: يتضمّن محتوى الرسالة مُعامِلات ديفي - هيلمان الثلاثة الخاصة بطريقة ديفي - هيلمان مجهولة المصدر، بالإضافة إلى توقيع لتلك المُعامِلات.
- طريقة RSA لتبادل المفاتيح التي يستخدم الخادم فيها خوارزمية RSA ولكن لديه مفتاح RSA للتوقيع فقط: وفقاً لذلك، لا يستطيع الزبون أن يرسل ببساطة مفتاحاً سرياً مُشفَّراً بمفتاح الخادم العام. بدلاً من ذلك، يجب أن يُنشئ الخادم زوجاً من مفاتيح RSA العامة/الخاصة المؤقتة، وأن يستخدم رسالة تبادل مفتاح الخادم (server_key_exchange) لإرسال المفتاح العام. يتضمّن محتوى الرسالة اثنين من مُعامِلات مفتاح RSA العام المؤقتة (الأس exponent ومعامِل حساب الباقي modulus؛ انظر الشكل 3-8)، بالإضافة إلى توقيع لتلك المُعامِلات.
- طريقة Fortezza.

لعله من المفيد هنا ذكر بعض التفاصيل الأخرى فيما يتعلق بالتوقيعات. كالمعتاد يتم إنشاء توقيع بإجراء تحوير للرسالة وتشفير الناتج بالمفتاح الخاص للمرسل. وفي هذه الحالة، يحسب كود التحوير من

Hash(ClientHello.radom || ServerHello.random || ServerParams)

وهكذا يغطي كود التحوير ليس فقط مُعامِلات ديفي - هيلمان أو RSA، ولكن أيضاً قيمتي الأعداد التي تُستخدم لمرة واحدة (nonces) من رسائل الترحيب الأولى. ويضمن هذا الحماية ضد هجمات إعادة التشغيل والتمثيل الخاطئ. وفي حالة توقيع DSS، يُحسب كود التحوير باستخدام خوارزمية SHA-1. وفي حالة توقيع RSA، يحسب كود التحوير باستخدام كلٍّ من MD5 وSHA-1، ثم يوصل كودي التحوير ويشفر الناتج (36 بايتاً) بالمفتاح الخاص للخادم.

بعد ذلك يمكن أن يطلب خادم غير مجهول (أي لا يستخدم طريقة ديفي - هيلمان مجهولة المصدر) شهادةً من الزبون. تتضمّن رسالة طلب الشهادة (certificate_request) معامِلين: نوع الشهادة، وسلطات الشهادة. ويبيّن نوع الشهادة خوارزمية المفتاح العام واستخدامها:

- RSA، توقيع فقط.
- DSS، توقيع فقط.
- RSA لطريقة ديفي - هيلمان الثابتة؛ في هذه الحالة يُستخدَم التوقيع فقط للتوثيق بإرسال الشهادة موقَّعة بـ RSA.
- DSS لطريقة ديفي - هيلمان الثابتة؛ مرة أخرى تُستخدَم فقط للتوثيق.
- RSA لطريقة ديفي - هيلمان العابرة.
- DSS لطريقة ديفي - هيلمان العابرة.
- Fortezza.

أما المُعامِل الثاني في رسالة طلب الشهادة فعبارة عن قائمة بالأسماء المميزة للسلطات المعتمدة لإصدار الشهادات. الرسالة النهائية في المرحلة 2، تُعدُّ رسالة مطلوبة دائماً، هي رسالة اكتمال الخدمة (server_done)، يرسلها الخادم للإشارة إلى الانتهاء من ترحيبه والرسائل المتعلقة به. وبعد إرسال تلك الرسالة ينتظر الخادم رداً من الزبون. وهذه الرسالة ليس لها مُعامِلات.

❖ المرحلة 3. توثيق الزبون وتبادل المفاتيح:

بعد استلام رسالة اكتمال الخدمة (server_done)، يجب أن يتحقق الزبون من أن الخادم قد أعطى شهادة صحيحة إذا كان ذلك مطلوباً وأن يتأكد من أن مُعامِلات رسالة ترحيب الخادم مقبولة. وإذا تم كل ذلك بشكلٍ مرضٍ، يرد الزبون بإرسال رسالة أو أكثر إلى الخادم. وإذا كان الخادم قد طلب شهادة، يبدأ الزبون هذه المرحلة بإرسال رسالة شهادة (certificate). وإذا لم تتوفر شهادة مناسبة، يُرسل الزبون بدلاً من ذلك رسالة تنبيه بعدم وجود شهادة.

بعد ذلك تأتي رسالة تبادل مفتاح الزبون (client_key_exchange)، ويجب إرسالها في تلك المرحلة. ويعتمد محتوى تلك الرسالة على نوع عملية تبادل المفاتيح المُستخدَمة؛ كما هو مبيّن أدناه:

- RSA: يولِّد الزبون السرّ قبل الرئيس (pre-master secret) ويتكون من 48 بايتاً ويشفره بالمفتاح العام الذي حصل عليه من شهادة الخادم أو بمفتاح RSA المؤقت من رسالة تبادل مفتاح الخادم (server_key_exchange). سنوضح كيفية استخدام ذلك السر لحساب "السر الرئيس" (master secret) فيما بعد.
- طريقة ديفي - هيلمان العابرة أو مجهولة المصدر: تُرسَل مُعامِلات ديفي - هيلمان العامة الخاصة بالزبون.

- طريقة ديفي - هيلمان الثابتة: نظراً لأن مُعامِلات ديفي - هيلمان العامة الخاصة بالزبون يكون قد سبق إرسالها في رسالة شهادة، فإن محتوى تلك الرسالة يكون فارغاً.
- Fortezza: تُرسَل مُعامِلات Fortezza الخاصة بالزبون.

في نهاية هذه المرحلة، قد يرسل الزبون رسالة للتحقق من الشهادة (certificate_verify) لتوفير تحقق صريح من شهادة الزبون. وترسل هذه الرسالة فقط بعد أي شهادة زبون لها قابلية للتوقيع (أي كل الشهادات ما عدا تلك التي تتضمن مُعامِلات ديفي - هيلمان الثابتة). وتوقَّع هذه الرسالة كود تحوير بناءً على الرسائل السابقة حسب التعريف الآتي:

```
CertificateVerify.signature.md5_hash
MD5(master_secret || pad_2 || MD5(handshake_messages ||
master_secret || pad_1));
Certificate.signature.sha_hash
SHA(master_secret || pad_2 || SHA(handshake_messages ||
master_secret || pad_1));
```

حيث يرمز المتغيران pad_1 وpad_2 للقيم المُعرَّفة سابقاً لكود توثيق الرسالة (الماك)، ويشير المتغير handshake_messages لجميع رسائل بروتوكول المصافحة التي أُرسِلت أو استُلِمت بعد رسالة ترحيب الزبون (client_hello) باستثناء تلك الرسالة. المتغير master_secret هو السرّ الرئيس المحسوب (وسنوضح كيفية حسابه لاحقاً في هذا الجزء). إذا كان المفتاح الخاص للمُستخدِم هو DSS، فإنه يُستخدَم لتشفير كود تحوير SHA-1. وإذا كان المفتاح الخاص للمُستخدِم هو RSA، فإنه يُستخدَم لتشفير أكواد التحوير لـ MD5 وSHA-1 بعد وصلها. وفي كلتا الحالتين يكون الهدف هو التحقق من ملكية الزبون للمفتاح الخاص لشهادة الزبون. حتى إذا أساء شخص ما استخدام شهادة الزبون، فإنه لن يتمكن من إرسال تلك الرسالة.

❖ **المرحلة 4. الإنهاء:**

تُنهي هذه المرحلة عملية تأسيس توصيلة آمنة. يرسل الزبون رسالة تغيير مواصفات التشفير (change_cipher_spec) وينسخ CipherSpec المعلّق إلى CipherSpec الحالي. لاحظ أن هذه الرسالة لا تُعدُّ جزءاً من بروتوكول المصافحة ولكنها تُرسَل باستخدام بروتوكول تغيير مواصفات التشفير. بعد ذلك مباشرةً يرسل الزبون رسالة الإنهاء (finished) مستخدماً الخوارزميات والمفاتيح والأسرار الجديدة.

تؤكد رسالة الإنهاء إنجاز عمليات تبادل المفاتيح والتوثيق بنجاح. ويتكون محتوى رسالة الإنهاء من قيمتين موصولتين من كودي تحوير:

```
MD5(master_secret || pad_2 || MD5(handshake_messages ||
      Sender || master_secret || pad_1))

SHA(master_secret || pad_2 || SHA(handshake_messages ||
      Sender || master_secret || pad_1))
```

حيث يمثل Sender الكود الذي يتحقق من أن المُرسِل هو الزبون، وتمثل handshake_messages كل البيانات من رسائل المصافحة حتى ما قبل تلك الرسالة.

استجابةً لهاتين الرسالتين، يرسل الخادم رسالة تغيير مواصفات التشفير (change_cipher_spec) الخاصة به، وينقل CipherSpec المعلّق إلى CipherSpec الحالي، ويرسل رسالة إنهاء. عندئذٍ تكون عملية المصافحة قد اكتملت، ويصبح بوسع الزبون والخادم البدء في تبادل بيانات طبقة التطبيقات.

6-2-7 حسابات التشفير

هناك عنصران آخران جديران بالاهتمام: توليد سرٍّ رئيس مشترك بواسطة تبادل المفاتيح، وتوليد مُعامِلات التشفير من السرّ الرئيس.

❖ توليد سرّ رئيس:

يتألف السر الرئيس المشترك من 48 بايتاً (384 بتاً)، ويتم توليده لهذه الجلسة بواسطة تبادل مفاتيح آمن؛ وذلك على مرحلتين. في المرحلة الأولى يتم تبادل السرّ قبل الرئيس (pre-master secret)، وفي الثانية يُحسب السرّ الرئيس من قِبَل كلا الطرفين. هناك طريقتان لتبادل السرّ قبل الرئيس:

- RSA: يتألف السرّ قبل الرئيس من 48 بايتاً ويولَّد من قِبَل الزبون، ويشفَّر بمفتاح RSA العام للخادم، ويُرسَل إلى الخادم. يقوم الخادم بإزالة التشفير باستخدام مفتاحه الخاص لاستعادة السرّ قبل الرئيس.
- ديفي - هيلمان: يولِّد كلٌّ من الزبون والخادم مفتاح ديفي - هيلمان العام. وبعد تبادل هذين المفتاحين، يقوم كل جانب بعملية ديفي - هيلمان الحسابية لتوليد السرّ قبل الرئيس المشترك.

الآن يحسب كلا الجانبين السرّ الرئيس على النحو الآتي:

```
master_secret =
MD5(pre_master_secret || SHA('A' || pre_master_secret || ClientHello.random ||
    ServerHello.random)) ||MD5(pre_master_secret || SHA('BB' ||
    pre_master_secret || ClientHello.random || ServerHello.random)) ||
    MD5(pre_master_secret || SHA('CCC' || pre_master_secret ||
    ClientHello.random || ServerHello.random))
```

حيث ClientHello.random وServerHello.random هما قيمتا العددين اللذين يستخدمان لمرة واحدة (nonces) وتم تبادلهما في رسائل الترحيب الأولى.

❖ توليد مُعامِلات التشفير:

تتطلب مواصفات التشفير CipherSpecs سرّ ماك للكتابة للزبون، وسرّ ماك للكتابة للخادم، ومفتاح كتابة للزبون، ومفتاح كتابة للخادم، ومتجه تهيئة (IV) للكتابة للزبون، ومتجه تهيئة (IV) للكتابة للخادم. وتولَّد تلك المُعامِلات من السرّ

الرئيس على التوالي بحساب كود التحوير للسرّ الرئيس للحصول على سلسلة بايتات آمنة بطول كافٍ لكل المُعامِلات المطلوبة. ويستخدم توليد مادّة المفتاح من السرّ الرئيس نفس الصيغة المتّبعة لتوليد السرّ الرئيس من السرّ قبل الرئيس:

```
key_block =
MD5(master_secret || SHA('A' || master_secret || ServerHello.random ||
    ClientHello.random)) || MD5(master_secret || SHA('BB' ||
    master_secret || ServerHello.random || ClientHello.random)) ||
    MD5(master_secret || SHA('CCC' || master_secret ||
    ServerHello.random || ClientHello.random)) || . . .
```

ويكرر ذلك إلى أن يتم توليد ناتج كافٍ. ويمثل هذا التركيب الخوارزمي دالةً شبه عشوائية. ويمكن اعتبار السرّ الرئيس كقيمة البذرة (seed value) شبه العشوائية للدالة (تعرف هذه القيمة أيضاً بالقيمة المبدئية لمولد الأعداد العشوائية). ويمكن اعتبار الأعداد العشوائية للزبون والخادم كقيم الملح (salt values) التي تقوم بتعقيد عملية تحليل الشفرة (راجع الفصل التاسع لمناقشة قيم الملح).

7-2-7 أمن طبقة النقل

بروتوكول TLS هو مبادرة معيارية للجنة هندسة الإنترنت (IETF) تهدف إلى إصدار معيار للإنترنت من SSL. تم تعريف TLS كمعيار مقترح للإنترنت في الوثيقة RFC 2246؛ وهو يشبه بدرجة كبيرة جداً SSLv3. نسلط الضوء في هذا الجزء على الاختلافات بينهما.

❖ رقم الإصدار:

صيغة سجل TLS هي نفسها تماماً صيغة سجل SSL (الشكل 7-4)، وحقول الترويسة لها نفس المعاني في كليهما. الاختلاف هو في قيم رقم الإصدار. لإصدار TLS الحالي يكون رقم الإصدار الرئيس هو 3 ورقم الإصدار الثانوي هو 1.

❖ كود توثيق الرسالة (الماك):

يختلف أسلوب الماك المستخدم في كلٍّ من SSLv3 وTLS في نقطتين: الخوارزمية الفعلية ومجال حساب الماك. يستخدم TLS خوارزمية HMAC المُعرَّفة في RFC 2104. ذكرنا في الفصل الثالث أن HMAC مُعرَّف على النحو الآتي:

$$HMAC_K(M) = H[(K^+ \oplus opad) \| H[(K^+ \oplus ipad) \| M]]$$

حيث:

H = دالة التحوير المتضمَّنة (يستخدم TLS دالة MD5 أو SHA-1).

M = رسالة المُدخَل لـ HMAC.

K^+ = السرّ الرئيس مطولٌ بأصفار من ناحية اليسار حتى يصبح الناتج مساوياً لطول الكتلة لكود التحوير (في حالة MD5 وSHA-1 طول الكتلة يساوي 512 بتاً).

ipad = 00110110 (أي 36 في النظام الست عشري) مكررة 64 مرة (ليصبح الطول 512 بتاً).

opad = 01011100 (أي 5C في النظام الست عشري) مكررة 64 مرة (ليصبح الطول 512 بتاً).

يستخدم SSLv3 نفس الخوارزمية باستثناء أن بايتات الحشوة (padding bytes) تُوصَل بالمفتاح السري بدلاً من حساب XOR بينها وبين المفتاح السري وطول الكتلة. مستوى الأمن هو تقريباً نفسه في الحالتين.

في حالة TLS، يتضمن حساب الماك الحقول المذكورة في التعبير الآتي:

```
HMAC_hash(MAC_write_secret, seq_num || TLSCompressed.type ||
          TLSCompressed.version || TLSCompressed.length ||
          TLSCompressed.fragment)
```

يغطي حساب الماك كل الحقول المغطاة في حساب SSLv3، بالإضافة إلى حقل TLSCompressed.version والذي يشير لرقم إصدار البروتوكول المُستخدَم.

❖ الدالة شبه العشوائية:

يستخدم TLS دالة شبه عشوائية تُعرف باسم PRF لتوسيع قيمة السرّ إلى كتل بيانات تُستخدم لتوليد المفاتيح أو التحقق من صحتها. ويكمن الهدف من ذلك في التمكن من استخدام سرٍّ مشتركٍ صغيرٍ نسبياً ومع ذلك توليد كتل أطول من البيانات بطريقة آمنة إزاء أنواع الهجمات التي قد تُشن على دوال التحوير والماك. تعتمد PRF على الدالة الآتية لتوسيع البيانات (الشكل 7-7):

$$\begin{aligned}P_hash(secret, seed) = {} & HMAC_hash(secret, A(1) \,||\, seed) \,|| \\ & HMAC_hash(secret, A(2) \,||\, seed) \,|| \\ & HMAC_hash(secret, A(3) \,||\, seed) \,|| \ldots\end{aligned}$$

حيث تُعرَّف $A()$ على النحو الآتي:

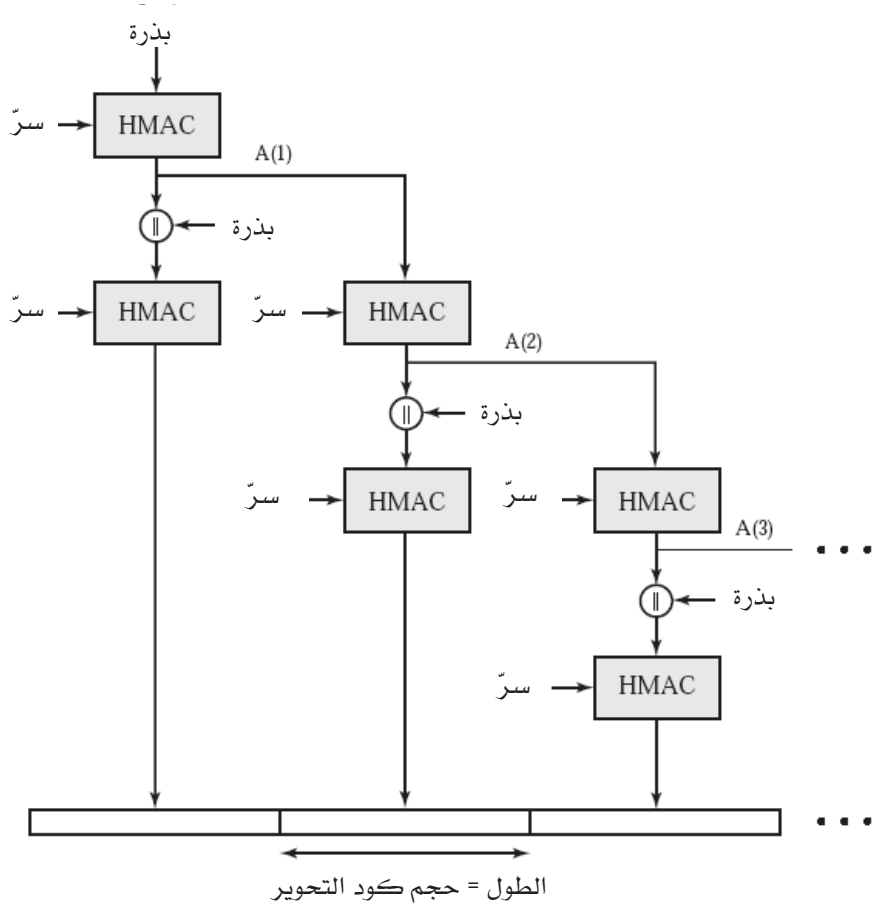
$$A(0) = seed$$
$$A(i) = HMAC_hash(secret, A(i\text{-}1))$$

تستخدم دالة توسيع البيانات خوارزمية HMAC، مع MD5 أو SHA-1 كدالة التحوير التحتية. وكما ترى يمكن تكرار P_hash أيّ عدد من المرات نحتاجه لتوليد كمية البيانات المطلوبة. فعلى سبيل المثال، إذا استخدمنا P_SHA-1 لتوليد 64 بايتاً من البيانات، يجب تكررها أربع مرات، ومن ثم إنتاج 80 بايتاً من البيانات، ويتم إهمال الـ 16 بايتاً الأخيرة منها. وفي هذه الحالة، نحتاج لتكرار P_MD5 أيضاً أربع مرات، مما ينتج 64 بايتاً بالضبط. لاحظ أن كل تكرار يتضمن تنفيذ HMAC مرتين، يتضمن كلٌّ منهما بدوره تنفيذ خوارزمية التحوير التحتية مرتين.

لكي تكون دالة PRF آمنة بأقصى قدرٍ ممكن فإنها تستخدم خوارزميتي تحوير بطريقةٍ تضمن أمنها إذا ما بقت إحدى الخوارزميتين آمنة. تُعرّف دالة PRF على النحو الآتي:

$$\begin{aligned}PRF(secret, label, seed) = {} & P_MD5(S1, label \,||\, seed) \oplus \\ & P_SHA\text{-}1(S2, label \,||\, seed)\end{aligned}$$

تأخذ PRF قيمة سرية ووسمة تعريف وقيمة بذرة، وتنتج مُخرجاً بطولٍ اعتباطي. ويتم إنشاء الناتج بتقسيم قيمة السرّ إلى نصفين (S1 وS2)، ثم حساب P_hash لكل نصف باستخدام MD5 على نصف وSHA-1 على النصف الآخر. ثم تُجرى عملية XOR على النتيجتين للحصول على المُخرج النهائي؛ لهذا الغرض وبشكلٍ عام تُكرّر P_MD5 أكثر من P_SHA-1 لإنتاج كمية متساوية من البيانات التي تدخل لعملية XOR.



الشكل 7-7: دالة P_hash(secret, seed) لـ TLS.

❖ أكواد التنبيهات:

يدعم TLS كل أكواد التنبيهات المُعرَّفة في SSLv3 باستثناء no_certificate (لا توجد شهادة). هناك عدة أكواد إضافية معرَّفة في TLS؛ منها الأكواد الآتية التي تُعدُّ دائماً تنبيهات فادحة (fatal):

- decryption_failed (فشل إزالة التشفير): تم إزالة تشفير النص على نحو خاطئ؛ إما لكونه ليس من المضاعفات الزوجية لطول الكتلة، أو لأنه عند فحص قيم الحشوة (padding) وجد أنها خطأ.
- record_overflow (فيض السجل): تم استلام سجل TLS بحمولة (نص مُشفَّر) يزيد طولها عن 2^{14}+2048 بايتاً، أو أن النص الناتج من إزالة التشفير يزيد طوله عن 2^{14}+1024 بايتاً.
- unknown_ca (سلطة شهادات غير معروفة): تم استلام سلسلة أو سلسلة جزئية للشهادة صحيحة، ولكن الشهادة لم تُقبل نظراً لأنه تعذّر تحديد مكان سلطة الشهادات أو لعدم تطابقها مع سلطة شهادات معروفة ومؤتمنة.
- access_denied (ممنوع الوصول): تم استلام شهادة صحيحة، غير أنه عند تطبيق إجراء التحكم في الوصول قرَّر المُرسِل عدم الاستمرار في التفاوض.
- decode_error (خطأ في إزالة التكويد): تعذّر إزالة تكويد الرسالة نظراً لأن قيمة أحد الحقول تقع خارج المدى المحدّد أو أن طول الرسالة غير صحيح.
- export_restriction (قيود على التصدير): تم اكتشاف تفاوض على طول المفتاح غير متوافق مع القيود على التصدير.
- protocol_version (رقم إصدار البروتوكول): تم التعرف على رقم إصدار البروتوكول الذي يستخدمه الزبون، ولكن وُجد أنه غير مدعوم.

- insufficient_security (أمن غير كافٍ): يتم إرسالها بدلاً من handshake_failed عندما يفشل التفاوض بالتحديد لأن الخادم يتطلّب شفرةً أكثر أمناً من تلك التي يدعمها الزبون.
- internal_error (خطأ داخلي): طرأ خطأ داخلي غير مرتبط بالنظير أو بصحة البروتوكول، مما أدى إلى استحالة الاستمرار.

تتضمّن بقيّة التنبيهات الجديدة مايلي:

- decrypt_error (خطأ في إزالة التشفير): فشلت عملية من عمليات التشفير الخاصة بالمصافحة؛ بما في ذلك عدم التمكن من التحقق من التوقيع أو إزالة تشفير تبادل مفتاح أو التحقق من سلامة رسالة إنهاء.
- user_canceled (تم الإلغاء من قِبَل المستخدم): تم إلغاء هذه المصافحة لسببٍ ما لا علاقة له بفشل البروتوكول.
- no_renegotiation (لا إعادة للتفاوض): تُرسَل من قِبَل الزبون ردّاً على طلب ترحيب، أو تُرسَل من قِبَل الخادم ردّاً على رسالة ترحيب الزبون بعد المصافحة الأولى. ويؤدّي أيٌّ من هاتين الرسالتين في العادة إلى إعادة التفاوض، ولكن رسالة التنبيه هذه تشير إلى أن المُرسِل غيرُ قادر على إعادة التفاوض. وهذه الرسالة دائماً تحذيرية.

مجموعات التشفير (cipher suites): هناك عدّة اختلافات طفيفة بين مجموعات التشفير المتوفرة في كلٍّ من SSLv3 وTLS:

- تبادل المفاتيح: يدعم TLS كلّ أساليب تبادل المفاتيح المتوفرة في SSLv3 ما عدا Fortezza.
- خوارزميات التشفير المتماثل: يتضمّن TLS كل خوارزميات التشفير المتماثل الموجودة في SSLv3 ما عدا Fortezza.

❖ أنواع شهادات الزبون:

يُعرِّف TLS أنواع الشهادات الآتية التي يمكن أن تُطلب في رسالة طلب شهادة (certificate_request): rsa_sig، dss_sign، rsa_fixed_dh، dss_fixed_dh. وكل هذه الأنواع مُعرَّفة في SSLv3. بالإضافة لذلك يتضمن SSLv3 الأنواع الآتية: rsa_ephemeral_dh، dss_ephemeral_dh، fortezza_kea. تتضمّن طريقة ديفي - هيلمان العابرة توقيع مُعامِلات ديفي - هيلمان بـ RSA أو DSS؛ لكن في TLS يُستخدم rsa_sign أو dss_sign لهذا الغرض، ولا يلزم استخدام نوع توقيع مستقل لتوقيع مُعامِلات ديفي - هيلمان. لا يتضمّن TLS طريقة Fortezza.

❖ رسائل التحقق من الشهادة ورسائل الإنهاء:

في رسالة التحقق من الشهادة بـ TLS، تُحسب أكواد MD5 وSHA-1 للتحوير فقط على رسائل المصافحة. تذكّر أنه في SSLv3 يتضمن حساب كود التحوير أيضاً السرّ الرئيس وبايتات الحشو. وهناك إحساس بأن هذه الحقول الإضافية لا تؤدي إلى أي أمن إضافي.

كما هو الحال مع رسالة الإنهاء في SSLv3، رسالة الإنهاء في TLS هي عبارة عن كود تحوير يعتمد على السرّ الرئيس المشترك، ورسائل المصافحة السابقة، والوسمة التي تميّز الزبون من الخادم. ويختلف الحساب نوعاً ما، ففي TLS:

```
PRF(master_secret, finished_label, MD5(handshake_messages) ||
    SHA-1(handshake_messages))
```

حيث finished_label هي سلسلة البتات التي تمثل حالة "الزبون انتهى" لدى الزبون وحالة "الخادم انتهى" لدى الخادم.

❖ حسابات التشفير:

يُحسب السرّ قبل الرئيس في TLS بنفس الطريقة المستخدمة في SSLv3. كما في SSLv3، يُحسب السرّ الرئيس في TLS كناتج دالة التحوير على السرّ قبل

الرئيس والعددين العشوائيين في رسالتي الترحيب. ويختلف شكل حساب TLS عن SSLv3، ويُعرَّف على النحو الآتي:

```
master_secret = PRF(pre_master_secret, "master secret",
                    ClientHello.random || ServerHello.random)
```

يُكرر تطبيق الخوارزمية حتى تنتج مُخرجاً شبه عشوائي يتألف من 48 بايتاً. ويُعرَّف حساب مادة كتلة المفتاح (مفاتيح ماك السرية، ومفاتيح تشفير الجلسة، ومتجهات التهيئة كالتالي:

```
key_block = PRF(master_secret, "key expansion",
                SecurityParameters.server_random ||
                SecurityParameters.client_random)
```

وتكرار ذلك إلى أن يتولد ناتجٌ كافٍ. كتلة المفتاح (key_block) في SSLv3 دالة في السرّ الرئيس والأعداد العشوائية للزبون والخادم؛ لكن تختلف الخوارزمية الفعلية في TLS.

❖ الحشوة (padding):

كمية الحشوة المضافة في SSL قبل تشفير بيانات المُستخدِم هي الحدّ الأدنى المطلوب لكي يصبح الحجم الكلي للبيانات التي ستُشفَّر من مضاعفات حجم كتلة الشفرة. أما في TLS فيمكن أن تكون الحشوة المضافة بأي كمية تجعل المجموع من مضاعفات حجم كتلة الشفرة، بما لا يزيد على 255 بايتاً. على سبيل المثال، إذا كان حجم النص الأصلي (plaintext) (أو النص المضغوط في حالة استخدام الضغط) مضافاً إليه الماك وبايت طول الحشوة (padding.length) يساوي 79 بايتاً، فإن طول الحشوة بالبايتات يمكن أن يكون 1 أو 9 أو 17 وهكذا حتى 249. قد يُستخدم طول حشوة متغيّر لإحباط الهجمات المعتمدة على تحليل أطوال الرسائل المتبادلة.

3-7 المُعامَلات الإلكترونية الآمنة (SET)

المُعامَلات الإلكترونية الآمنة (SET) هي نظام مفتوح للتشفير والأمن صُمِّم لحماية استخدام بطاقات الائتمان على الإنترنت. والنسخة الحالية منه هي SETv1، وقد ظهرت لتلبية الحاجة إلى معايير أمن من قِبَل شركات بطاقات الائتمان الماستر كارد (MasterCard) والفيزا (Visa) في فبراير/شباط عام 1996. وشاركت في تطوير المواصفات الأولية مجموعة كبيرة من الشركات كان من بينها: IBM، ومايكروسوفت، ونيتسكيب، وRSA، وTerisa، وVerisign. وبدأت منذ عام 1996 عدة محاولات لاختبار تلك الفكرة، وبحلول عام 1998 ظهرت أول موجة من المنتجات المتوافقة مع نظام SET.

لا يُعدُّ نظام SET في حد ذاته نظاماً للدفع (أي تسديد المبالغ المستحقة)؛ ولكنه بالأحرى مجموعة من بروتوكولات وصيغ الأمن التي تمكّن المستخدمين من استعمال البنية التحتية لأنظمة الدفع ببطاقات الائتمان الحالية على شبكة مفتوحة كالإنترنت بشكلٍ آمنٍ. يوفر SET ثلاث خدمات:

- قناة اتصال آمنة بين جميع الأطراف المشتركة في المُعامَلة.
- نظام للثقة باستخدام X.509 للشهادات الرقمية.
- ضمان الخصوصية لأن المعلومات تكون متوفرة فقط للأطراف المعنية بتلك المُعامَلة عندما وحيثما يكون ذلك ضرورياً.

مواصفات SET معقّدة ومُعرَّفة في ثلاثة كتب صدرت في مايو عام 1997:

- الكتاب الأول: وصف للمتطلبات التجارية (80 صفحة).
- الكتاب الثاني: دليل المبرمِج (629 صفحة).
- الكتاب الثالث: التعريف الرسمي للبروتوكول (262 صفحة).

أي ما يُشكِّل في مجموعه 971 صفحة من المواصفات. في المقابل، كانت مواصفات SSLv3 مؤلَّفة من 63 صفحة، بينما تألَّفت مواصفات TLS من 80 صفحة.

ولذا فسنقتصر في هذا الجزء فقط على إعطاء ملخص سريع لهذا المعيار متعدد الجوانب.

7-3-1 نظرة عامة على SET

من الطرق الجيدة لبدء مناقشة SET أن ننظر إلى المتطلبات التجارية له وميزاته الرئيسة والمشاركين في مُعامَلاته.

❖ **المتطلبات:**

يعرض الكتاب الأول للمواصفات المتطلبات التجارية الآتية لأنظمة الدفع الآمنة ببطاقات الائتمان على الإنترنت والشبكات الأخرى:

- توفير السرية لمعلومات الدفع والطلبية: من الضروري لحاملي البطاقات الاطمئنان إلى أن هذه المعلومات آمنة ويمكن الوصول إليها فقط بواسطة المستلم المعنيّ. وتقلل السرية أيضاً خطر الاحتيال بواسطة أيٍّ من الأطراف المشتركة في المُعامَلة أو من قِبَل أطراف ثالثة مؤذية. ويستخدم SET التشفير لتوفير السرية.
- ضمان سلامة البيانات المُرسَلة: أي ضمان عدم حدوث تغييرات (دون اكتشافها) في محتوى رسائل SET أثناء انتقالها. وتُستخدَم التوقيعات الرقمية لضمان ذلك.
- توفير آليات للتحقق من أن حامل البطاقة هو مُستخدِم مُصرَّح له: إن وجود آلية للربط بين حامل البطاقة ورقم حساب معيّن تقلل من حوادث الاحتيال والكلفة الإجمالية لمعالجة عملية الدفع. وتستخدم التوقيعات الرقمية والشهادات لإثبات أن حامل البطاقة هو المُستخدِم الحقيقي لحساب صحيح.
- توفير آليات للتحقق من أن التاجر يقبل الدفع ببطاقة الائتمان من خلال علاقته بمؤسسة مالية: يُعدُّ ذلك تكملةً للمطلب السابق. فمن الضروري أن يكون حاملو البطاقات قادرين على تمييز التجار الذين يمكنهم إجراء

مُعامَلات آمنة معهم. ومرة أخرى تُستخدَم التوقيعات الرقمية والشهادات لهذا الغرض.

- ضمان استخدام أفضل الممارسات الأمنية وأساليب تصميم النظام لحماية كل الأطراف الشرعية في صفقة تجارة عبر الإنترنت: يُعدُّ SET مواصفاتٍ مجرّبة بشكلٍ جيد ومبنيةٍ على خوارزميات وبروتوكولات تشفير آمنة للغاية.
- عدم اعتماد البروتوكول على استخدام آليات النقل الآمن، وكذلك عدم منع استخدامها: يمكن تشغيل SET بشكلٍ آمن على رصّة بروتوكولات TCP/IP. ومع ذلك لا يتداخل SET مع استخدام آليات الأمن الأخرى مثل IPSec وSSL/TLS.
- تسهيل العمل البيني بين موفري البرمجيات وموفري خدمة الشبكة وتشجيعه: ينبغي ألا تعتمد بروتوكولات وصيغ SET على المكونات المادية ونظم التشغيل وبرامج الويب.

7-3-2 الميزّات الرئيسة لنظام SET

لتلبية المتطلبات التي عرضناها آنفاً، يتضمن SET الميزّات الآتية:

- سرية المعلومات: يتم تأمين حساب حامل البطاقة ومعلومات الدفع أثناء انتقالها عبر الشبكة. من الميزات المثيرة المهمة لـ SET أنه يمنع التاجر من معرفة رقم بطاقة الائتمان لحامل البطاقة؛ هذا شأن خاص فقط بالمصرف الذي أصدرها. ويُستخدَم التشفير التقليدي بواسطة DES لتوفير السرية.
- سلامة البيانات: تتضمّن معلومات الدفع المُرسَلة من حاملي البطاقات إلى التجار معلومات الطلبية، وبيانات شخصية، وتعليمات الدفع. يوفر SET ضماناً بعدم تعديل محتويات تلك الرسائل أثناء انتقالها. وتُستخدم توقيعات RSA الرقمية بكود التحوير SHA-1 وسيلةً لفحص سلامة الرسائل. ويتم أيضاً حماية رسائل معينة عن طريق HMAC باستخدام SHA-1.

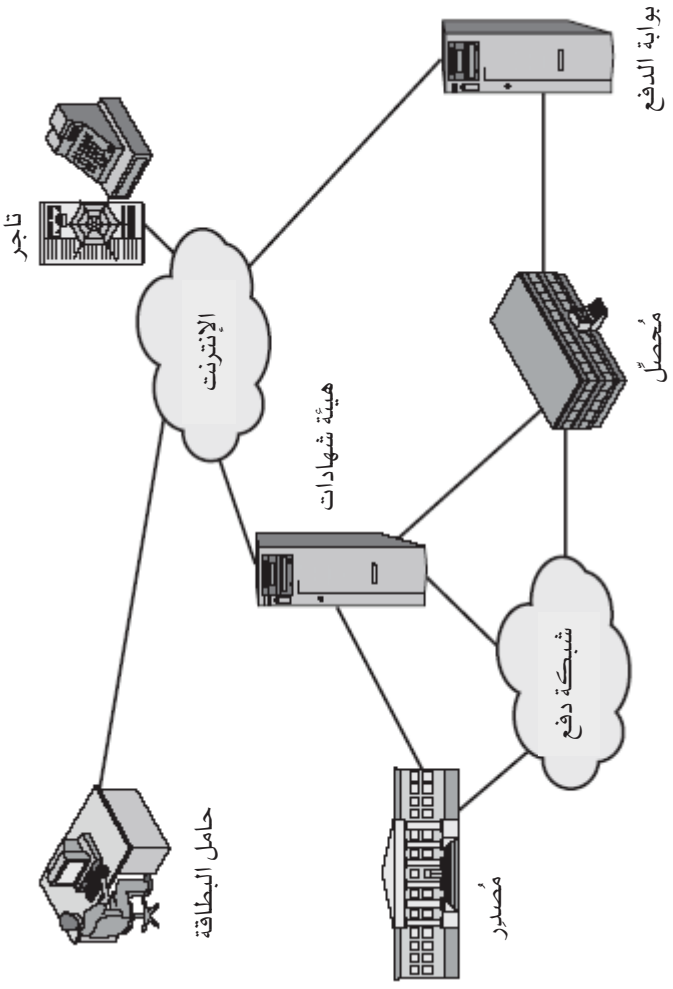
- التحقّق من حساب حامل البطاقة: يُمكِّن SET التجار من التحقق من أن حامل البطاقة هو المُستخدِم الحقيقي لرقم حساب بطاقة صحيح. ويستخدم SET لهذا الغرض شهادات X.509v3 الرقمية مع توقيعات RSA.
- التحقّق من التاجر: يُمكِّن SET حاملي البطاقات من التحقق من أن التاجر له علاقة بمؤسسة مالية تسمح له بقبول بطاقات الدفع. مرة أخرى يستخدم SET لهذا الغرض شهادات X.509v3 الرقمية بتوقيعات RSA.

لاحظ أنه على خلاف IPSec وSSL/TLS، يوفر SET خياراً واحداً فقط لكل خوارزمية تشفير. وهذا أمر متوقّع، فنظام SET يخدم تطبيقاً واحداً فقط يتضمّن مجموعة واحدة من المتطلبات، بينما يهدف IPSec وSSL/TLS إلى دعم تشكيلة من التطبيقات المختلفة.

❖ الأطراف المشاركة في SET:

يبيّن الشكل 7-8 الأطراف المشاركة في SET، وتشمل ما يلي:

- حامل البطاقة (cardholder): في البيئة الإلكترونية، يتفاعل العميل (المستهلك أو المشتري) مع التاجر من خلال الحاسبات الشخصية المتصلة بالإنترنت. حامل البطاقة هو حاملٌ مصرحٌ له لبطاقة دفع (كبطاقة الماستر كارد أو الفيزا) يُصدرها مُصدِر (issuer).
- التاجر (merchant): هو شخصٌ أو منظمة لديها سلعٌ أو خدمات للبيع لحاملي بطاقات الدفع. في العادة تُعرض هذه السلع أو الخدمات عن طريق موقع ويب أو رسائل البريد الإلكتروني. ويجب أن يكون للتاجر الذي يقبل بطاقات الدفع علاقة بمُحصِّل (acquirer).
- المُصدِر (issuer): هو مؤسسة مالية (كمصرف) توفر بطاقات الدفع لحاملي البطاقات. في العادة يتم التقدم لفتح الحسابات وفتحها إما بالبريد أو شخصياً. في النهاية يتحمّل المُصدِر مسؤولية دفع دين حامل البطاقة.



الشكل 7-8: مكوِّنات التجارة الإلكترونية الآمنة.

- المُحصِّل (acquirer): هو المؤسسة المالية التي تؤسّس حساباً للتاجر وتقوم بمعالجة عمليات التفويض والدفع بالبطاقات. وعادةً ما يقبل التجار أكثر من نوع من بطاقات الائتمان لكنهم لا يرغبون في التعامل بشكلٍ فردي مع هيئات مختلفة للبطاقات أو مع مُصدِرين متعددين. ويوفر المُحصِّل للتاجر خدمة التحقق من أن حساب البطاقة نشِط وأن مبلغ الشراء المقترح لا يتجاوز حدّ الائتمان المصرّح به. ويقوم المُحصِّل أيضاً بتحويل المبلغ المستحق لحساب التاجر إلكترونياً. وبعد ذلك، يُعوَّض المحصِّل من قبل المُصدِر عبر شبكةٍ من شبكات الدفع لتحويل الأموال بشكلٍ إلكتروني.
- بوابة الدفع (payment gateway): هي وظيفة يقوم بها المُحصِّل أو طرف ثالث معيّن، وذلك لمعالجة رسائل الدفع من التجار. وتمثل بوابة الدفع واجهة بين SET وشبكات الدفع الحالية للبطاقات المصرفية، وتقوم بوظائف الدفع والتفويض به. يتبادل التاجر رسائل SET مع بوابة الدفع على الإنترنت، بينما تتصل بوابة الدفع مباشرةً أو من خلال شبكة بنظام المعالجة المالية للمُحصِّل.
- سلطة الشهادات (CA): هي كيان مؤتمن لإصدار شهادات المفاتيح العامة X.509v3 لحاملي البطاقات والتجار وبوابات الدفع. يعتمد نجاح SET على توفر بنية تحتية لسلطات الشهادات لهذا الغرض. وكما نوقش في الفصول السابقة يُستخدم تركيب هرمي من سلطات الشهادات، بحيث لا يحتاج المشاركون للحصول على شهاداتهم مباشرةً من هيئة الجذر (root authority).

نستعرض الآن باختصار تسلسل الأحداث المطلوب لإجراء مُعامَلة، وبعد ذلك نناقش بعض تفاصيل التشفير.

1. يقوم الزبون بفتح حساب: يحصل الزبون على حساب بطاقة ائتمان (كالماستر كارد أو الفيزا) من مصرف يدعم الدفع الإلكتروني ونظام SET.

2. يستلم الزبون شهادة: بعد التحقق المناسب من هوية الزبون، يستلم الزبون شهادة رقمية X.509v3 موقَّعة من المصرف. توثق الشهادة مفتاح RSA العام للزبون وتاريخ انتهائه. أيضاً يتم تأسيس علاقة (يضمنها المصرف) بين زوج المفاتيح للزبون وبطاقته الائتمانية.
3. للتجار شهاداتهم الخاصة: التاجر الذي يقبل نوعاً معيناً من البطاقات يجب أن يكون لديه شهادتان لمفتاحين عامين خاصين به: واحد لتوقيع الرسائل، وآخر لتبادل المفاتيح. ويحتاج التاجر أيضاً لنسخة من شهادة المفتاح العام لبوابة الدفع.
4. يقوم الزبون بطلبية ما: قد تتضمّن هذه العملية أن يستعرض الزبون موقع الويب الخاص بالتاجر أولاً لاختيار المواد وتحديد أسعارها. ثم يرسل الزبون قائمة المواد التي سيشتريها للتاجر الذي يرُدّ بإرسال استمارة طلبية تتضمن قائمة بالمواد المطلوبة وأسعارها والسعر الإجمالي ورقم الطلبية.
5. يتم التحقق من هوية التاجر: بالإضافة إلى نموذج الطلبية، يُرسل التاجر نسخةً من شهادته، لكي يتمكن الزبون من التحقق من أنّه يتعامل مع متجر غير مزيف.
6. يتم إرسال بيانات الطلبية والدفع: يرسل الزبون المعلومات المتعلقة بكلٍّ من الطلبية والدفع للتاجر مع شهادة الزبون. تؤكد معلومات الطلبية شراء المواد المذكورة في نموذج الطلبية، بينما تتضمّن معلومات الدفع تفاصيل بطاقة الائتمان. تكون معلومات الدفع مُشفَّرة بحيث لا يمكن للتاجر قراءتها. تُمكِّن شهادة الزبون التاجر من التحقق من هوية الزبون.
7. يطلب التاجر ترخيصاً بالدفع: يرسل التاجر معلومات الدفع إلى بوابة الدفع، ويطلب تأكيداً بأن الحد الائتماني المتوفر للزبون كافٍ لعملية الشراء تلك.
8. يرسل التاجر إلى الزبون إشعاراً باستلام بيانات الطلبية.
9. يشحن التاجر السلع أو يوفر الخدمة المطلوبة للزبون.
10. يطلب التاجر الدفع: يُرسَل هذا الطلب إلى بوابة الدفع التي تقوم بكل عمليات المعالجة اللازمة لإتمام عملية الدفع.

7-3-3 التوقيع المزدوج (Dual Signature)

قبل النظر في تفاصيل بروتوكول SET، دعنا نناقش أسلوباً جديداً مهماً هو التوقيع المزدوج الذي تم استحداثه للاستخدام في SET. والهدف من التوقيع المزدوج هو الربط بين رسالتين ترسلان لمستلمين مختلفين. وفي الحالة التي نحن بصددها، يرغب الزبون في إرسال معلومات الطلبية (OI) للتاجر وإرسال معلومات الدفع (PI) للمصرف. ولا يحتاج التاجر لمعرفة رقم بطاقة الائتمان، ولا يحتاج المصرف لمعرفة تفاصيل الطلبية. ويتوفر للزبون حماية إضافية من حيث السرية وذلك بجعل كلٍّ من هذين العنصرين مستقلاً عن الآخر. ومع ذلك ينبغي الربط بين هذين العنصرين بطريقة ما بحيث يمكن أن تستخدم لحل النزاعات عند الضرورة. وهذا الربط مطلوب حتى يمكن أن يثبت الزبون أن عملية الدفع هذه مقصودة لهذه الطلبية بعينها وليست لسلعة أو خدمة أخرى.

لإدراك مدى الحاجة لهذا الربط، افترض أن الزبون أرسل رسالتين للتاجر: رسالة OI موقَّعة ورسالة PI مُوقَّعة. وافترض أن التاجر مرّر رسالة PI تلك إلى المصرف. وإذا تمكن التاجر من التقاط رسالة OI أخرى من هذا الزبون، فإنه بوسعه أن يدّعي بأنّ رسالة الـ OI الأخيرة، وليس رسالة الـ OI الأصلية، خاصة بذلك الـ PI . تمنع عملية الربط حدوث ذلك.

يبيّن الشكل 7-9 استخدام التوقيع المزدوج لتلبية المطلب في الفقرة السابقة. يأخذ الزبون كود التحوير (باستخدام SHA-1) لكلٍ من PI وOI. ثم يوصل هذين الكودين ويأخذ كود التحوير للناتج. وأخيراً، يُشفّر الزبون كود التحوير النهائي بمفتاحه الخاص لتوقيعه، ومن ثمَّ ينتج التوقيع المزدوج. ويمكن تلخيص تلك العملية على النحو الآتي:

$$DS = E(PR_c, [H(H(PI)||H(OI))])$$

حيث تمثل PR_c المفتاح الخاص لتوقيع الزبون. الآن افترض أن التاجر بحوزته التوقيع المزدوج (DS) وOI وPIMD (أي خلاصة رسالة PI). وافترض أيضاً أن التاجر لديه أيضاً المفتاح العام للزبون ويمكنه الحصول عليه من شهادة الزبون. عندئذ يمكن أن يحسب التاجر القيم:

H(PIMD||H[OI]); D(PU_c, DS)

حيث PU_c هو المفتاح العام لتوقيع الزبون. إذا كانت هاتان القيمتان متساويتين، يكون التاجر قد تحقق من صحة التوقيع. بنفس الطريقة، إذا كان لدى المصرف DS وPI وOIMD (خلاصة رسالة OI) ومفتاح الزبون العام، عندئذ يمكن أن يحسب المصرف:

H(H[OI]||OIMD); D(PU_c, DS)

مرة أخرى إذا تساوت هاتان القيمتان، يكون المصرف قد تحقق من صحة التوقيع.

الخلاصة:

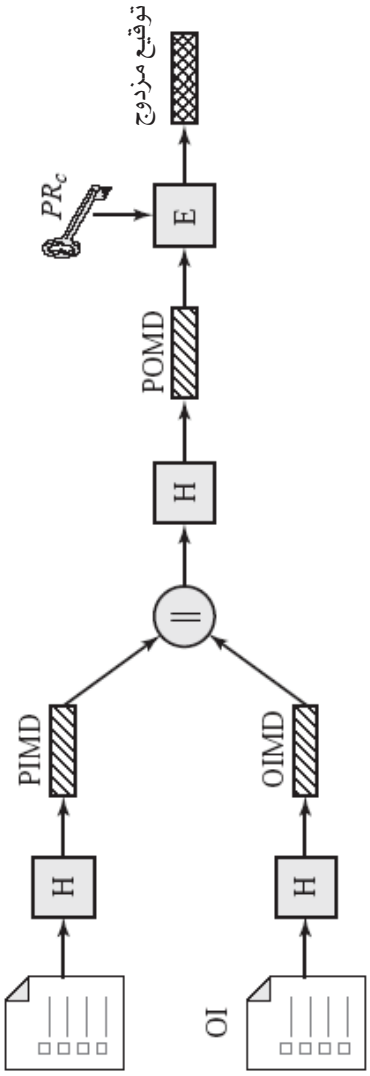
1. استلم التاجر OI وتحقق من صحة التوقيع.
2. استلم المصرف PI وتحقق من صحة التوقيع.
3. ربط الزبون بين OI وPI، ويمكنه إثبات الصلة بينهما.

على سبيل المثال، افترض أن التاجر يرغب في استبدال OI آخر في هذه المُعامَلة لغرضٍ ما يعود عليه بفائدة. عليه عندئذ أن يجد OI آخر له كود تحوير متوافق مع OIMD الحالية. عند استخدام SHA-1، يُعدُّ هذا الأمر غير ممكن. وعليه، لا يستطيع التاجر ربط OI آخر بذلك الـ PI.

4-3-7 معالجة عملية الدفع

يعرض الجدول 7-3 قائمةً بأنواع المُعامَلات التي يدعمها SET. سنناقش بعض تفاصيل المُعامَلات الآتية:

- طلب الشراء (purchase request).
- تصريح الدفع (payment authorization).
- تحصيل المبلغ المستحق (payment capture).



الشكل 7-9: إنشاء التوقيع المزدوج.

الجدول 7-3: أنواع مُعامَلات SET.

تسجيل حامل البطاقة	يجب على حاملي البطاقات التسجيل لدى CA قبل أن يمكنهم إرسال رسائل SET للتجار.
تسجيل التاجر	يجب أن يسجل التجار مع CA قبل أن يمكنهم تبادل رسائل SET مع الزبائن وبوابات الدفع.
طلب شراء	رسالة من الزبون إلى التاجر تتضمن معلومات طلب الشراء (OI) للتاجر ومعلومات الدفع (PI) للمصرف.
تفويض الدفع	تبادل بين التاجر وبوابة الدفع للتخويل بدفع مبلغ الشراء من حساب بطاقة الائتمان المُعطى.
تحصيل المبلغ المستحق	تسمح للتاجر بطلب مبلغ الشراء من بوابة الدفع.
استفسار عن حالة طلب شهادة	إذا لم تتمكن CA من إكمال معالجة طلب الشهادة (certificate request) بسرعة، تقوم بإرسال رد لحامل البطاقة أو التاجر تشير بأن عليه مراجعتها مرة أخرى لاحقاً. بعد ذلك يقوم حامل البطاقة أو التاجر بإرسال استفسار (certificate inquiry) لتحديد حالة طلب الشهادة واستلامها إذا كان قد تم التصديق على الطلب.
استفسار عن شراء	تسمح لحامل البطاقة بالاستفسار عن حالة أمر شراء بعد استلام الرد على أمر الشراء. لاحظ أن هذه الرسالة لا تتضمّن معلومات مثلاً عن حالة السلع المتأخرة وقام التاجر بطلبها، لكنها تشير إلى حالة التفويض بالدفع وتحصيل المبلغ المستحق.
تصحيح تفويض الدفع	تسمح للتاجر بتصحيح طلبات التفويض السابقة. إذا كان من المتعذر إكمال الطلبية، يقوم التاجر بتصحيح التفويض بالكامل. أما إذا تعذّر إكمال جزء من الطلبية (مثلاً عند عدم توفر سلع مطلوبة مؤقتاً لدى التاجر ولكنه قام بطلبها)، يقوم التاجر بتصحيح ذلك الجزء وحده فقط.
تصحيح المبلغ المستحق	تسمح للتاجر بتصحيح الأخطاء في طلبات التحصيل (كقيمة المبلغ المستحق) التي قد تنجم عن خطأ في الإدخال من قِبَل الموظف.
تقييد مبلغ لحساب العميل	تسمح للتاجر بتقيد مبلغ لحساب حامل البطاقة، وذلك في حالات، منها: إرجاع بعض السلع أو تلفها أثناء الشحن. لاحظ أن رسالة SET الخاصة بذلك تبدأ دائماً من قِبَل التاجر وليس حامل البطاقة. تتم كافة الاتصالات بين حامل البطاقة والتاجر وهي تؤدي إلى إجراء معالجة لتقييد مبلغ لحساب حامل البطاقة خارج SET.
تصحيح مبلغ الاعتماد	تسمح للتاجر بتصحيح مبلغ الاعتماد سبق للتاجر أن طلب تقييده لحساب العميل.
طلب شهادة بوابة الدفع	تسمح للتاجر بطلب شهادات التوقيع واستلامها وتبادل المفتاح من بوابة الدفع
إدارة بالدفعات (batch administration)	تسمح للتاجر بإرسال معلومات عن عدة أوامر شراء تقدّم دفعة واحدة إلى بوابة الدفع.
رسالة خطأ	رسالة تدل على حدوث خطأ عند اختبار محتوى أو صيغة رسالة مُستلَمه.

❖ طلب الشراء:

قبل أن يبدأ تبادل طلب الشراء، يجب أن يكون حامل البطاقة قد انتهى من التصفح والاختيار والطلب. وتأتي هذه المرحلة التمهيدية إلى نهايتها عندما يرسل التاجر نموذج طلب شراء مكتمل إلى الزبون. وتتم كل المراسلات السابقة بدون استخدام SET.

يشتمل تبادل طلب الشراء على أربع رسائل: طلب استهلالي، ردّ استهلالي، طلب الشراء، ردّ الشراء.

لكي يرسل حامل البطاقة رسائل SET للتاجر، يجب أن يكون لديه نسخة من شهادة التاجر وشهادة بوابة الدفع. ويطلب الزبون الشهادتين في رسالة الطلب الاستهلالي التي تُرسَل للتاجر. وتتضمّن هذه الرسالة نوع بطاقة الائتمان التي يستخدمها الزبون، كما تتضمّن رقماً تعريفياً (ID) مخصصاً من قبل الزبون لهذا الزوج من الطلب والردّ، بالإضافة إلى nonce يُستخدم لضمان الوقتية (timeliness) للرسالة.

يولّد التاجر ردّه ويوقّعه بمفتاح توقيعه الخاص. يتضمّن الردّ الـ nonce الذي أرسله الزبون، وnonce آخر إلى الزبون ليرده في الرسالة التالية، ورقماً تعريفياً لمعاملة الشراء هذه. بالإضافة إلى الردّ الموقّع، تتضمّن رسالة الردّ الاستهلالي شهادة توقيع التاجر وشهادة تبادل المفاتيح لبوابة الدفع.

يتحقق حامل البطاقة من صحة تلك الشهادات الخاصة بالتاجر والبوابة عن طريق توقيعات سلطات الشهادات لهما، وبعدها يُنشيء OI وPI. يوضع الرقم التعريفي المخصص لتلك المعاملة من قِبَل التاجر في كلٍّ من OI وPI. لا يحتوي OI بشكلٍ صريح على بيانات الطلب كعدد كل بند من المواد وسعره، ولكنّه يتضمّن إشارةً للطلب تم الحصول عليها في التبادل بين التاجر والزبون أثناء مرحلة التسوق التي تمت قبل رسالة SET الأولى. بعد ذلك يُعِدّ حامل البطاقة رسالة طلب الشراء

(الشكل 7-10). لهذا الغرض، يولّد حامل البطاقة مفتاح تشفير متماثل للاستخدام مرة واحدة K_s.

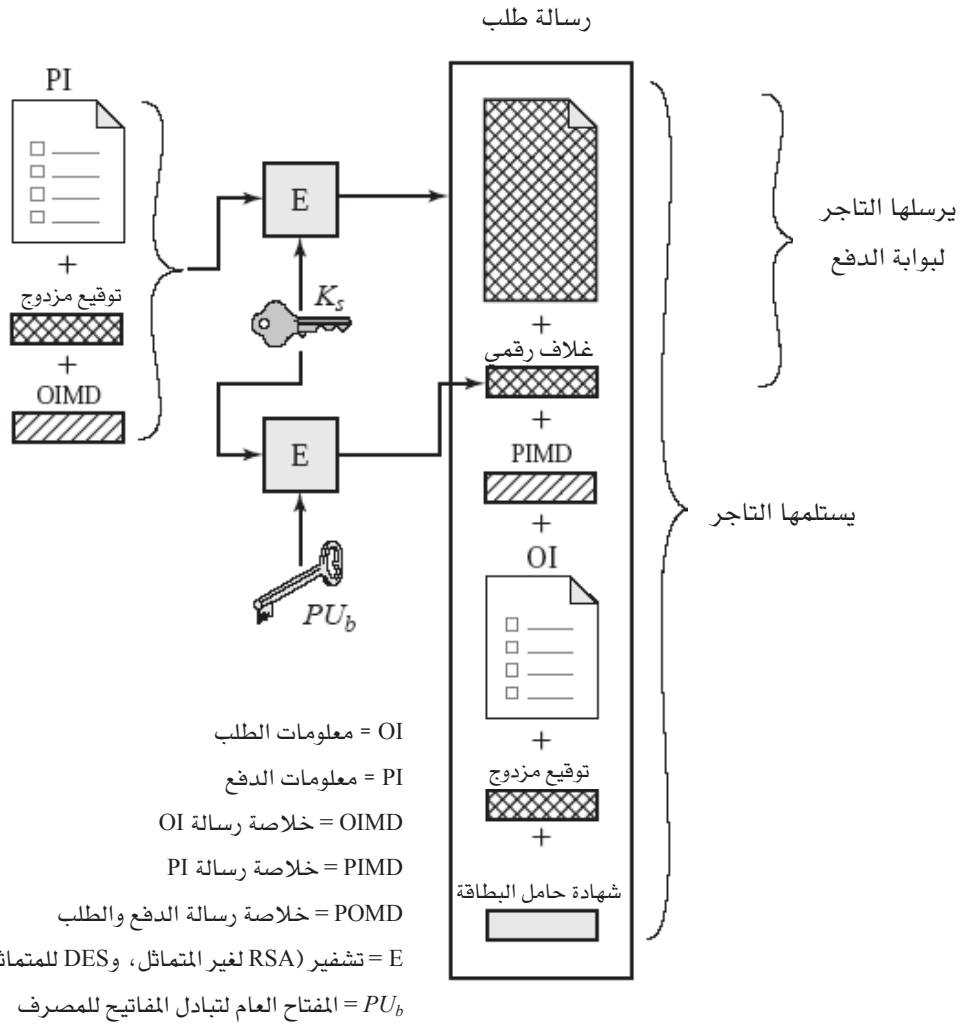
تتضمّن الرسالة ما يلي:

1. معلومات متعلقة بالشراء: يتم إرسال هذه المعلومات إلى بوابة الدفع بواسطة التاجر وتتكون من:
   - PI
   - التوقيع المزدوج: يُحسب من PI وOI ويوقّع بمفتاح توقيع الزبون الخاص
   - خلاصة رسالة OI (OIMD): وهي مطلوبة لبوابة الدفع للتحقق من صحة التوقيع المزدوج، كما أوضحنا آنفاً.

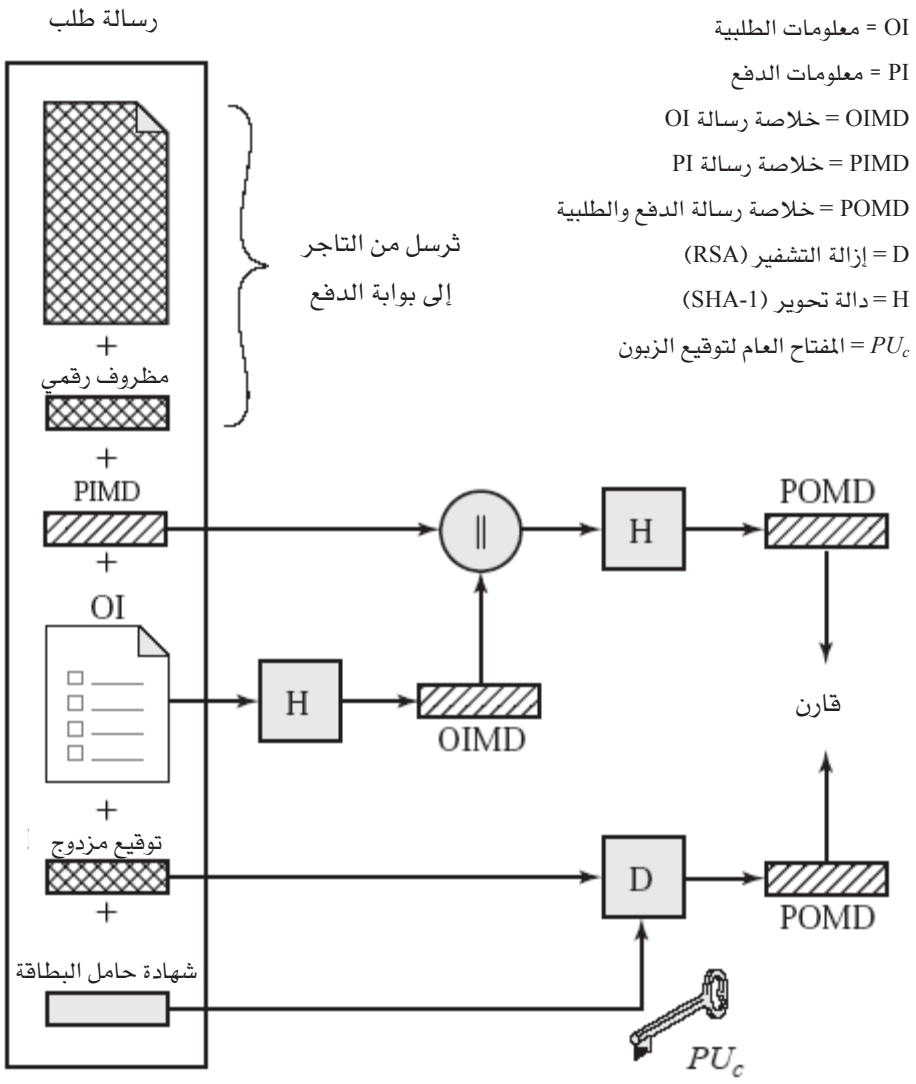
   يتم تشفير كل تلك العناصر بالمفتاح K_s. أما العنصر الأخير فهو:

   - الظرف الرقمي: ويتم حسابه بتشفير K_s بمفتاح تبادل المفاتيح العامة للبوابة. ويطلق عليه الظرف الرقمي لأنه يجب أن يُفتح (أي يُزال التشفير) قبل أن يتسنى قراءة أيٍّ من العناصر الأخرى السابقة.

   لاحظ أن قيمة K_s غير متوفرة للتاجر، ولذلك لا يستطيع التاجر قراءة أي من هذه المعلومات المتعلقة بالدفع.
2. معلومات متعلقة بالطلبية: هذه المعلومات مطلوبة للتاجر وتتضمّن:
   - OI: وتُرسَل بشكلٍ واضح (غير مُشفَّرة).
   - التوقيع المزدوج: ويحسب من PI وOI ويوقّع بمفتاح توقيع الزبون الخاص.
   - خلاصة رسالة PI (PIMD): وهي مطلوبة للتاجر لكي يتحقق من صحة التوقيع المزدوج.
3. شهادة حامل البطاقة: وتتضمّن مفتاح توقيع حامل البطاقة العام. وهي مطلوبة من قِبَل التاجر وبوابة الدفع.



الشكل 7-10: حامل بطاقة يرسل طلب شراء.



الشكل 7-11: تاجر يتحقق من طلب شراء من زبون.

عندما يستلم التاجر رسالة طلب الشراء، يقوم بالخطوات الآتية (الشكل 7-11):

1. يتحقق من شهادات حامل البطاقة عن طريق توقيع سلطة الشهادات لها.
2. يتحقق من التوقيع المزدوج باستخدام مفتاح توقيع الزبون العام. ويضمن ذلك أن الطلب لم يُعبث به أثناء النقل وأنه وُقِّع باستخدام مفتاح توقيع حامل البطاقة الخاص.
3. يعالِج الطلب ويرسل معلومات الدفع إلى بوابة الدفع للتصريح بالدفع (كما سنصف لاحقاً).
4. يرسل ردّ الشراء إلى حامل البطاقة.

تتضمّن رسالة ردّ الشراء كتلة ردّ للإشعار باستلام الطلب وإرجاع رقم تعريفي لتلك المعاملة. وهذه الكتلة موقَّعة من قِبَل التاجر باستخدام مفتاح توقيعه الخاص. وتُرسَل الكتلة وتوقيعها للزبون معاً مع شهادة توقيع التاجر.

عندما تتلقى برمجيات حامل البطاقة رسالة ردّ الشراء، تقوم بالتحقق من شهادة التاجر ثم تتحقق من توقيع كتلة الردّ. وأخيراً، تقوم البرمجيات بفعلٍ ما يعتمد على الردّ؛ كعرض رسالة للمستخدم أو تحديث قاعدة البيانات بحالة الطلبية.

❖ تفويض الدفع:

أثناء معالجة طلبٍ من حامل بطاقة، يُرخّص للتاجر بالمعاملة من بوّابة الدفع. ويضمن ترخيص الدفع أن المعاملة مسموحٌ بها من قِبَل المُصدِر. ويضمن هذا التصريح حصول التاجر على المبلغ المُستحق؛ لذا يمكن أن يوفر التاجر الخدمات أو السلع للزبون. ويشمل تبادل تفويض الدفع رسالتين: طلب التفويض وردّ التفويض.

يرسل التاجر رسالة طلب تفويض لبوابة الدفع تشمل ما يأتي:

1. معلومات متعلقة بالشراء: تم الحصول على هذه المعلومات من الزبون وتتكون من:
   - PI
   - التوقيع المزدوج: يُحسب من PI وOI ويوقَّع بمفتاح توقيع الزبون الخاص
   - خلاصة رسالة OI (OIMD)
   - الظرف الرقمي
2. معلومات متعلقة بالتفويض: هذه المعلومات يتم توليدها لدى التاجر وتتكون من:
   - كتلة التفويض: وتتضمّن الرقم التعريفي للمعاملة موقّعاً بمفتاح توقيع التاجر الخاص ومُشفّراً بمفتاح متماثل مولَّد من قِبَل التاجر للاستخدام لمرة واحدة .
   - ظرف رقمي: يتم إنشاؤه بتشفير مفتاح الاستخدام لمرة واحدة بمفتاح تبادل المفاتيح العامة لبوابة الدفع.
3. الشهادات: يُضمِّن التاجر شهادة مفتاح التوقيع لحامل البطاقة (تُستخدم للتحقق من التوقيع المزدوج)، وشهادة مفتاح توقيع التاجر (تُستخدم للتحقق من توقيع التاجر)، وشهادة تبادل المفاتيح للتاجر (مطلوبة في ردّ بوابة الدفع).

تقوم بوابة الدفع بالمهام الآتية:

1. التحقق من كل الشهادات.
2. إزالة تشفير الظرف الرقمي لكتلة التفويض للحصول على المفتاح المتماثل وبعد ذلك إزالة تشفير كتلة التفويض.
3. التحقق من توقيع التاجر على كتلة التفويض.

4. إزالة تشفير الظرف الرقمي لكتلة الدفع للحصول على المفتاح المتماثل وبعد ذلك إزالة تشفير كتلة الدفع.
5. التحقق من التوقيع المزدوج على كتلة الدفع.
6. التحقق من أن الرقم التعريفي للمعاملة الذي يُستلم من التاجر يطابق الموجود في PI القادم (بشكلٍ غير مباشر) من الزبون.
7. تطلب وتستلم تفويضاً بالدفع من المُصدِر.

بعد أن حصلت بوابة الدفع على التفويض من المُصدِر، ترسل رسالة ردّ التفويض للتاجر تتضمّن العناصر الآتية:

1. معلومات متعلقة بالتفويض: تتضمّن كتلة التفويض موقَّعةً بمفتاح توقيع البوابة الخاص ومُشفَّرةً بمفتاح متماثل يُستخدم لمرة واحدة ومولَّد بالبوّابة. وتتضمّن أيضاً ظرفاً رقمياً يشتمل على المفتاح الذي يُستخدَم لمرة واحدة مُشفَّراً بمفتاح التاجر لتبادل المفاتيح العامة.
2. معلومات توكن التحصيل (capture token): تُستخدم هذه المعلومات لإنجاز الدفع لاحقاً. ولهذه الكتلة نفس شكل المعلومات في (1) (أي رسالة توكن موقَّعة ومُشفَّرة مع مظروف رقمي). ولا يُعالج هذا التوكن من قِبَل التاجر. ولكن ينبغي إعادته كما هو مع طلب الدفع.
3. الشهادة: شهادة مفتاح توقيع البوابة.

بهذا التفويض من البوّابة، يمكن أن يورّد التاجر السلع أو يوفر الخدمة للزبون.

❖ تحصيل المبلغ المستحق:

لكي يحصل التاجر على المبلغ المستحق، يطلب من بوابة الدفع القيام بمُعامَلة تحصيل تتضمّن طلب التحصيل ورسالة ردّ التحصيل.

لإنشاء رسالة طلب التحصيل، يولِّد التاجر ويُوقِّع ويُشفِّر كتلة طلب التحصيل التي تتضمّن قيمة الدفع ورقم تعريف المُعامَلة. وتتضمّن الرسالة أيضاً توكن التحصيل مُشفَّراً، وقد تم استلامه في وقت سابق (ضمن ردّ التفويض) لهذه المُعامَلة، بالإضافة إلى شهادات مفتاح توقيع التاجر ومفتاح تبادل المفاتيح.

عندما تتلقّى بوابة الدفع رسالة طلب التحصيل، تقوم بإزالة تشفير كتلة طلب التحصيل والتحقق منها، وإزالة تشفير كتلة توكن التحصيل والتحقق منها. بعد ذلك تقوم بفحص الاتساق (consistency) بين طلب التحصيل وتوكن التحصيل، ثم تقوم بإنشاء طلب تسديد يُرسَل للمُصدِر على شبكة الدفع الخاصة. ويترتب على هذا الطلب تحويل المبلغ إلى حساب التاجر.

عندئذ تُخطِر البوابة التاجر بالدفع في رسالة ردّ التحصيل. وتتضمّن الرسالة كتلة ردّ التحصيل موقَّعة ومشفَّرة من قِبَل البوابة. وتتضمّن الرسالة أيضاً شهادة مفتاح توقيع البوابة. وتخزّن برمجيات التاجر ردّ التحصيل لاستخدامه لمطابقة المبلغ المدفوع الذي تم استلامه من المُحصِّل.

4-7 توصيات للمطالعة

[DREW99] Drew, G. *Using SET for Secure Electronic Commerce.* Upper Saddle River, NJ: Prentice Hall, 1999.

[MACG97] Macgregor, R.; Ezvan, C.; Liguori, L.; and Han, J. *Secure Electronic Transactions: Credit Card Payment on the Web in Theory and Practice.* IBM RedBook SG24-4978-00, 1997. Available at www.redbooks.ibm.com.

[RESC01] Rescorla, E. *SSL and TLS: Designing and Building Secure Systems.* Reading, MA: Addison-Wesley, 2001.

5-7 مصادر للمعلومات على الويب

- صفحة SSL لدى شركة نيتسكيب: وتحتوي على مواصفات SSL.
- ميثاق أمن طبقة النقل (TLS Charter): يتضمّن أحدث مسوّدات الإنترنت (Internet drafts) وطلبات التعليق (RFCs) الخاصة ببروتوكول TLS.
- مشروع OpenSSL: وهو مشروع لتطوير برمجيات SSL وTLS مفتوحة المصدر. يتضمن الموقع مستندات وروابط لمواقع أخرى.

6-7 مصطلحات رئيسة

acquirer	مُحصِّل
cardholder	حامل البطاقة
Certification Authority (CA)	سلطة التوثيق (التصديق)
dual signature	توقيع مزدوج
issuer	مُصدِر
merchant	تاجر
payment gateway	بوابة الدفع
Secure Electronic Transaction (SET)	المُعامَلات الإلكترونية الآمنة (SET)
Secure Socket Layer (SSL)	طبقة المقابس الآمنة (SSL)
Transport Layer Security (TLS)	أمن طبقة النقل (TLS)

7-7 أسئلة للمراجعة ومسائل

7-7-1 أسئلة للمراجعة

7-1 ما مميزات كلٍّ من الطرق الثلاثة الموضحة بالشكل 7-1؟

7-2 اذكر البروتوكولات التي يتكون منها SSL.

7-3 ما الفرق بين توصيلة SSL وجلسة SSL؟

7-4 اذكر وعرِّف باختصار المُعامِلات التي تُعرِّف حالة جلسة SSL.

7-5 اذكر وعرِّف باختصار المُعامِلات التي تُعرِّف حالة توصيلة SSL.

7-6 اذكر الخدمات التي يوفرها بروتوكول سجل SSL.

7-7 اذكر خطوات النقل في بروتوكول سجل SSL.

7-8 اذكر وعرّف باختصار الفئات الرئيسة للأطراف المشتركة في SET.

7-9 ما المقصود بالتوقيع المزدوج وما الغرض منه؟

7-7-2 مسائل

7-1 في SSL وTLS لماذا يوجد بروتوكول منفصل لتغيير مواصفات التشفير بدلاً من تضمين رسالة تغيير مواصفات التشفير (change_cipher_spec) في بروتوكول المصافحة؟

7-2 لكلٍّ من التهديدات الآتية لأمن الويب، صف كيفية المواجهة باستخدام خاصية معينة في SSL:

a. الهجوم الاستقصائي على التشفير: البحث الشامل لفضاء المفتاح لخوارزمية تشفير تقليدية.

b. هجوم القاموس بمعرفة نص أصلي غير مُشفَّر: تحتوي كثير من الرسائل على نصوص أصلية يمكن توقعها مثل أمر GET في بروتوكول HTTP. وينشئ المهاجم قاموساً يتضمن كل التشفير المحتملة برسالة النص الأصلي المعروف. وعندما يعترض المهاجم

رسالة مُشفَّرة، يأخذ الجزء الذي يحتوي النص الأصلي المعروف المُشفَّر ويبحث في القاموس عن النص المُشفَّر المناظر. ويجب أن يتوافق النص المُشفَّر مع أحد المُدخَلات بالقاموس تم تشفيره بنفس المفتاح السري. إذا حدث توافق مع عدة مُدخلات، فيمكن تجريب كلٍّ منها لتحديد النص الأصلي الكامل للنص المُشفَّر. وهذا النوع من الهجوم فعّال، خصوصاً عند استخدام مفاتيح ذات أطوال قصيرة (مثلاً بطول 40 بتاً).

c. هجوم إعادة التشغيل: وفيه يتم إعادة إرسال رسائل مصافحة SSL السابقة.
d. هجوم رجل في الوسط: يتدخل المهاجم أثناء تبادل المفاتيح، ويتصرف كالزبون مع الخادم وكالخادم مع الزبون.
e. التقاط كلمة السر: يتم التنصت على حركة مرور بيانات HTTP أو أي تطبيق آخر لمعرفة كلمات السر.
f. تزييف عنوان IP: وفيه يتم استخدام عناوين IP مزيّفة لخداع المضيف لقبول بيانات ليست حقيقة.
g. اختطاف عنوان IP: وفيه يقوم المهاجم بعرقلة اتصال موثق نشيط بين مضيّفين، وأخْذ مكان أحدهما للاتصال بالآخر.
h. فيضان SYN: يرسل المهاجم رسائل SYN في بروتوكول TCP لطلب إنشاء اتصال، ولكنه لا يرد على الرسالة النهائية لإتمام عملية تأسيس الاتصال. وعادةً ما تترك وحدة TCP المهاجِمة التوصيلة نصف مفتوحة لمدّة بضع دقائق. ويمكن أن يؤدي تكرار إرسال رسائل SYN إلى إعاقة TCP عن العمل.

3-7 بناءً على ما تعلمته في هذا الفصل، هل من الممكن للمُستلِم في SSL أن يعيد ترتيب كتل سجل SSL التي تصله بترتيب مختلف عما أُرسِلت به؟ إذا كان الأمر كذلك، فوضِّح كيف يحدث ذلك. وإلا فاذكر سبب عدم إمكانية ذلك.

8

الفصل الثامن

أمن إدارة الشبكات

محتويات الفصل:

8-1 المفاهيم الأساسية لبروتوكول إدارة الشبكات البسيط (SNMP)

8-1-1 البنية المعمارية لإدارة الشبكات

8-1-2 البنية المعمارية لبروتوكول إدارة الشبكات البسيط

8-1-3 المُفوَّضات (Proxies)

8-1-4 الإصدار الثاني لبروتوكول إدارة الشبكات البسيط

8-2 تسهيلات تجمعات SNMPv1

8-2-1 التجمعات وأسماؤها (Communities and Community Names)

8-2-2 خدمة التوثيق (Authentication Service)

8-2-3 سياسة الوصول (Access Policy)

8-2-4 خدمة المُفوَّض (Proxy Service)

8-3 الإصدار الثالث لبروتوكول إدارة الشبكات البسيط

8-3-1 بنية SNMP المعمارية

8-3-2 تمرير الرسائل ونموذج أمن المُستخدِم

8-3-3 التحكم في الوصول المبنيّ على المشهد

8-4 توصيات للمطالعة

8-5 مصادر للمعلومات على الويب

8-6 مصطلحات رئيسة

8-7 أسئلة للمراجعة ومسائل

"إن قيادة كتيبة كبيرة تماثل – من حيث المبدأ – قيادة عدد قليل من الرجال، فهي مجرد مسألة وضع أسس للعلامات والإشارات" – من كتاب "فن الحرب" ، لصن تزو.

لنظم المعالجة الموزَّعة وللشبكات أهمية كبيرة ومتزايدة لدى المنظمات والمؤسسات التجارية والهيئات الحكومية. وحالياً تتجه الأمور داخل المؤسسات نحو استخدام شبكات أكبر وأكثر تعقيداً لدعم مزيدٍ من التطبيقات وعددٍ أكبر من المُستخدِمين؛ ومع النمو المستمر لحجم تلك الشبكات، تتضح حقيقتان مهمتان:

- أصبحت الشبكة ومواردها الملحقة بها وما يرتبط بها من تطبيقات موزَّعة، لا غنى عنها للمؤسسات.
- مع زيادة تعقيد الشبكة زاد احتمال حدوث أعطال بها أو بجزء منها أو تدني أداء الشبكة إلى مستوى غير مقبول.

لذا أصبح من غير الممكن تأسيس شبكات ضخمة وإدارتها بالجهد البشري وحده، وتطلب الأمر استخدام أدوات آلية (مؤتمتة) لإدارة الشبكة. كما زادت الحاجة إلى مثل تلك الأدوات وزادت صعوبة توفيرها للشبكات التي تضم أجهزة وبرمجيات من شركات متعددة. ولتلبية هذه الحاجة، تم تطوير معايير (standards) – للتعامل مع إدارة الشبكة – تشمل الخدمات والبروتوكولات وقواعد معلومات الإدارة.

المعيار الأكثر استخداماً في هذا المجال هو بروتوكول إدارة الشبكات البسيط ((Simple Network Management Protocol (SNMP). ومنذ نشر مواصفاته في عام 1988، تم استخدام بروتوكول SNMP في أعدادٍ متزايدة من الشبكات وفي بيئات متزايدة التعقيد. ومع زيادة انتشاره ظهرت الحاجة إلى إضافة قدرات وظيفية جديدة للبروتوكول لاستيعاب المتطلبات الجديدة. كما اتضح أيضاً وبصورة متزايدة أهمية توفير قدرات أمنية كمكوِّن أساسي من مكوِّنات إدارة الشبكة. وتضمَّن

الإصدار الثاني من SNMP وظائف محسنَّة.[1] أما التحسينات الأمنية فقد جاءت في الإصدار الثالث (SNMPv3).

يبدأ هذا الفصل بوصف الإمكانات الأمنية الأوَّليَّة المتاحة في SNMPv1، ثم يناقش السمات الأمنية الأكثر تطوراً المتوفِّرة في الإصدار الثالث.

8-1 المفاهيم الأساسية لبروتوكول إدارة الشبكات البسيط (SNMP)

يقدم هذا الجزء لمحة عامة عن الإطار الأساسي لبروتوكول SNMP.

8-1-1 البنية المعمارية لإدارة الشبكات

يتألف نظام إدارة الشبكة من مجموعة من أدوات الرصد والمراقبة والتحكم في الشبكة، والمتوفرة بشكلٍ شبه مُجمَّع يتمثل في:

- وجود واجهة تشغيل واحدة مع مجموعة من الأوامر الفعَّالة سهلة الاستخدام لأداء معظم مهام إدارة الشبكة إن لم يكن جميعها.
- وجود أقل عدد من الأجهزة المنفصلة. بمعنى أن تكون معظم المكوِّنات المادية والبرمجيات اللازمة لإدارة الشبكة مُتضمَّنة في جهاز المُستخدِم.

يتألَّف نظام إدارة الشبكة من إضافات متزايدة من الأدوات والبرمجيات التي تعمل بين المكونات الموجودة بالشبكة. وتوجد البرامج المُستخدَمة في إنجاز مهام إدارة الشبكة بأنظمة الحاسبات المضيفة وبمعالجات الاتصالات (على سبيل المثال معالجات المقدمة (front-end processors)، ووحدات تحكم العناقيد الطرفية (terminal cluster controllers). تم تصميم نظام إدارة الشبكة لرؤية الشبكة بأكملها كبنية معمارية موحدة، بما في ذلك العناوين والتسميات المخصصة لكل نقطة والسمات المحدّدة لكل العناصر والوصلات المعروفة للنظام. وتوفر العناصر

[1] يشار إلى الإصدار الثاني بـ SNMPv2 لتمييز الإصدار الجديد عن الإصدار الأصلي الذي يشار إليه الآن عموماً بـ SNMPv1.

النشطة في الشبكة تغذية مرتدَّة منتظمة عن معلومات الحالة إلى مركز التحكم بالشبكة.

يتضمن نموذج إدارة الشبكة المُستخدَم في SNMP العناصر الرئيسة التالية:

- محطة الإدارة.
- وكيل الإدارة.
- قاعدة معلومات الإدارة.
- بروتوكول إدارة الشبكة.

عادةً ما تكون محطة الإدارة جهازاً مستقلاً بذاته أو قد تكون إمكانية ضمن نظام مشترك (shared system). وفي كلتا الحالتين، تُعد محطة الإدارة بمثابة الواجهة للشخص المسؤول عن إدارة الشبكة للتعامل مع نظام إدارة الشبكة. وكحد أدنى ينبغي توفر الآتي لمحطة الإدارة:

- مجموعة من تطبيقات الإدارة لتحليل البيانات، والتعافي من الأخطاء، إلخ.
- واجهة يستطيع من خلالها مدير الشبكة مراقبة الشبكة والتحكم فيها.
- إمكانية ترجمة طلبات مدير الشبكة إلى أوامر فعلية لمراقبة العناصر البعيدة في الشبكة والتحكم فيها.
- قاعدة بيانات للمعلومات المستخلصة من قواعد معلومات الإدارة MIBs لكل الكيانات المُتحكَّم فيها من قِبَل نظام إدارة الشبكة.

ويستهدف التوحيد القياسي لـ SNMP العنصرين الأخيرين فقط.

ويمثل "وكيل الإدارة" العنصر الآخر الفعَّال في نظام إدارة الشبكة. وقد تُجهَّز المنصَّات الأساسية – مثل الأنظمة المضيفة، والجسور (bridges)، والموجّهات (Routers)، والمُجمِّعات (Hubs) – ببروتوكول SNMP حتى يمكن إدارتها من خلال محطة الإدارة. ويقوم وكيل الإدارة بالاستجابة لطلبات المعلومات أو طلبات القيام بعمل معيّن الواردة من أيٍّ من محطات الإدارة، كما يمكنه بشكلٍ غير متزامن إمداد محطة الإدارة بمعلومات مهمة تلقائياً وبدون طلب منها.

ولإدارة الموارد في الشبكة، يتم تمثيل كل مورد ككائن. والكائن أساساً، هو متغير بيانات يمثل سمة واحدة للوكيل المُدار. ويُشار إلى مجموعة الكائنات بـ "قاعدة معلومات الإدارة" ((Management Information Base (MIB). وتعمل MIB كمجموعة من نقاط الوصول (النفاذ) لدى الوكيل تستخدمها محطة الإدارة. وقد تم توحيد هذه الكائنات بطريقة معيارية عبر الأنظمة المختلفة حسب فئتها (مثلاً: تقوم جميع الجسور بدعم نفس كائنات الإدارة). تؤدي محطة الإدارة وظيفة المراقبة عن طريق استرجاع قيم كائنات MIB. كما يمكن أن تتسبب محطة الإدارة في القيام بإجراءٍ محدّد لدى وكيل معين، وتستطيع تغيير إعدادات التهيئة لوكيلٍ ما بتعديل قيم كائنات معيّنة.

يتم ربط محطة الإدارة ووكلائها عن طريق بروتوكول إدارة الشبكة (network management protocol). فالبروتوكول المُستخدَم لإدارة شبكات TCP/IP هو بروتوكول إدارة الشبكة البسيط (SNMP). ويتضمن هذا البروتوكول الإمكانات الرئيسة الآتية:

- جلب (Get): تُمكّن محطة الإدارة من الحصول على قيم الكائنات لدى الوكيل.
- تعيين (Set): تُمكّن محطة الإدارة من تعديل قيم الكائنات لدى الوكيل.
- إخطار (Notify): تُمكّن الوكيل من إخطار محطة الإدارة بالأحداث المهمة.

8-1-2 البنية المعمارية لبروتوكول إدارة الشبكات

صدرت مواصفات SNMP في عام 1988، وسرعان ما أصبحت المعيار القياسي المهيمن على إدارة الشبكات. وتوفر بعض الشركات أنظمة محطات عمل قائمة بذاتها لإدارة الشبكات مبنيَّة على بروتوكول SNMP. كما يوفِّر معظم مصنعي الأجهزة - كالموجّهات (Routers)، والجسور (bridges)، ومحطات العمل، وأجهزة الحاسب - حزماً لوكيل بروتوكول SNMP مما يسمح بإدارة منتجاتهم بواسطة محطة إدارة SNMP.

وكما يشير الاسم، يعدّ بروتوكول SNMP أداة بسيطة لإدارة الشبكة. حيث يُعرِّف قاعدة محدودة وسهلة التنفيذ لمعلومات الإدارة (MIB) تتألَّف من متغيرات قياسية (scalar variables) وجداول ثنائية الأبعاد، كما يُعرِّف أسلوباً بسيطاً لتمكين المدير من الحصول على قيم متغيرات MIB وتعديلها، وتمكين وكيلٍ ما من إرسال إخطارات بشكلٍ تلقائي (unsolicited) (تُعرَف بـ traps). وهذه البساطة هي أساس القوة لبروتوكول SNMP. علاوةً على ذلك، فإن بروتوكول SNMP سهل التنفيذ واستهلاكه متواضع لموارد الشبكة والمعالج. كما أن بنية البروتوكول وقواعد معلومات الإدارة MIB واضحة بما فيه الكفاية وليس من الصعب تحقيق التوافق للعمل البيني (interoperability) لمنتجات من شركات مختلفة لمحطات الإدارة وبرمجيات الوكلاء.

المواصفات الثلاثة الأساسية هي:

- هيكل معلومات الإدارة للشبكات المبنيَّة على مجموعة بروتوكولات TCP/IP (RFC 1155): يصف كيفية تعريف الكائنات المُدارة المضمَّنة في قاعدة معلومات الإدارة MIB.
- قاعدة معلومات الإدارة لإدارة الشبكات البينية التي تعتمد على بروتوكولات TCP/IP: MIB-II (RFC 1213)؛ يصف الكائنات المُدارة المُضمَّنة في MIB.
- بروتوكول إدارة الشبكات البسيط (RFC 1157): يُعرِّف البروتوكول المُستخدَم لإدارة هذه الكائنات.

لقد صُمِّم SNMP ليكون بروتوكولاً ضمن طبقة التطبيقات في مجموعة بروتوكولات TCP/IP، حيث يعمل من خلال بروتوكول وحدة بيانات المُستخدِم (UDP)، وقد تم تعريفه في طلب التعليقات رقم RFC 768. وبالنسبة لمحطة إدارة قائمةً بذاتها، تتحكم عملية الإدارة (manager process) في الوصول إلى قاعدة معلومات الإدارة المركزية (Central MIB) بمحطة الإدارة كما توفر واجهة لمدير الشبكة. وتنجز عملية الإدارة تلك مهام إدارة الشبكة باستخدام SNMP المبنيّ على

بروتوكول بيانات المَستخدِم (UDP) وبروتوكول الإنترنت (IP) والبروتوكولات ذات الصلة المُعتمِدة على الشبكة (مثل Ethernet، FDDI، X.25).

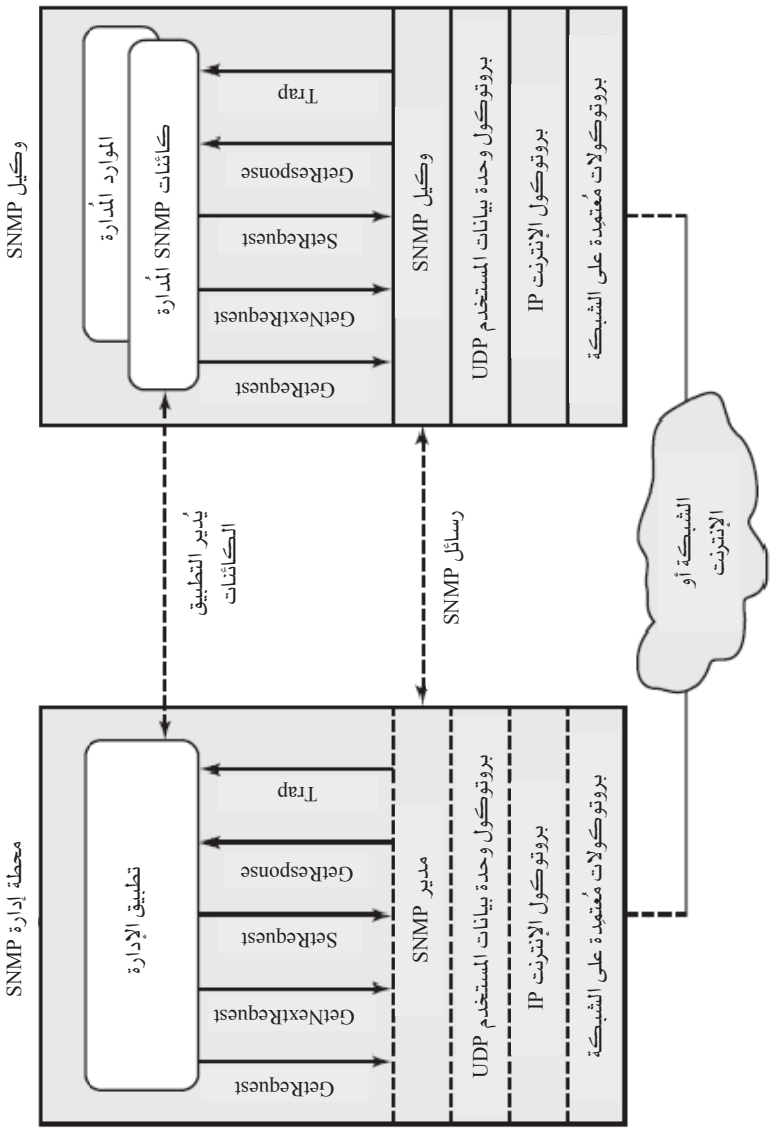
يتعين أيضاً على كل وكيل تنفيذ بروتوكولات SNMP وUDP وIP. وبالإضافة إلى ذلك، هناك عملية وكيل (agent process) تقوم بتفسير رسائل SNMP وبالتحكم في قاعدة معلومات الإدارة لديه. يحتاج جهاز وكيل يدعم تطبيقات أخرى مثل FTP إلى كلٍّ من بروتوكولي TCP وUDP.

يوضّح الشكل 8-1 سياق بروتوكول SNMP. تصدر محطة الإدارة ثلاثة أنواع من رسائل SNMP نيابةً عن تطبيقات الإدارة؛ وهي: GetRequest، وGetNextRequest، وSetRequest. والرسالتان الأوليان هما شكلان متباينان من Get. يُرَد على تلك الرسائل الثلاث من قِبَل الوكيل في شكل رسالة GetResponse، والتي يتم تمريرها إلى تطبيق الإدارة. وبالإضافة إلى ذلك، يمكن أن يُصدِر الوكيل رسالة إخطار تلقائي (trap) استجابةً لحدثٍ ما يؤثر على MIB والموارد التحتية المُدارة.

نظراً لأن SNMP يعتمد على UDP – وهو بروتوكول لاتوصيلي – فإن SNMP نفسه بروتوكول لاتوصيلي. فلا تتم المحافظة على الوصلات الجارية بين محطة الإدارة ووكلائها، وإنما يُعدُّ كل تبادل مُعامَلة منفصلة بين محطة الإدارة والوكيل.

8-1-3 المُفوَّضات (Proxies)

في الإصدار الأول من بروتوكول SNMP يجب أن تدعم كل محطات الإدارة والوكلاء (agents) بروتوكولات UDP وIP. وهذا يُقصِر الإدارة المباشرة على مثل تلك الأجهزة ويستبعد الأجهزة الأخرى كبعض الجسور وأجهزة المودم التي لا تدعم أيًّا من بروتوكولات مجموعة TCP/IP. وعلاوةً على ذلك، قد توجد نظم عديدة صغيرة (مثل أجهزة الحاسبات الشخصية، ومحطات العمل، ووحدات التحكم القابلة للبرمجة) التي تُنفِّذ فعلاً مجموعة TCP/IP من أجل دعم تطبيقاتها، لكنها لا ترغب في تحمل العبء الإضافي لبروتوكول SNMP أو لعمليات الوكيل أو لصيانة قاعدة معلومات الإدارة (MIB).



الشكل 8-1: دور/ وظيفة SNMP.

ولاستيعاب الأجهزة التي لا تُنفِّذ بروتوكول SNMP تم تطوير مفهوم المُفوَّض (Proxy). وفي هذا النظام يعدّ وكيل SNMP بمثابة مُفوَّض (Proxy) لواحدٍ أو أكثر من الأجهزة الأخرى؛ بمعنى آخر يتصرف وكيل SNMP نيابةً عن تلك الأجهزة المُفوِّضة (proxied devices).

يبيّن الشكل 8-2 النوع المُستخدَم في الغالب لبنية البروتوكول. حيث تُرسِل محطة الإدارة استفسارات عن جهاز معين إلى الوكيل المُفوَّض، فيقوم الوكيل المُفوَّض بتحوّيل كل استفسار إلى بروتوكول الإدارة الذي يستخدمه الجهاز المعني. وعندما يتلقى الوكيل رداً على استفسارٍ ما، يقوم بتمريره إلى محطة الإدارة. وبالمثل، إذا تم إرسال إبلاغ عن حدثٍ من أي نوع إلى المُفوَّض، يقوم المُفوَّض بدوره بإرساله إلى محطة الإدارة في شكل رسالة إخطار تلقائي.

يُتيح الإصدار الثاني من بروتوكول SNMP ليس فقط استخدام مجموعة بروتوكولات TCP/IP بل يسمح بغيرها كذلك. ويستهدف SNMPv2 تحديداً العمل على سلة بروتوكولات الأنظمة المفتوحة (OSI). ومن ثَمَّ، يمكن استخدام SNMPv2 لإدارة تشكيلة أكبر من التكوينات الشبكية. وفيما يتعلق بالمُفوَّضات، فإن الأجهزة التي لا تدعم تنفيذ SNMPv2 يمكن إدارتها فقط بواسطة المُفوَّض. ويشمل ذلك حتى الأجهزة التي تدعم SNMPv1؛ أي أنه في حالة جهاز يُنفِّذ برنامج الوكيل لبروتوكول SNMPv1، يمكن الوصول إليه من مدير SNMPv2 فقط عن طريق جهاز مُفوَّض يُنفِّذ وكيل SNMPv2 ومدير SNMPv1.

في بروتوكول SNMPv2 تُعرَف الحالات المُبيَّنة في الفقرة السابقة بعلاقات المُفوَّض الخارجي (foreign proxy relationships). وبالإضافة إلى ذلك، يدعم SNMPv2 علاقات المُفوَّضين المحليين التي يدعم فيها الجهاز المُفوِّض بروتوكول SNMPv2. وفي هذه الحالة، يتصل مدير SNMPv2 بوكيل SNMPv2 والذي يقوم بدور المدير للوصول إلى الجهاز المُفوِّض الذي يقوم بدور وكيل SNMPv2. والهدف من دعم هذا النوع من التعامل غير المباشر هو تمكين المُستخدِمين من تهيئة أنظمة شبكات هرمية لامركزية الإدارة، كما سنناقش لاحقاً.

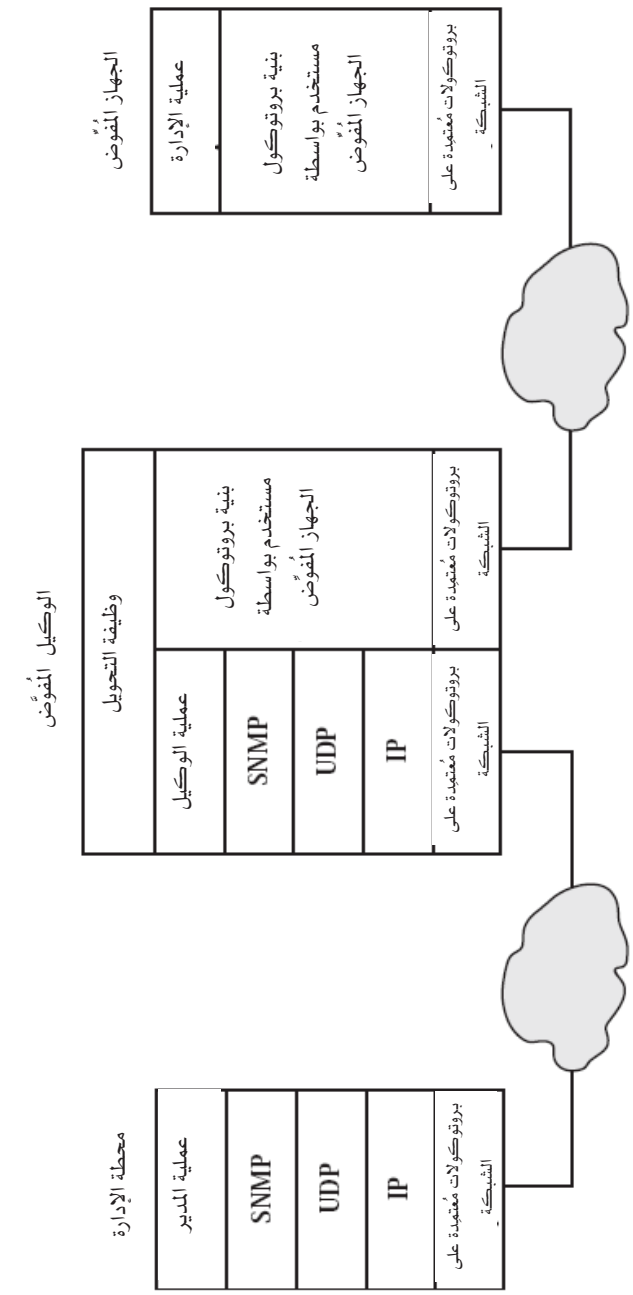
8-1-4 الإصدار الثاني لبروتوكول إدارة الشبكات البسيط (SNMPv2)

تكمن قوة بروتوكول SNMP في بساطته، حيث يوفر مجموعة أساسية من أدوات إدارة الشبكة في حزمة سهلة التنفيذ وسهلة التهيئة. ومع زيادة اعتماد المُستخدِمين أكثر وأكثر على SNMP لإدارة الشبكات والتي بدورها تزداد توسعاً وتزداد حمولتها، بدأت تتضح أوجه القصور في هذا البروتوكول، ويمكن تقسيمها إلى ثلاث فئات:

- الافتقار إلى الدعم المطلوب لإدارة الشبكات الموزَّعة.
- أوجه القصور الوظيفية.
- القصور الأمني.

تمت معالجة الفئتين الأوليين من أوجه القصور في SNMPv2، والذي صدر في عام 1993، ثم صدرت نسخته المنقّحة (RFCs 1901, 1904 ... 1908, 2578, 2579) في عام 1996. وسرعان ما اكتسب SNMPv2 الدعم، وأُعلِنت المنتجات من عدد من الشركات خلال أشهر من إصداره. أما أوجه القصور الأمني فقد تمت معالجتها في الإصدار الثالث للبروتوكول (SNMPv3).

سنلخص بإيجاز فيما تبقى من هذا الجزء السمات الجديدة المتوفرة في SNMPv2. وبعد ذلك سنقوم بفحص سمات الأمن لكلٍّ من SNMPv1 وSNMPv3 بالتفصيل في الجزأين 8-2 و8-3 على التوالي.



الشكل 8-2: تهيئة المُفوَّض.

❖ إدارة الشبكات الموزَّعة

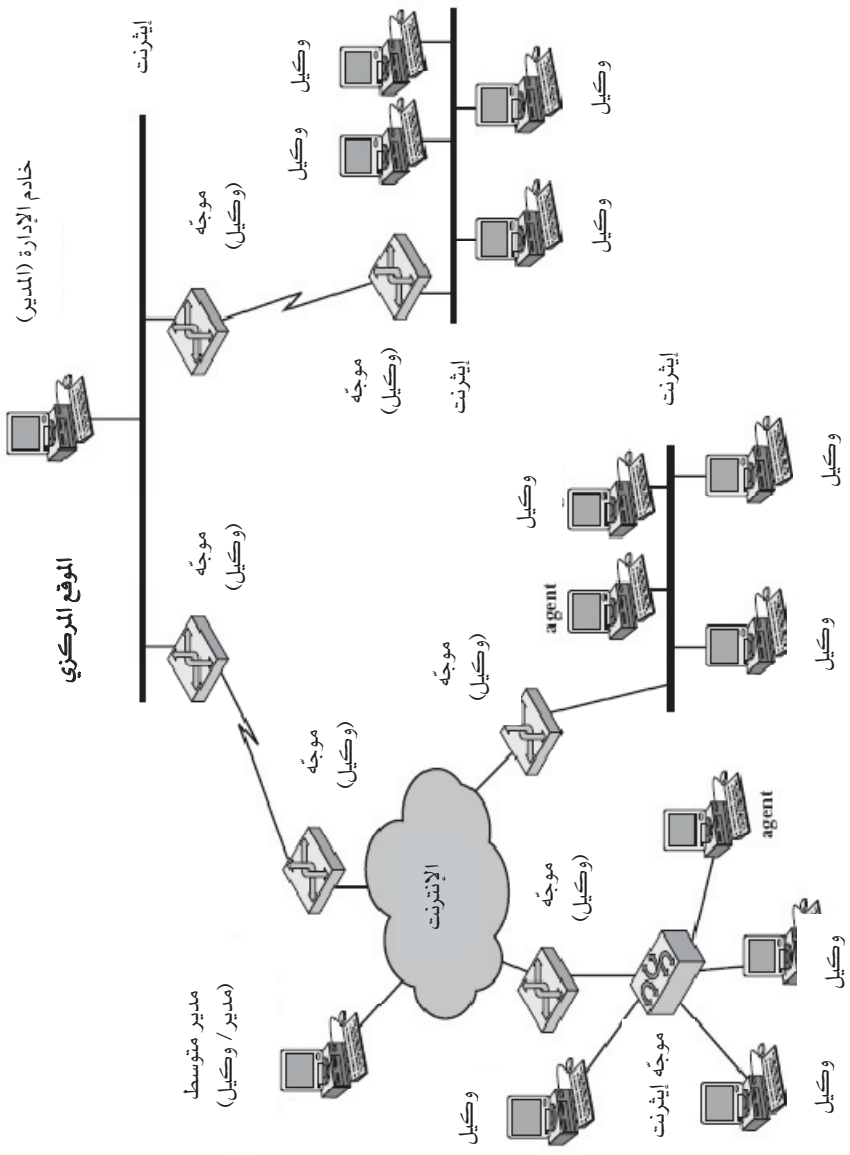
في النظم التقليدية للإدارة المركزية للشبكات، يقوم نظام مضيف واحد بدور محطة إدارة الشبكة؛ وقد تكون هناك محطة إدارة احتياطية أو اثنتان. أما ما تبقى من أجهزة الشبكة فيحتوي على برامج الوكلاء (agent software) وقاعدة معلومات الإدارة (MIB)، وذلك للسماح بالمراقبة والتحكم من محطة الإدارة. ونظراً لنمو حجم الشبكات وزيادة حركة المرور بها، أصبح مثل هذا النظام المركزي غير قابل للتطبيق عملياً. يرجع ذلك إلى العبء الكبير الذي تتحمله محطة الإدارة، إضافةً إلى حركة المرور العالية من التقارير الواردة من كل وكيل على حِدَة والتي يجب أن تشق طريقها إلى المقر الرئيس عبر الشبكة بأكملها. في مثل هذه الظروف، تعمل الطريقة اللامركزية الموزَّعة بشكلٍ أفضل (انظر الشكل 8-3). ففي نظام لامركزي لإدارة الشبكات، يمكن وجود عدد من محطات الإدارة بالمستوى الأعلى، والتي قد يُشار إليها بخادمات الإدارة (management servers). ويمكن لكل خادم منها أن يدير مباشرة بعضاً من مجموعة الوكلاء. وقد يُفوِّض خادم الإدارة مسؤولية إدارة عدد من الوكلاء إلى مدير وسيط، والذي يقوم بدور المدير في المراقبة والتحكم في الوكلاء المُندرجين تحت مسؤوليته، كما يقوم أيضاً بأداء دور الوكيل في تقديم المعلومات وتَلَقِّي التعليمات من خادم إدارة في مستوى أعلى. هذا النوع من البنية المعمارية يوزع عبء المعالجة ويقلل حركة المرور الكلية على الشبكة.

يدعم SNMPv2 استراتيجية شديدة المركزية أو استراتيجية موزَّعة لإدارة الشبكات. في الحالة الأخيرة، تؤدي بعض النظم دوراً مزدوجاً حيث تضطلع بدور المدير والوكيل. فعندما تقوم بدور الوكيل، تتلقى تلك النظم الأوامر من نظام إدارة أعلى. تتعلق بعض هذه الأوامر بقاعدة معلومات الإدارة المحلية لدى الوكيل، في حين تتطلب أوامر أخرى من الوكيل القيام بدور المُفوَّض للأجهزة البعيدة. وفي هذه الحالة، يتولى الوكيل المُفوَّض أولاً دور المدير للوصول لمعلومات عند وكيل بعيد ثم يتولى دور الوكيل لتمرير تلك المعلومات إلى مدير أعلى.

❖ تحسينات وظيفية

يشير الجدول 8-1 إلى التحسينات الوظيفية التي أضيفت في SNMPv2 مقارنةً بالإصدار الأول (SNMPv1). ويُعرَّف كلٌ من البروتوكولين بمجموعة الأوامر التي يُرسلها في صورة وحدات بيانات بروتوكول (PDU). وفي حالة SNMPv1، هناك خمسة أوامر. يُصدر المدير الأمر Get إلى وكيل لاسترجاع قيم كائنات معيّنة في قاعدة معلومات الإدارة (MIB). أما الأمر GetNext فيستغل البنية الشجرية (الهرمية) للكائنات في MIB. عندما تتم تسمية كائن في الأمر GetNext، يبحث الوكيل عن الكائن التالي في الشجرة ويسترجع القيمة الخاصة به. ويُعدّ الأمر GetNext مفيداً لأنه يسمح لمديرٍ ما بتصفّح بنية شجرية عند وكيلٍ ما في حالة عدم معرفة مجموعة الكائنات المدعومة من قِبَل ذلك الوكيل بالضبط. ويتيح الأمر Set لمديرٍ ما تحديث القيم لدى وكيلٍ ما؛ كما يُستخدَم أيضاً في إضافة الصفوف في الجداول وحذفها. أما الأمر GetResponse فيُستخدَم من قِبَل وكيل للاستجابة لتعليمات المدير. وأخيراً، يُمكِّن الأمر Trap وكيلاً ما من إرسال معلومات إلى مدير دون انتظار طلب من الإدارة. فمثلاً؛ يمكن تهيئة وكيل ليرسل إخطاراً تلقائياً عند فشل ارتباط معين أو عند تجاوز حركة المرور حداً معيناً.

يتضمن SNMPv2 كافة الأوامر الموجودة في SNMPv1 إضافةً إلى أمرين جديدين؛ أهمهما الأمر Inform، والذي يتم إرساله من محطة إدارة إلى أخرى، ويتضمن معلومات تتعلق بظروف أو أحداث في الجهاز المُرسِل، كما في الأمر Trap. يتميز الأمر Inform بإمكانية استخدامه لبناء تكوين يتعاون فيه عدد من المديرين في مسؤولية إدارة شبكة كبيرة.



الشكل 8-3: مثال لنظام موزَّع لإدارة الشبكة.

الجدول 8-1: مقارنة وحدات بيانات البروتوكول (PDUs) في SNMPv1، وSNMPv2.

الوصف	الاتجاه	SNMPV2 PDU	SNMPV1 PDU
طلب قيمة لكل كائن مسجّل في القائمة	من مدير إلى وكيل	GetRequest	GetRequest
طلب القيمة التالية لكل كائن مسجّل في القائمة	من مدير إلى وكيل	GetNextRequest	GetNextRequest
طلب قيم متعددة	من مدير إلى وكيل	GetBulkRequest	--
تعيين قيمة لكل كائن مسجّل في القائمة	من مدير إلى وكيل	SetRequest	SetRequest
إرسال معلومات غير مطلوبة	من مدير إلى مدير	InformRequest	--
رد على طلب مدير	من وكيل إلى مدير أو من مدير إلى مدير (SNMPv2)	Response	GetResponse
إرسال معلومات غير مطلوبة	من وكيل إلى مدير	SNMPv2-Trap	Trap

الأمر الجديد الآخر هو GetBulk، ويتيح لمديرٍ ما استرداد كتلة كبيرة من البيانات في وقتٍ واحد، وبالتحديد تم تصميم الأمر GetBulk لنقل جداول كاملة باستخدام أمر واحد.

فرق أخير: يُعدُّ الأمر Get أمراً أوّليًّا (atomic) في حالة SNMPv1 لكنه ليس كذلك في حالة SNMPv2. فمثلاً إذا تطلب أمر Get في بروتوكول SNMPv1 استرجاع قيم عدد من الكائنات وكان أحد تلك الكائنات (على الأقل) غير موجود عند المُفوَّض، يتم رفض الأمر بأكمله. أما بالنسبة لـ SNMPv2 فيسمح الأمر Get

باسترجاع نتائج جزئية، ولذا يسمح أمر Get غير الأوَّليّ (nonatomic) للمدير باستخدام قدرات الشبكة بشكلٍ أكفأ.

2-8 تسهيلات تجمعات SNMPv1

يوفِّر SNMPv1 – كما تم تعريفه في RFC 1157 – تسهيلات أمن بدائية مبنيَّة على مفهوم التجمع. تتيح تلك التسهيلات مستوىً معيناً من الأمن ولكنه أمنٌ مُعرَّضٌ لشتى أنواع الهجمات [JIAN02، CERT02].

1-2-8 التجمعات وأسماؤها (Communities and Community Names)

تنطوي إدارة الشبكات مثل غيرها من التطبيقات الموزَّعة الأخرى على التفاعل بين عدد من كيانات التطبيق المُعتمِدة على بروتوكول تطبيقٍ ما. وفي حالة إدارة الشبكة ببروتوكول SNMP، تكون كيانات التطبيق هي تطبيقات المدير وتطبيقات الوكيل التي تَستخدِم بروتوكول SNMP. تتسم إدارة الشبكة ببروتوكول SNMP بعدة خصائص غير معهودة في التطبيقات الموزَّعة الأخرى.

ينطوي التطبيق على علاقة "واحد إلى متعدد" (one-to-many) بين مدير ومجموعة من الوكلاء: فالمدير قادر على الحصول على قِيَم كائنات في الوكلاء وتعديلها، وقادر أيضاً على تَلَقِّي الإخطارات التلقائية من الوكلاء. وهكذا من وجهة النظر التشغيلية أو من وجهة نظر التحكم، فإن المدير "يدير" عدداً من الوكلاء. وقد يكون هناك عددٌ من المديرين يدير كلٌّ منهم كافة الوكلاء في تشكيلة النظام أو مجموعة جزئية منهم. ويمكن أن تتداخل هذه المجموعات الجزئية.

نحتاج أيضاً للنظر إلى إدارة شبكة SNMP بوصفها علاقة "واحد إلى متعدد" بين وكيل ومجموعة من المديرين. حيث يتحكم كل وكيل في عناصر قاعدة معلومات الإدارة المحلية الخاصة به، ويجب أن يكون قادراً على التحكم في استخدام تلك القاعدة من قِبَل عددٍ من المديرين. هناك ثلاثة جوانب لهذا التحكم:

- خدمة التوثيق: قد يرغب الوكيل في قصر الوصول إلى قاعدة معلومات الإدارة على المديرين المُرخّص لهم.
- سياسة الوصول: قد يرغب الوكيل في إعطاء امتيازات وصول متباينة لمديرين مختلفين.
- خدمة المُفوَّض: قد يقوم الوكيل بدور المُفوَّض لوكلاء آخرين. وقد يشمل ذلك تنفيذ خدمة التوثيق: و/أو سياسة الوصول بالنسبة للوكلاء الآخرين على النظام المُفوَّض.

تتصل جميع هذه الجوانب بالشؤون الأمنية. ففي بيئة تقسَّم فيها المسؤولية عن مكوِّنات الشبكة بين عدد من الكيانات الإدارية، يحتاج الوكلاء إلى حماية أنفسهم وحماية قواعد معلومات الإدارة الخاصة بهم من الوصول غير المرغوب فيه أو غير المصرّح به. يقوم SNMP – كما تم تعريفه في طلب تعليقات RFC 1157 – بتوفير قدرة بدائية ومحدودة لمثل هذا الأمن؛ ألا وهو مفهوم "التجمُّع" (Community).

تجمُّع SNMP هو علاقة تُعَرِّف التوثيق والتحكم في الوصول وخصائص المُفوَّض بين وكيل ومجموعة من المديرين في SNMP. يعدّ مفهوم التجمُّع مفهوماً محلياً يتم تعريفه لدى الوكيل. حيث يؤسِّس الوكيل عدة تجمعات لكل تجمعٍ منها نفس المزيج من خدمة التوثيق، وسياسة التحكم في الوصول، وخصائص المُفوَّض. ويتم إعطاء كل تجمُّع اسماً فريداً (لدى هذا الوكيل). ويتم تعريف المديرين من أعضاء هذا التجمُّع بهذا الاسم والذي يجب أن يستخدمونه في جميع عمليات أوامر Get وSet. ويمكن للوكيل إنشاء عددٍ من التجمعات التي قد تتداخل نتيجة عضوية مدير بأكثر من تجمُّع.

نظراً لأن تعريف التجمعات يتم محلياً عند الوكيل، فيمكن أن يستخدم وكلاء مختلفون نفس الاسم. وليست هوية هذه الأسماء ذات صلة ولا تشير إلى أي تشابه بين التجمعات المُعرَّفة. ولذا يجب على المدير أن يتتبع أسماء التجمعات المقترنة بكل الوكلاء المرغوب في الوصول إليها.

8-2-2 خدمة التوثيق

الغرض من خدمة التوثيق في SNMPv1 هو التأكيد للمستلم أن رسالة SNMPv1 التي استلمها هي من المصدر الذي تدّعي أنها منه. يوفّر SNMPv1 إمكانية متواضعة للتوثيق. تتضمن كل رسالة Put أو Get من مدير إلى وكيل اسم تجمّع. ويُعدُّ هذا الاسم بمثابة كلمة سر، وتُعدُّ الرسالة موثّقة إذا كان المرسل يعرف كلمة السر.

مع هذا الشكل المحدود من التوثيق، يُحجِم عددٌ من مديري الشبكات عن السماح لأي شيء بخلاف مراقبة الشبكة؛ أي عمليات Get وTrap. ومن الواضح أن التحكم في الشبكة، عن طريق Set، يعدّ موضوعاً أكثر حساسية. ويمكن أن يُستخدم اسم التجمُّع لتشغيل إجراءات التوثيق، حيث يُوظف الاسم ببساطة كجهاز أوَّليّ لفحص كلمة السر. ويمكن أن تشمل إجراءات التوثيق استخدام التشفير وإزالة التشفير لدوال توثيق أكثر أمناً، وهذا يتجاوز نطاق RFC 1157.

8-2-3 سياسة الوصول (Access Policy)

يقصر الوكيل الوصول إلى قاعدة معلومات الإدارة الخاصة به على مجموعة مختارة من المديرين عن طريق تعريف تجمُّع يضم تلك المجموعة من المديرين. باستخدام أكثر من تجمُّع، يمكن أن يوفّر الوكيل فئات مختلفة من حقوق الوصول إلى MIB لمديرين مختلفين. وهناك جانبان لهذا التحكم في الوصول:

- مشهد قاعدة معلومات الإدارة في SNMP: مجموعة جزئية من الكائنات الموجودة في MIB. ويمكن تعريف مشهد مختلف لكل تجمُّع. ومجموعة الكائنات بمشهد معين لا تنتمي بالضرورة إلى شجرة فرعية واحدة من قاعدة معلومات الإدارة.
- نمط الوصول في SNMP: هناك نمطان فقط؛ الأول: نمط "قراءة - فقط" (Read-Only)، والثاني: نمط "قراءة - كتابة" (Read-Write)؛ يتم تحديد نمط وصول لكل تجمُّع.

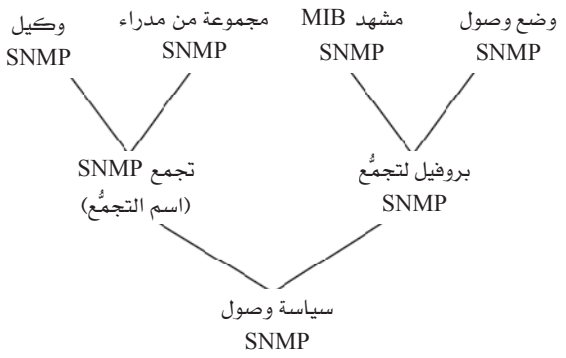
يُشار إلى تركيبة مشهد قاعدة معلومات الإدارة ونمط الوصول بالشكل العام (بروفيل) لتجمُّع SNMP. وهكذا، يتكون الشكل العام للتجمع من مجموعة جزئية من كائنات MIB مُعرَّفة عند الوكيل، بالإضافة إلى نمط وصول لتلك الكائنات. يُطبَّق نمط الوصول في SNMP بشكلٍ موحد على كافة الكائنات الموجودة في مشهد MIB. ولذا فإنه إذا تم اختيار نمط الوصول "قراءة - فقط"، فإنه ينطبق على كافة الكائنات في المشهد ويقتصر وصول المديرين على عمليات "قراءة - فقط".

يقترن الشكل العام للتجمُّع مع كل تجمُّع يتم تعريفه من قِبَل الوكيل ؛ ويُشار إلى التركيبة المكوَّنة من تجمُّع SNMP والشكل العام له بـ "سياسة وصول SNMP". ويوضّح الشكل 8-4 مختلف المفاهيم سالفة الذكر.

8-2-4 خدمة المُفوَّض

مفهوم التجمُّع مفيدٌ أيضاً في دعم خدمة المُفوَّض. وهنا نُذكِّر بأن المُفوَّض هو وكيل SNMP يعمل نيابةً عن أجهزة أخرى. وفي العادة تكون تلك الأجهزة الأخرى أجهزة خارجية بمعنى أنها لا تدعم بروتوكولات TCP/IP وSNMP. وفي بعض الحالات، قد يدعم النظام المفوِّض بروتوكول SNMP ولكنه بالرغم من ذلك يستخدم المُفوَّض ليقلل من تعاملاته مع نظم إدارة الشبكة.

يحتفظ النظام المُفوَّض بسياسة SNMP للوصول لكل جهاز يمثله. ولذا فهو يعرف أي الكائنات التي في MIB يمكن استخدامها لإدارة النظام المفوِّض (proxied system) ونمط الوصول الخاص بها.



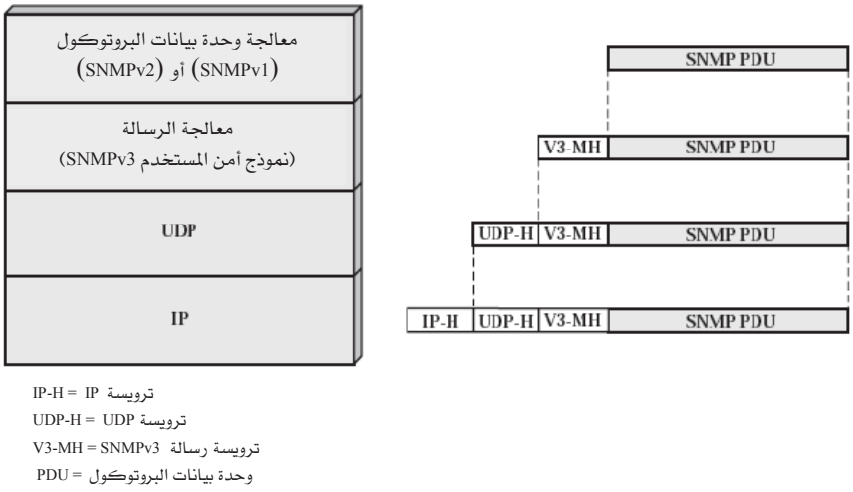
الشكل 8-4: المفاهيم الإدارية لبروتوكول SNMPv1.

8-3 الإصدار الثالث لبروتوكول إدارة الشبكات البسيط

في عام 1998 وضعت مجموعة عمل SNMPv3 – التابعة لـ فريق مهام هندسة الإنترنت (IETF) – مجموعة من معايير الإنترنت المقترحة، والمعروفة حالياً بمجموعة طلبات التعليقات من RFC 2570 حتى RFC 2576. تُحدد هذه المجموعة من الوثائق إطاراً لدمج السمات الأمنية في إمكانية شاملة تتضمن وظائف SNMPv1 أو SNMPv2. كما تُعَرِّف الوثائق أيضاً مجموعة محدّدة من الإمكانات لأمن الشبكات والتحكم في الوصول.

من المهم إدراك أن SNMPv3 ليس بديلاً عن أيٍّ من SNMPv1 أو SNMPv2 ولا لكليهما. حيث يُعرِّف SNMPv3 قدرات أمنية لاستخدامها مع SNMPv2 (المفضَّل) أو SNMPv1. بالإضافة إلى ذلك، يصف RFC 2571 بنية معمارية تُلائم جميع إصدارات SNMP الحالية والمستقبلية. كما يصف RFC 2575 وسيلة للتحكم في الوصول، والتي يقصد منها أن تعمل بشكلٍ مستقل عن الإمكانات الأساسية لـ SNMPv3. سنقوم في هذا الجزء بإلقاء نظرة شاملة وعمل دراسة مسحية عن الإمكانات المُعرَّفة في RFC 2570 حتى RFC 2576 .

يبيّن الشكل 8-5 العلاقة بين الإصدارات المختلفة من SNMP من خلال الصيغ المُستخدَمة. ويتم تبادل المعلومات بين محطة إدارة ووكيل عن طريق رسالة SNMP. وتتم المعالجة الأمنية على مستوى الرسالة؛ فعلى سبيل المثال، يُحدِّد SNMPv3 نموذجاً لأمن المُستخدِم ((User Security Model (USM) والذي يستخدم حقول ترويسة الرسالة. وتكون حمولة رسالة SNMP إما في شكل وحدة بيانات بروتوكول SNMPv1 (SNMPv1 PDU) أو وحدة بيانات بروتوكول SNMPv2 (SNMPv2 PDU). تُشير وحدة بيانات البروتوكول (PDU) إلى نوع من الإجراءات الإدارية (مثل استرداد أو تعيين قيم الكائنات المُدارة)، وتُشير أيضاً إلى قائمة بأسماء المتغيرات ذات العلاقة بذلك الإجراء.



الشكل 8-5: البنية المعمارية لبروتوكول SNMP.

تصف طلبات التعليقات من RFC 2570 حتى RFC 2576 بنية معمارية شاملة بالإضافة إلى السمات الأمنية والهياكل المحدّدة للرسائل، لكنها لا تُعرِّف صيغاً جديدة لوحدة بيانات البروتوكول (PDU) في SNMP. ولذلك، يجب استخدام صيغ PDU الموجودة في SNMPv1 أو SNMPv2 داخل البنية الجديدة. تتضمن تطبيقات

SNMPv3 السمات الأمنية والمعمارية المُعرَّفة في RFC 2570 حتى RFC 2576 إضافةً إلى صيغة PDU والوظائف المُعرَّفة في وثائق SNMPv2. وقد تم التعبير عن ذلك في RFC 2570 بالنص التالي: "يمكن اعتبار SNMPv3 على أنه SNMPv2 مع بعض الإمكانات الأمنية والإدارية المضافة."

فيما تبقى سنعرض في هذا الجزء أولاً مقدمة موجزة للبنية المعمارية الأساسية لبروتوكول SNMP المُعرَّفة في RFC 2571، ثم نتناول ميزات السرية والتوثيق التي يوفرها نموذج أمن المُستخدِم (USM) لبروتوكول SNMPv3. وأخيراً، سنناقش التحكم في الوصول ونموذج التحكم في الوصول المبنيّ على المشهد (View-based Access Control Model (VACM)).

8-3-1 البنية المعمارية لبروتوكول SNMP

تتكون البنية المعمارية لـ SNMP – كما هو معروض في RFC 2571 – من مجموعة كيانات SNMP موزَّعة ومتفاعلة. ويقوم كل كيان بتنفيذ جزء من إمكانات SNMP وقد يتصرف كوكيل أو مدير أو مزيج من الاثنين معاً. ويتكون كيان SNMP من مجموعة من الوحدات (modules) التي تتفاعل مع بعضها بعضاً لتوفير الخدمات. يمكن نمذجة هذه التفاعلات كمجموعة من العمليات الأوَّليَّة والمُعامِلات.

تعكس البنية المعرَّفة في RFC 2571 متطلباً رئيساً لتصميم SNMPv3 وهو: تصميم بنية معمارية مكونة من وحدات والتي سوف: (1) تسمح بالتنفيذ عبر مجموعة واسعة من البيئات التشغيلية، والتي قد يحتاج بعضها إلى استخدام القليل من الإمكانات غير المكلِّفة، في حين قد يدعم البعض الآخر منها ميزات إضافية لإدارة شبكات كبيرة؛ (2) تجعل من الممكن إحداث تقدم من حيث إقرار معيارية بعض أجزاء البنية المعمارية حتى لو لم يتم التوصل إلى اتفاق موحَّد على كافة الأجزاء؛ و(3) يمكنها استيعاب نماذج أمنية بديلة.

❖ كيان SNMP

يتضمَّن كل كيان SNMP مُحرّك SNMP واحداً. ويقوم محرك SNMP بتنفيذ وظائف إرسال الرسائل واستقبالها، وتوثيق الرسائل وتشفيرها وإزالة تشفيرها، والتحكم في الوصول إلى الكائنات المُدارة. ويتم توفير هذه الوظائف كخدمات لتطبيق واحد أو لعدد أكبر من التطبيقات والتي يتم تهيئتها بمحرك SNMP لتُشكِّل "**كيان** SNMP".

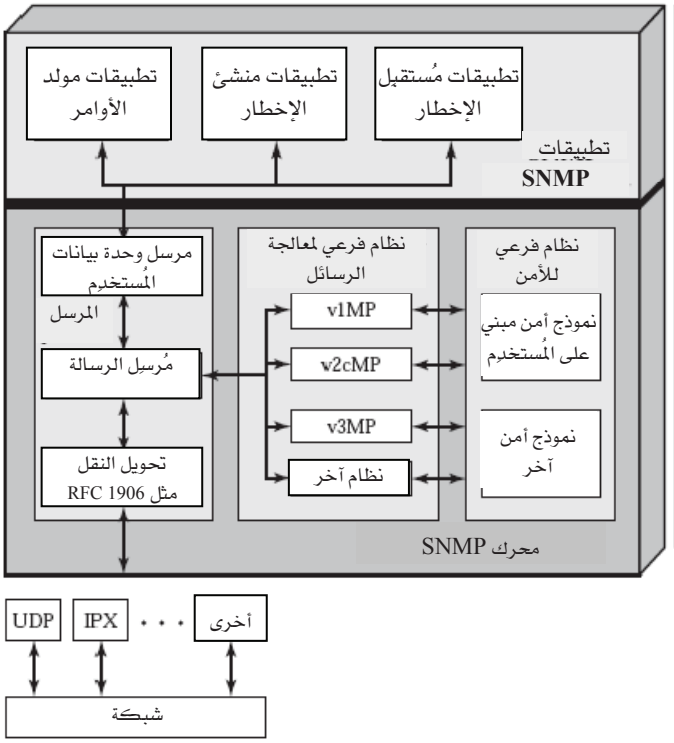
يُعرَّف كلٌّ من محرك SNMP والتطبيقات التي يدعمها بمجموعة من الوحدات المنفصلة. وتوفر هذه البنية المعمارية مزايا عدة كما يأتي:

أولاً: يتم تحديد دور كيان SNMP على أساس الوحدات المُنفَّذة في هذا الكيان. يحتاج وكيل SNMP إلى مجموعة معيّنة من الوحدات بينما يحتاج مدير SNMP إلى مجموعة مختلفة (مع وجود بعض التطابق) من الوحدات.

ثانياً: يسهل الهيكل التركيبي (modular structure) للمواصفات من عملية تعريف إصدارات مختلفة لكل وحدة. وهذا بدوره يجعل من الممكن: (1) تعريف إمكانات بديلة أو محسنَّة لجوانب معيّنة من SNMP دون الحاجة إلى إصدار جديد للمعيار القياسي بأكمله (مثلاً: SNMPv4)، و(2) تحديد واضح لاستراتيجيات التعايش والانتقال بين الإصدارات المختلفة (RFC 2576).

لفهم دور كل وحدة وعلاقتها بالوحدات الأخرى بشكلٍ أفضل، نحتاج إلى النظر إلى استخداماتها في مديري ووكلاء SNMP التقليديين. ويُستخدَم المصطلح "تقليدي" بمعنى نقي، ليؤكد على حقيقة أن تنفيذاً معيناً لا يلزم أن يكون مديراً أو وكيلاً خالصاً ولكن قد يكون لديه الوحدات التي تسمح للكيان بأداء كلٍّ من مهام الوكيل والمدير.

كيان SNMP



الشكل 8-6: مدير SNMP التقليدي.

يعتمد الشكل 8-6 على شكل في RFC 2571، ويمثل رسماً تخطيطياً لمدير SNMP تقليدي. يتفاعل مدير SNMP التقليدي مع وكلاء SNMP عن طريق إصدار أوامر Get وSet وباستقبال رسائل الإخطارات التلقائية (Trap)؛ كما يمكن أن يتفاعل أيضاً مع مديرين آخرين من خلال إصدار وحدات بيانات البروتوكول لطلب الإبلاغ (Inform Request PDUs) – والتي توفر تنبيهات – وكذلك تَلَقِّي رسائل وحدات بيانات البروتوكول كاستجابة للإبلاغ (Inform Response PDUs)، والتي تقر باستلام طلبات إبلاغ (Inform Requests). في مصطلحات SNMPv3، يتضمن مدير SNMP التقليدي ثلاث فئات من التطبيقات. تطبيقات مولِّد الأوامر (Command Generator Applications) والتي تراقب وتعالج بيانات الإدارة في الوكلاء البعيدين – باستخدام وحدات بيانات بروتوكول SNMPv1 و/أو SNMPv2

– ويشمل ذلك أوامر Get وGetNext وGetBulk وSet. أما تطبيق مُنشئ الإخطارات (Notification Originator Application) فيقوم بإرسال الرسائل غير المتزامنة. ويتم استخدام وحدة بيانات البروتوكول لطلب الإبلاغ (Inform Request PDU) لهذا التطبيق في حالة مدير تقليدي. ويقوم تطبيق مُستقبِل الإخطارات (Notification Receiver Application) بمعالجة ما يَرِد من الرسائل غير المتزامنة؛ والتي تشمل طلب الإبلاغ (InformRequest)، وإخطارات SNMPv2 التلقائية، ووحدات بيانات بروتوكول SNMPv1 للإخطارات التلقائية. ويستجيب هذا التطبيق (مُستقبِل الإخطارات) بإرسال وحدة بيانات البروتوكول للاستجابة (Response PDU) في حالة ورود وحدة بيانات البروتوكول لطلب الإبلاغ (InformRequest PDU).

تستفيد جميع التطبيقات المشروحة آنفاً من الخدمات التي يقدمها محرك SNMP لهذا الكيان. ويؤدي محرك SNMP مهمتين عامتين:

- يستقبل وحدات بيانات البروتوكول الصادرة من تطبيقات SNMP؛ ويقوم بالمعالجة الضرورية، بما في ذلك إدراج شفرات/ رموز التوثيق والتشفير؛ ومن ثمَّ يقوم بتغليف وحدات بيانات البروتوكول داخل رسائل للنقل.
- يستقبل رسائل SNMP الواردة من طبقة النقل؛ ويقوم بالمعالجة الضرورية بما في ذلك التوثيق وإزالة التشفير؛ ومن ثم يستخلص وحدات بيانات البروتوكول من الرسائل ويمررها إلى تطبيق SNMP المناسب.

يحتوي محرك SNMP في حالة المدير التقليدي على مُرسِل، ونظام فرعي لمعالجة الرسائل، ونظام فرعي للأمن. ويكون المُرسِل هنا هو مجرد مدير بسيط لحركة المرور. وبالنسبة لوحدات بيانات البروتوكول الصادرة، يستقبل المُرسِل وحدات بيانات البروتوكول من التطبيقات ويؤدي الوظائف الآتية: يحدِّد المُرسِل نوع المعالجة المطلوبة لكل وحدة بيانات (على سبيل المثال: SNMPv1، أو SNMPv2c، أو SNMPv3)، ثم يمرِّرها إلى الوحدة المناسبة لمعالجة الرسالة في النظام الفرعي لمعالجة الرسائل. بعد ذلك يقوم النظام الفرعي لمعالجة الرسائل بإرجاع رسالة تحتوي على وحدة بيانات البروتوكول تلك بالإضافة إلى ترويسة مناسبة للرسالة. بعدها يقوم المُرسِل بترحيل تلك الرسالة إلى طبقة النقل.

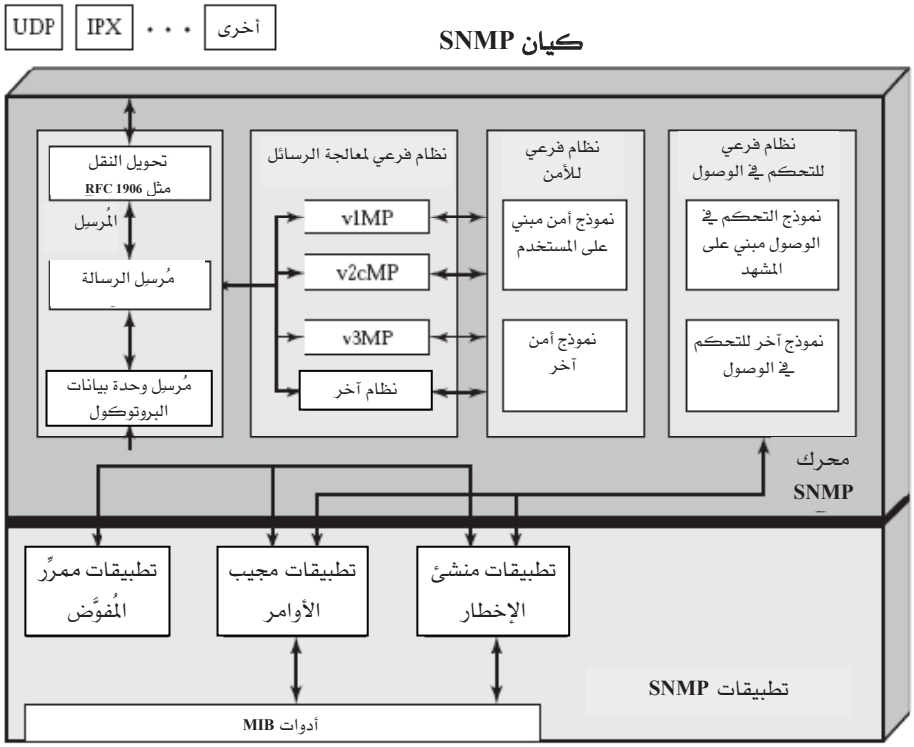
بالنسبة للرسائل الواردة، يستقبل المُرسِل الرسائل من طبقة النقل ويؤدي الوظائف الآتية. يقوم المُرسِل بتوجيه كل رسالة إلى الوحدة المناسبة لمعالجة الرسالة. بعد ذلك يعيد النظام الفرعي لمعالجة الرسائل وحدة بيانات البروتوكول الواردة بالرسالة، فيمررها المُرسِل إلى التطبيق المناسب.

يقوم النظام الفرعي لمعالجة الرسائل باستقبال PDUs الصادرة من المُرسِل وتجهيزها للنقل بتغليفها في ترويسة مناسبة ثم إعادتها إلى المُرسِل. أما بالنسبة للرسائل الواردة فيقوم النظام الفرعي لمعالجة الرسائل باستقبال الرسائل أيضاً من المُرسِل، ومعالجة ترويسة كل رسالة، ومن ثَمَّ إعادة PDU المرفقة بها إلى المُرسِل. وقد يدعم تطبيق النظام الفرعي لمعالجة الرسائل صيغة واحدة للرسائل موافقة لإصدار واحد من SNMP (SNMPv1، SNMPv2c، SNMPv3)، أو قد يحتوي النظام على عددٍ من الوحدات يدعم كلٌ منها إصداراً مختلفاً من SNMP.

أما النظام الفرعي للأمن فيقوم بأداء وظائف التشفير والتوثيق. يتم تمرير كل رسالة صادرة من النظام الفرعي لمعالجة الرسائل إلى النظام الفرعي للأمن. بناءً على الخدمات المطلوبة، قد يقوم النظام الفرعي للأمن بتشفير PDU المرفقة وربما بعض الحقول في ترويسة الرسالة، وقد يقوم بتوليد شفرة التوثيق وإدرجها في ترويسة الرسالة. ثم يتم إعادة الرسالة المُعالَجة إلى النظام الفرعي لمعالجة الرسائل. وبالمثل يتم تمرير كل رسالة واردة إلى النظام الفرعي للأمن من النظام الفرعي لمعالجة الرسائل، حيث يتحقق النظام الفرعي للأمن من رمز التوثيق – إذا تطلب الأمر ذلك – ويقوم بإزالة التشفير. ثم يقوم بإعادة الرسالة المُعالَجة إلى النظام الفرعي لمعالجة الرسائل. وقد يدعم تطبيق النظام الفرعي للأمن واحداً أو أكثر من نماذج الأمن المتميزة. وحتى الآن، يُعدّ نموذج الأمن الوحيد المُعرَّف هو نموذج أمن المُستخدِم (USM) لبروتوكول SNMPv3، والمُعرَّف في RFC 2574.

الشكل 8-7 – والمُعتمِد على شكل في RFC 2571 – عبارة عن رسم تخطيطي لوكيل SNMP تقليدي والذي قد يتضمن ثلاثة أنواع من التطبيقات. توفّر تطبيقات مجيب الأوامر (Command Responder) الوصول إلى بيانات الإدارة. تستجيب هذه

التطبيقات للطلبات الواردة من خلال استرداد و/أو تعيين قيم الكائنات المُدارة ومن ثمّ إصدار PDU للاستجابة. يقوم تطبيق مُنشئ الإخطارات بإصدار رسائل غير متزامنة؛ في حالة الوكيل التقليدي تُستخدم إخطارات SNMPv2 أو SNMPv1 التلقائية لهذا التطبيق. يعيد تطبيق ممرِّر المُفوَّض (Proxy Forwarder Application) توجيه الرسائل بين الكيانات.



الشكل 8-7: وكيل SNMP التقليدي.

يتضمن محرك SNMP لوكيل تقليدي كافة المكوِّنات الموجودة في مُحرّك SNMP للمدير التقليدي، بالإضافة إلى نظام فرعي للتحكم في الوصول. يوفر هذا النظام الفرعي خدمات الترخيص للتحكم في الوصول إلى قواعد معلومات الإدارة لقراءة وتعيين قِيم كائنات الإدارة. ويتم تنفيذ هذه الخدمات على أساس محتويات وحدات بيانات البروتوكول (PDUs). وقد يدعم تنفيذ معيَّن للنظام الفرعي للأمن نموذجاً واحداً أو عدة نماذج منفصلة للتحكم في الوصول. وحتى الآن يُعدّ نموذج الأمن الوحيد المُعرَّف هو نموذج التحكم في الوصول المبنيّ على المشهد (VACM) بالنسبة SNMPv3، والمعرَّف في RFC 2575.

لاحظ أنه يتم تنظيم المهام المتصلة بالأمن في نظامين فرعيين منفصلين وهما: نظام الأمن ونظام التحكم في الوصول. ويُعدُّ هذا مثالاً ممتازاً على التصاميم التركيبية الجيدة، لأن كلا النظامين الفرعيين يؤديان وظائف مختلفةً تماماً، ولذا فمن الطبيعي السماح للتوحيد القياسي في هذين المجالين بالمُضي قُدماً بشكلٍ مستقل. ويُعنَى النظام الفرعي للأمن بالخصوصية والتوثيق ويعمل على رسائل SNMP. أما النظام الفرعي للتحكم في الوصول فيهتم بالوصول المُرخّص به إلى معلومات الإدارة ويعمل على وحدات بيانات بروتوكول SNMP.

❖ المصطلحات

يُعرِّف الجدول 8-2 بإيجاز بعض المصطلحات التي تم عرضها في RFC 2571. ويقترن بكل كيان هوية فريدة للمحرك (snmpEngineID). وللتحكم في الوصول يمكن اعتبار أن كل كيان SNMP يقوم بإدارة عددٍ من سياقات المعلومات المُدارة (contexts of managed information)، ولكل سياق اسمٌ فريد (contextName) داخل ذلك الكيان. للتأكيد على أن هناك مديراً واحداً للسياقات داخل كيانٍ ما، يخصّص لكل كيان هوية محرك سياق فريدة (contextEngineID). ونظراً لوجود تناظر "واحد إلى واحد" بين محرك السياق ومحرك SNMP في هذا الكيان، فإن هوية محرك السياق (contextEngineID) تطابق قيمة هوية محرك SNMP (snmpEngineID). ويخضع التحكم في الوصول للسياق المحدّد الذي تمت فيه محاولة الوصول ولهوية المُستخدِم الذي طلب الوصول؛ يتم التعبير عن هذه الهوية الأخيرة بكيان رئيس (principal) والذي قد يكون فرداً أو تطبيقاً أو مجموعةً من الأفراد أو التطبيقات.

الجدول 8-2: مصطلحات SNMPv3.

snmpEngineId هوية فريدة لا لبس فيها لمحرك SNMP، فضلاً عن كيان SNMP الذي يتوافق مع ذلك المحرك. والذي يُعرَّف بجملة نصّية بصيغة سلسلة بايتات.
contextEngineID يُعرِّف على نحو فريد كيان SNMP الذي قد يحقق مثيلاً لسياق له اسم محدّد contextName.
contextName يعرّف سياقاً معيَّناً داخل محرك SNMP. وهذا يتم تمريره كمُعامِل إلى المرسل وإلى النظام الفرعي للتحكم في الوصول.
scopedPDU كتلة من البيانات التي تتألف من (contextEngineID) و(contextName) و وحدة بيانات بروتوكول SNMP (SNMP PDU). يتم تمريرها كمُعامِل من/إلى النظام الفرعي للأمن.
snmpMessageProcessingModel هوية فريدة لنموذج معالجة الرسالة من النظام الفرعي لمعالجة الرسائل. تتضمن القيم المحتملة SNMPv1 و SNMPv2c و SNMPv3. تُعرَّف هذه الهوية بجملة نصّية بصيغة عدد صحيح.
snmpSecurityModel هوية فريدة لنموذج أمني من النظام الفرعي للأمن. تتضمن القيم المحتملة SNMPv1 وSNMPv2c وUSM. وتُعرَّف بجملة نصية بصيغة عدد صحيح.
snmpSecurityLevel مستوى من الأمن الذي يمكن فيه إرسال رسائل SNMP أو يمكن من خلاله معالجة أية عمليات، أو التعبير عنها بمعايير توفير أو عدم توفير التوثيق و/أو الخصوصية. القيم البديلة هي (noAuthnoPriv) و(authNoPriv) و(authPriv). يتم تعريفه بواسطة بجملة نصية بصيغة عدد صحيح.
Principal الرئيس: الكيان الذي تُقدَّم الخدمات أو تتم المعالجة نيابةً عنه. ويمكن أن يكون الرئيس فرداً يعمل في دور محدّد؛ أو مجموعةً من الأفراد مع دور محدّد لكلٍّ منها؛ أو تطبيقاً أو مجموعةً من التطبيقات؛ أو تشكيلات من كلٍّ منها.
securityName سلسلة يمكن للإنسان قراءتها تمثل الرئيس. ويتم تمريرها كمُعامِل في جميع أوَّليَّات SNMP (المُرسِل، معالجة الرسائل، الأمن، التحكم في الوصول).

تتعلق بعض المصطلحات الأخرى ذات الأهمية بمعالجة الرسائل. ويحدِّد المصطلح snmpMessageProcessingModel صيغة الرسالة ورقم إصدار SNMP لمعالجة الرسالة. كما يحدِّد snmpSecurityModel النموذج الأمني الذي سيتم استخدامه. ويحدِّد المصطلح snmpSecurityLevel الخدمات الأمنية المطلوبة لهذه العملية تحديداً. ويمكن للمُستخدِم أن يطلب عملية التوثيق فقط أو التوثيق بالإضافة إلى الخصوصية (التشفير)، أو لا يطلب أيّاً منهما.

❖ تطبيقات SNMPv3

تم تعريف الخدمات بين الوحدات النمطية في كيان SNMP في RFCs في شكل عمليات أوَّليَّة ومُعامِلات. تحدد العمليات الأوَّليَّة الوظيفة التي سيتم تنفيذها، في حين تستخدم المُعامِلات لتمرير البيانات ومعلومات التحكم. ويمكن اعتبار هذه العمليات الأوَّليَّة والمُعامِلات أسلوباً رسمياً لتعريف خدمات SNMP. وتعتمد الصورة الفعلية للعمليات الأوَّليَّة على كيفية التنفيذ؛ مثال على ذلك طلب استدعاء إجراء (procedure call). في المناقشة التالية، سيكون من المفيد الإشارة إلى الشكل 8-8 والمبنيّ على شكل في RFC 2571 لمعرفة كيف تتواءم كل هذه العمليات الأوَّليَّة معاً. ويبيّن الشكل 8-8 (a) تسلسل الأحداث حيث يطلب أيٌّ من تطبيقي مولِّد الأوامر أو مُنشئ الإخطارات إرسال PDU، وما يتبع ذلك من كيفية إعادة الاجابة على هذا الطلب إلى ذلك التطبيق؛ وتقع هذه الأحداث عند نظام مدير. ويبيّن الشكل 8-8 (b) الأحداث المقابلة حين تقع عند نظام وكيل. كما يوضّح الشكل كيف تؤدي رسالة واردة إلى إرسال PDU المرفقة بالرسالة إلى تطبيق ما، وكيف تؤدي استجابة ذلك التطبيق إلى رسالة صادرة. لاحظ أن بعض الأسهم الموجودة في الرسم التخطيطي والتي تم وسمها باسم عملية أوَّليَّة تمثل استدعاء إجراء. وتمثل الأسهم غير الموسومة العودة من الإجراء، كما يشير التظليل إلى المطابقة بين الاستدعاء والعودة.

يُعرِّف RFC 2573، بصورة عامة، الإجراءات المتَّبعة لكل نوع من أنواع التطبيقات عند إنشاء PDUs لإرسالها أو معالجة PDUs واردة. وفي جميع الحالات،

يتم تعريف الإجراءات من حيث التفاعل مع المُرسِل (Dispatcher) بواسطة العمليات الأوَّليَّة للمرسل (Dispatcher primitives).

يَستخدم تطبيق مولِّد الأوامر العمليات الأوَّليَّة للمُرسِل والمسماة sendPdu وResponsePdu. تزوِّد sendPdu المُرسِل بمعلومات حول الوِجهة المقصودة، ومُعامِلات الأمن، ووحدة بيانات البروتوكول المُعَدَّة فعلياً للإرسال. ثم يستدعي المُرسِل نموذج معالجة الرسائل، والذي يستدعي بدوره النموذج الأمني، لإعداد الرسالة. يقوم المُرسِل بتسليم الرسالة المُعَدَّة إلى طبقة النقل (مثلاً UDP) لإرسالها. في حالة فشل إعداد الرسالة، فإن القيمة الأوَّليَّة الراجعة من sendPdu، والمُحدَّدة سلفاً من قِبَل المُرسِل، هي مؤشر خطأ. في حال نجاح عملية إعداد الرسالة، يحدِّد المُرسِل قيمةً للمُعامِل sendPduHandle كهوية لوحدة بيانات البروتوكول تلك ويعيد تلك القيمة إلى مولِّد الأوامر. ويقوم مولِّد الأوامر بتخزين قيمة المُعامِل sendPduHandle بحيث يمكنه مطابقة الطلب الأصلي مع وحدة بيانات البروتوكول بالاستجابة التالية. يقوم المُرسل بتسليم كل PDU واردة إلى تطبيق مولِّد الأوامر الصحيح، باستخدام processResponsePdu.

يَستخدم تطبيق مجيب الأوامر أربعاً من عمليات المُرسل الأوَّليَّة: registerContextEngineID، unregisterContextEngineID، processPdu، returnResponsePdu. كما يستخدم عملية أوَّليَّة للنظام الفرعي للتحكم في الوصول هي isAccessAllowed.

تُمكِّن العملية الأوَّليَّة registerContextEngineID تطبيق مجيب الأوامر من ربط نفسه مع محرك SNMP من أجل معالجة بعض أنواع PDUs لمحرك السياق. وبمجرد تسجيل مجيب الأوامر، يتم إرسال كافة الرسائل التي تم استقبالها بشكلٍ غير متزامن والتي تحتوي على التركيبة المسجَّلة من هوية محرك السياق (contextEngineID) ونوع بيانات البروتوكول المدعوم (pduType) إلى مجيب الأوامر الذي تم تسجيله لدعم هذه التركيبة. ويمكن لمجيب الأوامر فك ارتباطه مع محرك SNMP باستخدام العملية الأوَّليَّة unregisterContextEngineID.

(a) مولد الأوامر أو منشئ الإخطار

(b) مجيب الأوامر

الشكل 8-8: تدفق SNMPv3.

يُسلِّم المُرسِل وحدة بيانات البروتوكول الواردة بكل طلب إلى تطبيق مجيب الأوامر المناسب باستخدام العملية الأوَّليَّة processPdu. ثم يقوم مجيب الأوامر بالخطوات الآتية:

- يفحص مجيب الأوامر محتويات PDU لهذا الطلب. ويجب أن يتطابق نوع العملية مع أحد الأنواع التي تم تسجيلها من قِبَل هذا التطبيق.
- يحدِّد مجيب الأوامر ما إذا كان الوصول مسموحاً لأداء عملية الإدارة المطلوبة من قِبَل تلك الـ PDU. ولهذا الغرض، يتم استدعاء العملية الأوَّليَّة isAccessAllowed. ويُشير مُعامِل securityModel إلى نموذج الأمن الذي سيستخدمه النظام الفرعي للتحكم في الوصول في الرد على هذا الاستدعاء. ويحدِّد النظام الفرعي للتحكم في الوصول ما إذا كان مسؤول الطلب (securityName) في المستوى الأمني (securityLevel) لديه الإذن لطلب عملية الإدارة (viewType) على كائن الإدارة (variabeName) في هذا السياق (contextName).
- إذا تم السماح بالوصول، يُنجز مجيب الأوامر عملية الإدارة ويجهز PDU للاستجابة. وفي حال فشل الوصول، يقوم مجيب الأوامر بإعداد PDU استجابة ملائمة للإشارة إلى ذلك الفشل.
- يستدعي مجيب الأوامر المُرسِل باستخدام العملية الأوَّليَّة returnResponsePdu لإرسال PDU الاستجابة.

يتبع تطبيق مُنشئ الإخطارات نفس الإجراءات العامة التي تستخدم في تطبيق مولِّد الأوامر. في حالة إرسال PDU لطلب إبلاغ، يتم استخدام العمليتين الأوليين sendPdu وprocessResponsePdu بنفس الطريقة كما في تطبيق مولِّد الأوامر. في حالة إرسال PDU لإخطار تلقائي، تُستخدم فقط العملية الأوَّليَّة sendPdu.

يتبع تطبيق مُستقبِل الإخطارات جزءاً محدّداً من الإجراءات العامة المتَّبعة في تطبيق مجيب الأوامر. ويجب أولاً على مُستقبِل الإخطارات التسجيل ليستقبل PDUs الخاصة بالإبلاغ و/أو الإخطار التلقائي. كلا النوعين من وحدات بيانات

البروتوكول يتم استقبالهما بواسطة العملية الأوَّليَّة processPdu. وبالنسبة لوحدة بيانات البروتوكول للإبلاغ، يتم الرد باستخدام العملية الأوَّليَّة returnResponsePdu.

يستفيد تطبيق مُمرِّر المُفوَّض من العمليات الأوَّليَّة لدى المُرسِل لإعادة توجيه رسائل SNMP. ويعالج مُمرِّر المُفوَّض أربعة أنواع أساسية من الرسائل:

- الرسائل التي تحتوي على أنواع وحدات بيانات البروتوكول من تطبيق مولِّد الأوامر. يحدِّد تطبيق مُمرِّر المُفوَّض المحرك المستهدف ليكون إما محرك SNMP المستهدف أو محرك SNMP الأقرب أو الذي يقع في نفس الاتجاه، ومن ثَمَّ يُرسِل وحدة بيانات البروتوكول المناسبة للطلب.
- الرسائل التي تحتوي على أنواع وحدات بيانات البروتوكول من تطبيق مُنشئ الإخطارات. ويحدِّد تطبيق مُمرِّر المُفوَّض محركات SNMP التي ينبغي أن تستلم الإخطار ثم يُرسِل وحدة أو وحدات بيانات البروتوكول المناسبة للإخطار.
- الرسائل التي تحتوي على وحدات بيانات البروتوكول من نوع استجابة. يحدِّد مُمرِّر المُفوَّض الطلب أو الإخطار الذي سبق توجيهه الذي يناظر هذه الاستجابة – إن وُجد – ثم يقوم بإرسال وحدة بيانات البروتوكول الملائمة كاستجابة.
- الرسائل التي تتضمن إشارة تقرير. تمثل وحدات بيانات البروتوكول للتقرير اتصالات SNMPv3 من محرك إلى محرك. ويحدِّد مُمرِّر المُفوَّض الطلب أو الإخطار الذي سبق توجيهه – إن وُجد – والذي يناظر إشارة هذا التقرير ويعيد توجيه إشارة التقرير مرة أخرى إلى بادئ الطلب أو الإخطار.

8-3-2 معالجة الرسالة ونموذج أمن المُستخدِم

تتضمن معالجة الرسائل نموذجاً عاماً لمعالجة الرسائل بالإضافة إلى نموذج محدَّد للأمن؛ ويوضّح الشكل 8-8 هذه العلاقة.

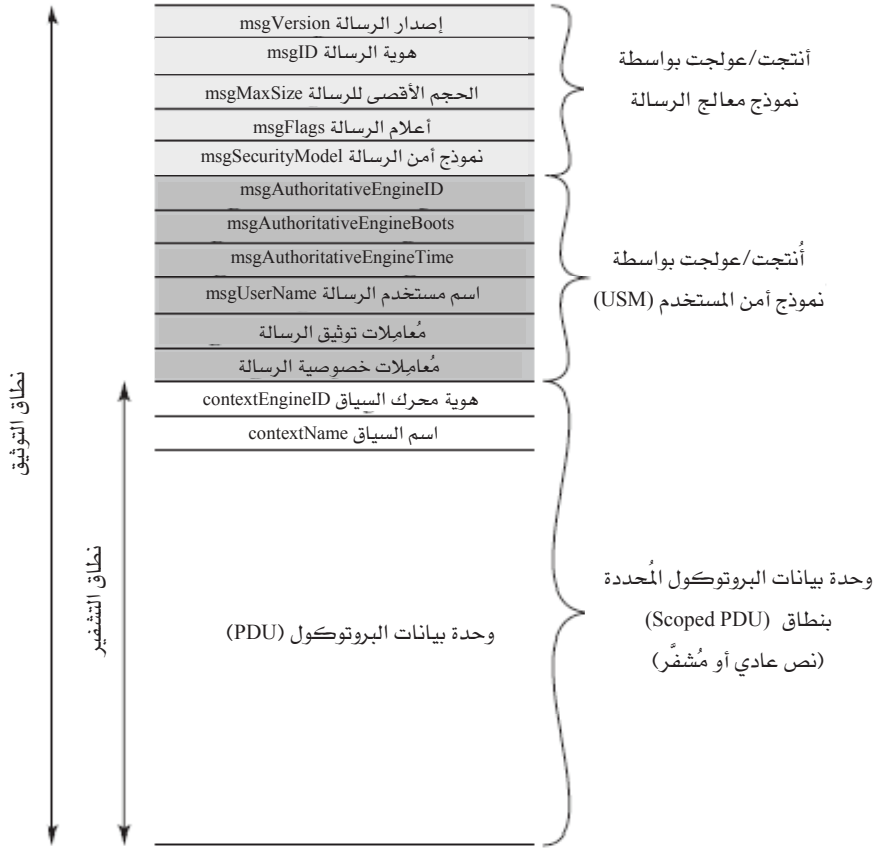
❖ نموذج معالجة الرسائل

يُعرِّف RFC 2572 نموذجاً عاماً لمعالجة الرسائل. وهذا النموذج مسؤول عن قبول وحدات بيانات البروتوكول من المُرسِل، وتغليفها في رسائل، والاستعانة بنموذج أمن المُستخدِم (USM) لإدراج المُعامِلات (البارامترات) الأمنية في ترويسة الرسالة. كما يقبل نموذج معالجة الرسالة أيضاً الرسائل الواردة، ويستعين بنموذج أمن المُستخدِم USM لمعالجة المُعامِلات الأمنية في ترويسة الرسالة، ثم يُسلِّم وحدة بيانات البروتوكول المغلَّفة إلى المرسل.

يوضّح الشكل 8-9 هيكل الرسالة. وتنتج الحقول الخمسة الأولى عن نموذج معالجة الرسالة للرسائل الصادرة ويتم معالجتها بواسطة نموذج معالجة الرسالة للرسائل الواردة. وتعرض الحقول الستة التالية مُعامِلات الأمن المُستخدَمة من قِبَل نموذج أمن المُستخدِم (USM). وأخيراً، تُشكل وحدة بيانات البروتوكول مع هوية محرك السياق (contextEngineID) واسم السياق (contextName) ما يمكن أن يسمى وحدة بيانات البروتوكول المُحدَّدة بنطاق (scoped PDU)، وتستخدم في معالجة وحدة بيانات البروتوكول.

الحقول الخمسة الأولى هي:

- رقم إصدار الرسالة (msgVersion): يأخذ القيمة 3 (أي SNMPv3).
- هوية الرسالة (msgID): مُعرِّف فريد يُستخدَم بين كيانَي SNMP لتنسيق رسائل الطلب والاستجابة، وبواسطة معالج الرسالة لتنسيق معالجة الرسالة بواسطة نماذج نظام فرعي مختلف داخل البنية. نطاق هذا المُعرِّف هو 0 إلى $(2^{31}-1)$.
- الحد الأقصى لحجم الرسالة (msgMaxSize): الحد الأقصى لحجم الرسائل المُعتمَد من قِبَل مُرسِل الرسالة بالبايتات (octets)، وتتراوح قيمته من 484 إلى $(2^{31}-1)$. هذا هو الحد الأقصى لحجم الجزء الذي يمكن للمرسل قبوله من محرك SNMP آخر (سواء كان رداً أو نوع رسالة آخر).



الشكل 8-9: صيغة رسالة SNMPv3 مع نموذج أمن المُستخدِم (USM).

- أعلام الرسالة (msgFlags): سلسلة من ثمانية بتات تحتوي على ثلاثة أعلام في البتات الثلاثة الأقل وزناً: reportableFlag، وprivFlag، وauthFlag. إذا كان العلم reportableFlag =1، فيجب إعادة PDU من نوع تقرير إلى المُرسِل في الحالات التي يمكن أن تتسبب في إنتاج PDU للتقرير؛ وعندما تكون قيمة العلم صفراً، يمكن ألا يتم إرسال PDU للتقرير. يتم وضع reportableFlag عند القيمة 1 بواسطة المُرسِل في كافة الرسائل التي

تحتوي على طلب (Get، Set) أو على طلب إبلاغ، بينما يوضع عند القيمة 0 للرسائل التي تحتوي على PDU للاستجابة أو للإخطار التلقائي، أو للتقرير. ويُعدّ reportableFlag بمثابة عامل ثانوي لتحديد متى يتم إرسال تقرير. ويُستخدم فقط في الحالات التي لا يمكن ترجمة الجزء الخاص بوحدة بيانات البروتوكول لهذه الرسالة (مثلاً عند فشل إزالة التشفير بسبب مفتاح غير صحيح). يتم تحديد قيم privFlag وauthFlag بواسطة المُرسِل ليشير إلى مستوى الأمن الذي تم تطبيقه على الرسالة. فحالة privFlag =1، تعني أنه قد تم تطبيق التشفير، أما حالة privFlag = 0، فتعني أنه قد تم تطبيق التوثيق. ويُسمح بجميع القيم لهذين العلمين باستثناء (privFlag = 1، وauthFlag = 0)؛ أي أن التشفير بدون التوثيق غير مسموح به.

- نموذج أمن الرسالة (msgSecurityModel): هو مُعرِّف تتراوح قيمته من 0 إلى $(2^{31}-1)$ ويشير إلى نموذج الأمن الذي استخدمه المُرسِل لإعداد تلك الرسالة ومن ثَمّ نموذج الأمن الذي يجب أن يستخدمه المُستقبِل لمعالجة تلك الرسالة. القيم المحجوزة تشمل 1 لـ SNMPv1، و2 لـ SNMPv2c (SNMPv2 مع تجمُّع SNMPv1)، و3 لـ SNMPv3.

❖ نموذج أمن المُستخدِم (USM)

يعرِّف RFC 2574 نموذج أمن المُستخدِم (USM)، وهو يوفِّر خدمات التوثيق والخصوصية لـ SNMP. وعلى وجه التحديد، تم تصميم USM للتأمين ضد التهديدات الرئيسة الآتية:

- تعديل المعلومات: يمكن أن يغيِّر كيانٌ ما رسالة عابرة تم إنشاؤها بواسطة كيان مرخّص له، على نحو يتسبب في عمليات إدارة غير مرخّصة، بما في ذلك تغيير قيم الكائنات. ويتمثل جوهر هذا التهديد في إمكانية تغيير كيان غير مرخّص له أيّ مُعامِل من مُعامِلات الإدارة، بما في ذلك المُعامِلات ذات الصلة بالتهيئة والإجراءات والمحاسبة.

- انتحال هوية كيان آخر (masquerade): يمكن أن يقوم كيانٌ ما بمحاولة إجراء عمليات إدارة غير مرخّصة له بواسطة انتحال هوية كيان آخر مرخّص له.
- تعديل تدفق الرسائل: تم تصميم بروتوكول SNMP لكي يعمل على بروتوكول نقل لاتوصيلي. حيث يوجد خطر من إمكانية تأخير رسائل SNMP أو إعادة ترتيبها أو إعادة إرسالها لإجراء عمليات إدارة غير مرخّصة. فعلى سبيل المثال، يمكن أن يتم نسخ رسالة بدء تشغيل (reboot) جهاز ما ثم تكرارها فى وقتٍ لاحقٍ.
- إفشاء البيانات: يمكن لكيانٍ ما مراقبة التبادلات بين مدير ووكيل ومن ثَمَّ معرفة قيم الكائنات المُدارة ومعرفة الأحداث المُبلَّغ عنها. على سبيل المثال، قد يتمكن مهاجمٌ ما عن طريق مراقبة الأمر Set الذي يقوم بتغيير كلمات السر من معرفة كلمات السر الجديدة.

لا يهدف نموذج أمن المُستخدِم (USM) إلى التأمين ضد التهديدين الآتيين:

- حجب الخدمة: قد يمنع مهاجمٌ ما التبادلات بين مدير ووكيل.
- تحليل حركة المرور: قد يراقب مهاجمٌ ما النمط العام لحركة المرور بين المديرين والوكلاء.

يمكن تبرير عدم وجود إجراء مضاد لحالات حجب الخدمة بالسببين الآتيين:

1. يتعذر تمييز هجمات حجب الخدمة في كثير من الأحيان عن حالات فشل الشبكة والتي تواجهها جميع برامج إدارة الشبكة وتتعامل معها كأمر اعتيادي.
2. من المُرجح أن يعطل هجوم حجب الخدمة جميع أنواع التبادلات وهذا أمر يخص أمن الشبكة ككل، وليس خاصاً ببروتوكول إدارة الشبكة.

أما بالنسبة لتحليل حركة المرور، فإن كثيراً من أنماط حركة مرور إدارة الشبكة يمكن التنبؤ بها (فمثلاً، قد تُدار كيانات معيّنة بواسطة أوامر SNMP

الصادرة بانتظام من محطة إدارة واحدة أو من عدد من محطات الإدارة) ولذلك لا توجد ميزة كبيرة للحماية ضد مراقبة هذه الأنماط من حركة المرور.

❖ وظائف التشفير

تم تعريف وظيفتي تشفير لنموذج أمن المُستخدِم (USM) هما: التوثيق والتشفير. ولدعم هاتين الوظيفتين، يحتاج محرك SNMP إلى قيمتين: مفتاح خاص (privKey) ومفتاح توثيق (authKey). يتم الاحتفاظ بقيمتين منفصلتين لهذين المفتاحين للمُستخدِمين الآتيين:

- المستخدمون المحليون: وهم الكيانات الرئيسة (principals) بمحرك SNMP هذا والمرخّص لهم بعمليات الإدارة.
- المستخدمون البعيدون: وهم الكيانات الرئيسة (principals) بمحرك SNMP البعيد والمطلوب الاتصال بها.

تُعدُّ هذه القيم خصائص للمُستخدِم ويتم تخزينها لكل مُستخدِم ذي صلة. ولا يمكن الحصول بواسطة SNMP على قيم المفتاح الخاص (privKey) أو مفتاح التوثيق (authKey).

يسمح USM باستخدام أحد بروتوكولي التوثيق البديلة: HMAC-MD5-96 وHMAC-SHA-96. يستخدم HMAC – والذي تم شرحه في الفصل الثالث – دالة تحوير آمنة ومفتاحاً سرياً لإنتاج شفرة توثيق لرسالةٍ ما. بالنسبة لـ HMACMD5-96 يتم استخدام HMAC مع MD5 كدالة التحوير الأساسية. يتم استخدام مفتاح للتوثيق (authKey) مكوَّن من 128 بتاً (16 بايتاً) . كمُدخَل إلى خوارزمية HMAC. تنتج الخوارزمية مُخْرَجاً طوله 128 بتاً يتم بتره إلى 96 بتاً. وبالنسبة لـ -HMAC-SHA) (96، فإن دالة التحوير الأساسية هي SHA-1، وطول مفتاح التوثيق authKey 20 بايتاً، وتنتج الخوارزمية مُخْرَجاً طوله 20 بايتاً يتم بتره أيضاً إلى 12 بايتاً.

وللتشفير يستخدم USM نمط سلسلة كتل الشفرة (CBC) لخوارزمية DES. ويتم توفير مفتاح خاص (privKey) طوله 16 بايتاً كمُدخَل إلى بروتوكول التشفير.

وتُستخدم الـ 64 بتاً الأولى من قيمة privKey كمفتاح DES. ولأن معيار تشفير البيانات DES يتطلب مفتاحاً بطول 56 بتاً يتم تجاهل البت الأقل وزناً من كل بايت. ويتطلب نمط سلسلة كتل الشفرة مُتجَه تهيئة (IV) طوله 64 بتاً. تستخدم الـ 64 بتاً الأخيرة من المفتاح الخاص (privKey) لتوليد قيمة هذا المتَّجَه.

❖ المحركات السيادية وغير السيادية

في أي عملية انتقال رسالة بين كيانين، يتم تعيين أحدهما (المرسل أو المستقبل) كمحرك SNMP سيادي، وفقاً للقواعد الآتية:

- عندما تحتوي رسالة SNMP على حمولةٍ تتوقع رداً ما (على سبيل المثال: Get، أو GetNext،أو GetBulk، أو Set، أو PDU Inform)، يُعدُّ مُستقبِل مثل هذه الرسائل سيادياً.
- عندما تحتوي رسالة SNMP على حمولةٍ لا تتوقع رداً (على سبيل المثال وحدة بيانات البروتوكول لإخطار SNMPv2 تلقائي أو استجابة أو تقرير)، يُعدُّ مُرسِل مثل هذه الرسالة سيادياً.

وهكذا، بالنسبة للرسائل المُرسَلة نيابةً عن مولِّد للأوامر أو لرسائل الإبلاغ من مُنشئ الإخطارات، فإن المُستقبِل يكون سيادياً. وبالنسبة للرسائل المُرسَلة نيابةً عن مجيب الأوامر أو رسائل الإخطارات التلقائية من مُنشئ الإخطارات، فإن المُرسِل يكون سيادياً. يخدم هذا التعيين هدفين هما:

- يتم تحديد التوقيت للرسالة بواسطة ساعة يرعاها المحرك السيادي. فعندما يرسل محرك سيادي رسالة (إخطار تلقائي، استجابة، تقرير)، فإنها تحتوي على القيمة الحالية للساعة الخاصة به، وهكذا يمكن للمُستقبِل غير السيادي أن يتزامن مع تلك الساعة. وعندما يقوم محرك غير سيادي بإرسال رسالة (Get، GetNext، GetBulk، Set، Inform)، فإنها تتضمن تقديراته الحالية لقيمة الوقت عند الوجهة، مما يسمح للوجهة بتقييم ملاءمة توقيت الرسالة.

- تُمكِّن عملية توطين المفاتيح (key localization) – والتي سيرد وصفها لاحقاً – أحد الكيانات الرئيسة من امتلاك المفاتيح المخزنة في عدة محركات؛ يتم توطين هذه المفاتيح بالمحرك السيادي بحيث يكون الكيان الرئيس مسؤولاً عن مفتاح واحد مع تجنب المخاطر الأمنية لتخزين نسخ متعددة من نفس المفتاح في شبكة موزَّعة.

من الطبيعي أن يتم تعيين المُستقبل لمولِّد الأوامر ووحدات بيانات البروتوكول للإبلاغ كمحرك سيادي، ومن ثَمَّ مسؤول عن التحقق من توقيت الرسالة. فإذا حدث تأخير استجابة أو إخطار تلقائي أو تم إعادة إرسالهما يكون الضرر الناتج ضئيلاً. بيد أن وحدات بيانات البروتوكول لمُولِّد الأوامر – وإلى حدٍّ ما وحدات بيانات البروتوكول للإبلاغ – تؤدي إلى عمليات إدارة، مثل قراءة أو تغيير قيم كائنات MIB. لذا فإنه من المهم ضمان أن مثل تلك الوحدات لا يتم تأخيرها أو إعادة إرسالها، الأمر الذي قد يؤدي إلى تأثيراتٍ غير مرغوب فيها.

❖ مُعامِلات (بارامترات) رسائل USM

عندما يتم تمرير رسالة صادرة إلى USM بواسطة معالج الرسائل، يقوم USM بتعبئة المُعامِلات المتصلة بالأمن في ترويسة الرسالة. وعندما يتم تمرير رسالة واردة إلى USM بواسطة معالج الرسائل، يقوم USM بمعالجة القيم الواردة في تلك الحقول. والمُعامِلات المتصلة بالأمن هي كما يأتي:

- msgAuthoritativeEngineID: هو مُعرِّف محرك SNMP لمحرك SNMP السيادي المشارك في تبادل هذه الرسالة. وعلى هذا، فإن هذه القيمة تشير إلى المصدر لرسالة Trap أو Response أو Report، وإلى الوجهة لرسالة Get أو GetNext أو GetBulk أو Set أو Inform.
- msgAuthoritativeEngineBoots: هو قيمة المُعامِل snmpEngineBoots لمحرك SNMP السيادي المشارك في تبادل هذه الرسالة. أما قيمة الكائن snmpEngineBoots فهي عدد صحيح يقع بين 0 و(2^{31}-1) ويمثل عدد المرات

التي قام فيها محرك SNMP هذا بالتهيئة أو إعادة التهيئة منذ تهيئته أول مرة.

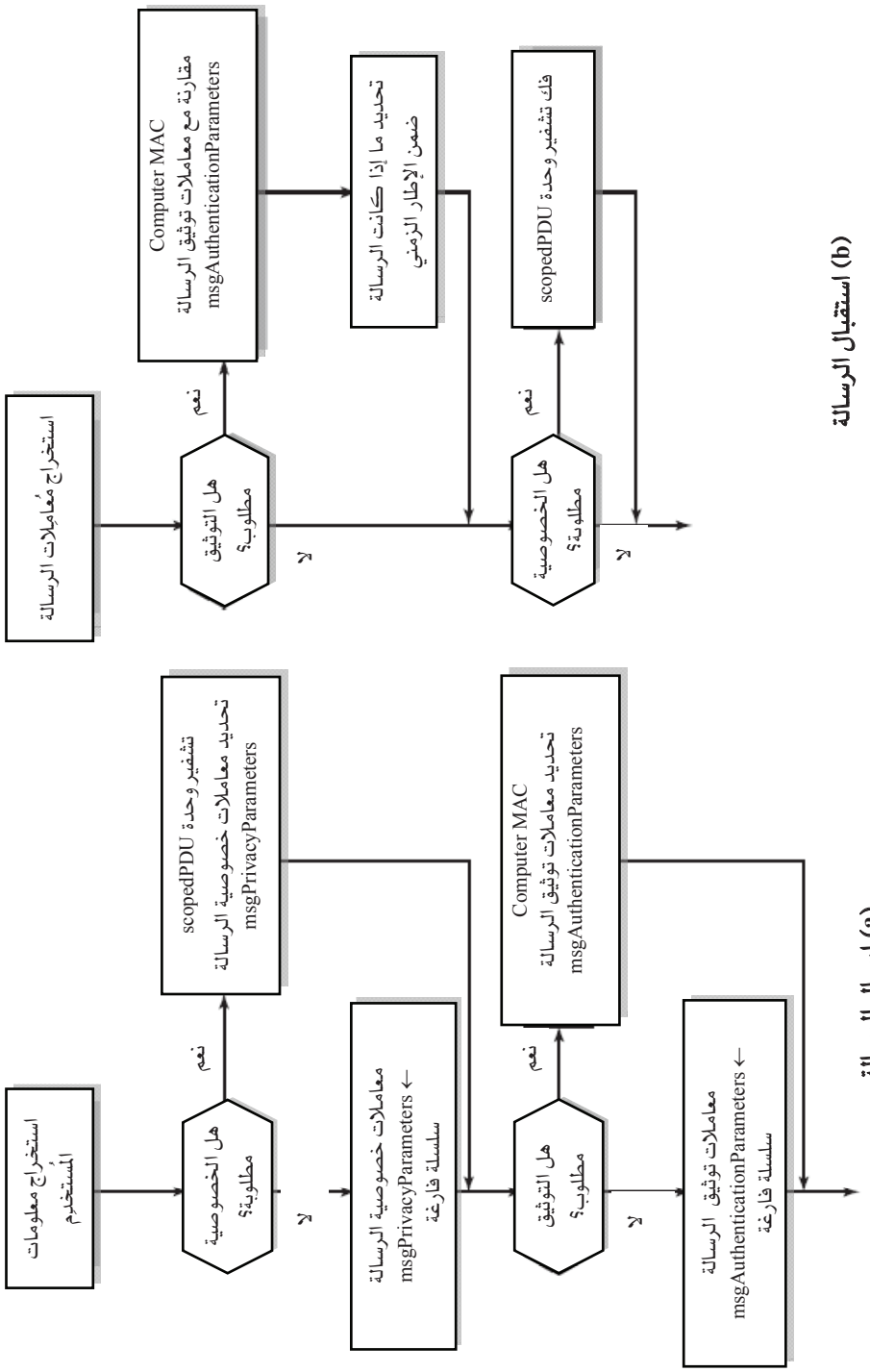
- msgAuthoritativeEngineTime: قيمة المُعامِل snmpEngineTime لمحرك SNMP السيادي المشارك في تبادل هذه الرسالة. أما قيمة الكائن snmpEngineTime فهي عدد صحيح يقع بين 0 و(2^{31}-1) ويمثل عدد الثواني منذ آخر إضافة قام بها محرك SNMP السيادي لقيمة هذا الكائن. يُعدُّ كل محرك SNMP سيادي مسؤولاً عن زيادة قيمة وقت محرك SNMP الخاص به (snmpEngineTime) مرة واحدة كل ثانية. يُعدُّ المحرك غير السيادي مسؤولاً عن زيادة القيمة التقديرية لمعامل snmpEngineTime الخاص بكل محرك سيادي بعيد يتواصل معه.
- msgUserName: هو اسم المُستخدِم (الكيان الرئيس) الذي يتم تبادل الرسالة نيابةً عنه.
- msgAuthenticationParameters: تكون قيمته لاشيء (null) في حالة عدم استخدام التوثيق لهذا التبادل. وإلا، يُعدُّ هذا مُعامِل توثيق. بالنسبة للتعريف الحالي لنموذج أمن المُستخدِم (USM)، مُعامِل التوثيق هو شفرة HMAC لتوثيق الرسالة.
- msgPrivacyParameters: تكون قيمته لاشيء (null) في حالة عدم كون هذا التبادل خصوصياً. وإلا، يُعدُّ هذا مُعامِل خصوصية. بالنسبة للتعريف الحالي لنموذج أمن المُستخدِم (USM)، تُستخدم قيمة مُعامِل الخصوصية لتحديد القيمة الأوَّليَّة (IV) في خوارزمية تسلسل كتل الشفرة (CBC) للمعيار القياسي لتشفير البيانات (DES).

ويلخص الشكل 8-10 طريقة عمل نموذج أمن المُستخدِم (USM). بالنسبة لنقل الرسالة، يتم إجراء التشفير أولاً، إذا لزم الأمر. يتم تشفير وحدة بيانات البروتوكول المُحدَّدة بنطاق ووضعها في حمولة الرسالة، وتعيين قيمة مُعامِل خصوصية الرسالة (msgPrivacyParameters) إلى القيمة المطلوبة لتوليد القيمة الأوَّليَّة IV، ثم يتم تنفيذ التوثيق إذا دعت الحاجة. يتم استخدام الرسالة بأكملها بما في ذلك وحدة بيانات

البروتوكول المُحدَّدة بنطاق كمُدخَل لـ HMAC، ثم توضع شفرة التوثيق الناتجة في مُعامِل توثيق الرسالة (msgAuthenticationParameters). بالنسبة للرسائل الواردة يتم تنفيذ التوثيق أولاً إذا دعت الحاجة. يفحص USM أولاً شفرة التوثيق الواردة مقارنةً مع شفرة توثيق يقوم بحسابها؛ فإذا تطابقت القيمتان، يفرض أن الرسالة موثَّقة (أي آتية من المصدر المفترض ولم تتغير أثناء الإرسال). ثم يقوم USM بالتحقق من كون الرسالة وصلت في مدى زمني صحيح، كما سنوضّح لاحقاً. إذا لم يكن توقيت الرسالة مناسباً، يتم تجاهلها على اعتبار أنها غير موثَّقة. وأخيراً، إذا كان قد تم تشفير وحدة بيانات البروتوكول، يقوم USM بإزالة التشفير واستعادة النص الأصلي.

❖ آليات التوقيت في USM

يتضمن USM مجموعة من آليات التوقيت للوقاية ضد تأخير أو إعادة إرسال الرسالة. ويجب على كل محرك SNMP يمكنه العمل في أي وقت كمحرك سيادي أن يحتفظ بالكائنين snmpEngineBoots وsnmpEngineTime اللذين يقومان بتعريف وقته المحلي. عند أول تثبيت لمحرك SNMP، يتم تصفير قيمة هذين الكائنين. بعدها تتم زيادة قيمة snmpEngineTime مرة واحدة كل ثانية. إذا وصلت قيمة snmpEngineTime في أي وقت إلى الحد الأقصى (2^{31}-1)، يتم زيادة قيمة snmpEngineBoots كما لو أن النظام قد أعيد تشغيله، ويعاد تصفير قيمة snmpEngineTime وزيادة قيمته كل ثانية مرة أخرى. باستخدام آلية تزامن، يقوم كل محرك غير سيادي بالاحتفاظ بتقدير لقيم الوقت لكل محرك سيادي يتواصل معه. وتوضع هذه القيم المُقدَّرة في كل رسالة صادرة لتُمكِّن المحرك السيادي المُستقبِل للرسالة من تحديد ما إذا كان توقيت الرسالة الواردة مناسباً أو لا.



الشكل 8-10: معالجة رسالة USM.

تعمل آلية التزامن بالطريقة الآتية. يحتفظ محرك غير سيادي بنسخة محلية من متغيرات ثلاث خاصة بكل محرك SNMP سيادي معروف لهذا المحرك:

- snmpEngineBoots: أحدث قيمة للمُعامِل snmpEngineBoots الخاص بالمحرك السيادي البعيد.
- snmpEngineTime: قيمة تقديرية من هذا المحرك للمُعامِل snmpEngineTime الخاص بالمحرك السيادي البعيد. وتتم مزامنة هذه القيمة مع المحرك السيادي البعيد بواسطة عملية التزامن التي سنوضحها فيما بعد. تتم زيادة هذه القيمة بين أحداث المزامنة بشكلٍ منطقي مرة كل ثانية للحفاظ على تزامن غير دقيق مع المحرك السيادي البعيد.
- latestReceivedEngineTime: أعلى قيمة للمُعامِل msgAuthoritativeEngineTime التي تلقاها هذا المحرك من المحرك السيادي البعيد؛ ويتم تحديث هذه القيمة كلما وردت قيمة أكبر لها. والغرض من هذا المتغير هو الحماية ضد هجوم إعادة الإرسال الذي قد يمنع تقدم قيمة snmpEngineTime التقديرية لمحرك SNMP غير السيادي.

يتم الحفاظ على مجموعة واحدة من هذه المتغيرات الثلاث لكل محرك سيادي بعيد معروف لهذا المحرك. وتتم المحافظة بشكلٍ منطقي على القيم في نوع من ذاكرة التخزين المخبأة (Cache) والمعنونة بواسطة مُعرِّف محرك SNMP الفريد لكل محرك سيادي بعيد (snmpEngineID).

ولتمكين المحركات غير السيادية من الحفاظ على تزامن الوقت، يُدرِج كل محرك سيادي القيم الحالية للتشغيل (boot) والوقت الخاصة به، وكذلك قيمة مُعرِّف محرك SNMP (snmpEngineID)، بكل رسالة استجابة أو تقرير أو إخطار صادرة في الحقول msgAuthoritativeEngineBoots، وmsgAuthoritativeEngineTime، وmsgAuthoritativeEngineID. إذا كانت الرسالة موثَّقة وكانت ضمن الإطار الزمني المضبوط، يقوم المحرك غير السيادي المُستلِم بتحديث المتغيرات المحلية

snmpEngineBoots، وsnmpEngineTime، وlatestReceivedEngineTime لذلك المحرك البعيد وفقاً للقواعد الآتية:

1. يتم التحديث إذا تحقق أحد الشرطين الآتيين على الأقل:
   - msgAuthoritativeEngineBoots > snmpEngineBoots
     أي عدد مرات التشغيل لمحرك سيادي > عدد مرات التشغيل لمحرك SNMP
   - msgAuthoritativeEngineBoots = snmpEngineBoots AND msgAuthoritativeEngineTime > latestReceivedEngineTime
     أي عدد مرات التشغيل لمحرك سيادي = عدد مرات التشغيل لمحرك SNMP، ووقت محرك سيادي > آخر وقت محرك تم استقباله

ينص الشرط الأول على وجوب التحديث إذا زادت قيمة مُعامِل التشغيل (msgAuthoritativeEngineBoots) من قِبَل المحرك السيادي منذ آخر تحديث. وينص الشرط الثاني على أنه إذا لم تتم زيادة قيمة معامل التشغيل، فيجب التحديث إذا كانت قيمة وقت المحرك الواردة أكبر من آخر قيمة لوقت المحرك تم تلقيها. وتكون قيمة الوقت الواردة للمحرك أقل من آخر قيمة وقت للمحرك تم تلقيها في حالة وصول رسالتين بترتيب عكسي؛ وهذا أمر وارد الحدوث. كما يمكن حدوث ذلك أيضاً في حالة هجوم إعادة الإرسال. في كلتا الحالتين، لن يقوم المحرك المتلقي بالتحديث.

2. إذا وجب التحديث، يتم إجراء التغييرات الآتية:
   - msgAuthoritativeEngineBoots → snmpEngineBoots
   - msgAuthoritativeEngineTime → snmpEngineTime
   - msgAuthoritativeEngineTime → latestReceivedEngineTime

إذا قلبنا المنطق، فسوف نرى أنه في حالة msgAuthoritativeEngineBoots < snmpEngineBoots، لا يتم التحديث. مثل هذه الرسالة تُعدُّ غير موثَّقة ويجب تجاهلها. وفي حالة msgAuthoritativeEngineBoot = snmpEngineBoots، ولكن في

نفس الوقت latestReceivedEngineTime > msgAuthoritativeEngineTime، لا يتم التحديث أيضاً. في هذه الحالة، قد تكون الرسالة موثّقة إلا أنها قد تكون بترتيب خاطئ، وفي هذه الحالة يكون تحديث snmpEngineTime غير مبرَّر.

لاحظ أن توظيف التزامن يتم فقط إذا كانت خدمة التوثيق مُستخدَمة في هذه الرسالة وأنه قد تم توثيق الرسالة باستخدام HMAC. هذا التقييد ضروري لأن نطاق التوثيق يشمل msgAuthoritativeEngineID، وmsgAuthoritativeEngineBoots، وmsgAuthoritativeEngineTime، ومن ثَمَّ يمكن ضمان كون هذه القيم صالحة.

ينص SNMPv3 على أنه يجب استلام الرسالة ضمن إطار زمني مقبول لتجنب التأخير وهجمات إعادة الإرسال. ويجب اختيار الإطار الزمني صغيراً بقدر الإمكان بالنظر إلى دقة الساعات المستعملة والتأخير المصاحب للاتصال ذهاباً وإياباً ومعدل مزامنة الساعات. وفي حال تعيين إطار زمني صغير جداً، سوف يتم رفض الرسائل الموثَّقة على اعتبار أنها غير موثَّقة،ومن ناحية أخرى فإن تعيين إطار زمني كبير يجعل الرسائل أكثر عرضة للتأخيرات الناتجة عن هجمات خبيثة.

سندرس الحالة الأكثر أهمية المتعلقة بمُستقبِل سيادي (يختلف اختبار التوقيت قليلاً في حالة المُستقبِل غير السيادي). مع كل رسالة واردة موثَّقة وقيمة مُعامِلها msgAuthoritativeEngineID مساوية لقيمة snmpEngineID لهذا المحرك، يقوم المحرك بمقارنة قيمة المُعامِلين msgAuthoritativeEngineBoots وmsgAuthoritativeEngineTime بالرسالة الواردة مع قيمة المُعامِلين snmpEngineBoots وsnmpEngineTime اللتين يحتفظ بهما هذا المحرك. تعتبر الرسالة الواردة خارج الإطار الزمني إذا تحقق أي من الشروط الآتية:

- snmpEngineBoots = 2^{31} - 1
- snmpEngineBoots ≠ msgAuthoritativeEngineBoots
- قيمة msgAuthoritativeEngineTime تختلف عن قيمة snmpEngineTime بأكثر من ± 150 ثانية

ينص الشرط الأول على أنه إذا تم تخزين snmpEngineBoots عند قيمته القصوى، فلا يمكن اعتبار أي رسالة واردة رسالةً موثقة. وينص الشرط الثاني على أنه يجب أن يكون للرسالة وقت تشغيل مُساوياً لوقت تشغيل المحرك المحلي؛ على سبيل المثال، في حالة إعادة تشغيل المحرك المحلي وعدم قيام المحرك البعيد بالمزامنة مع المحرك المحلي منذ إعادة التشغيل، تُعدُّ الرسائل من ذلك المحرك البعيد غير موثَّقة. وينص الشرط الأخير على أن الوقت المُتضمَّن في الرسالة الواردة يجب أن يكون أكبر من الوقت المحلي ناقص 150 ثانية وأقل من الوقت المحلى زائد 150 ثانية.

في حالة اعتبار رسالةٍ ما على أنها خارج الإطار الزمني، تُعدُّ الرسالة غير موثَّقة، ويتم إرجاع إشارة خطأ خارج إطار الوقت (notInTimeWindow) إلى الوحدة المتصلة (calling module).

ومرة أخرى، كما هو الحال مع التزامن، فإن التحقق من مناسبة الوقت يتم فقط في حالة استخدام خدمة التوثيق وعندما تكون الرسالة موثَّقة، مما يضمن صلاحية حقول ترويسة الرسالة.

❖ توطين المفاتيح (Key Localization)

المطلب الرئيس لاستخدام خدمات التوثيق والخصوصية في SNMPv3 هو أنه لا بد لأي اتصال بين كيان رئيس (principal) على محرك غير سيادي ومحرك سيادي بعيد من وجود مفتاحين سريين مُشتركين لعمليتي التوثيق والخصوصية، يُمكِّن هذان المفتاحان مُستخدِماً ما بمحرك غير سيادي (في العادة نظام إدارة) من استخدام التوثيق والخصوصية مع الأنظمة السيادية البعيدة التي يديرها المُستخدِم (في العادة أنظمة الوكلاء). يوفِّر المرجع RFC 2574 مبادئ توجيهية لإنشاء تلك المفاتيح وتحديثها وإدارتها.

لتبسيط عبء إدارة المفاتيح على الكيانات الرئيسة (principals)، يُطلب فقط من كل كيان رئيس الاحتفاظ بمفتاح توثيق واحد ومفتاح تشفير واحد. ولا يتم

تخزين هذه المفاتيح في MIB ولا يمكن الوصول إليها عبر SNMP. وفيما يلي سنتناول أولاً أسلوب توليد هذه المفاتيح من كلمة السر. ثم نستعرض مفهوم توطين المفاتيح، والذي يُمكِّن الكيان الرئيس من مشاركة مفتاح توثيق وتشفير فريد مع كل محرك بعيد مع الاحتفاظ فقط بمفتاح توثيق وتشفير واحد محلياً.

يتطلب المُستخدِم مفتاحاً خاصاً مُكوَّناً من 16 بايتاً ومفتاح توثيق مُكوَّناً من 16 أو 20 بايتاً. بالنسبة للمفاتيح التي يملكها مُستخدِمون من البشر، من المرغوب فيه أن يكون المُستخدِم قادراً على توظيف كلمة سر مقروءة بشرياً بدلاً من مفتاح مُكوَّن من سلسلة من البتات. ويُعرِّف المرجع RFC 2574 خوارزمية لاستخلاص مفتاح بطول 16 أو 20 بايتاً من كلمة السر للمُستخدِم. ولا يضع (USM) قيوداً على كلمة السر نفسها، ولكن ينبغي لسياسات الإدارة المحلية أن تفرض على المُستخدِمين استعمال كلمات سر لا يمكن تخمينها بسهولة.

يتم استخلاص المفتاح من كلمة السر كما يأتي:

- اتخاذ كلمة السر الخاصة بالمُستخدِم كمُدخل وتوليد سلسلة بطول 2^{20} بايتاً (1,048,576 بايتاً) بتكرار قيمة كلمة السر عدة مرات حسب الضرورة، وبتر القيمة الأخيرة إذا دعت الضرورة، لتشكيل سلسلة الحروف digest0. على سبيل المثال، إذا كانت كلمة السر مكونة من 8 أحرف (2^3 بايتات) فسوف تُسلسل مع نفسها 2^{17} مرة لتشكيل digest0.
- في حال الرغبة في استعمال مفتاح مكوَّن من 16 بايتاً، تُستخدم دالة التحوير MD5 لتشكيل digest1 من digest0. أما في حال مفتاح مكوَّن من 20 بايتاً، تُستخدم دالة التحوير SHA-1 لتشكيل digest1 من digest0. ويكون الناتج هنا هو مفتاح المُستخدِم.

من مزايا هذا الأسلوب أنه يبطئ إلى حدٍّ كبير هجوم القاموس الاستقصائي (brute-force dictionary attack)، والذي يقوم فيه الخَصْم بتجربة كلمات سر مختلفة كثيرة محتملة لتوليد المفتاح من كل واحدة من تلك الكلمات، ومن ثَمَّ اختبار ما إذا كان المفتاح الناتج يعمل مع بيانات التوثيق أو التشفير المتاحة له. فعلى

سبيل المثال، لو أن مهاجماً ما اعترض رسالة موثَّقة، يمكنه أن يحاول توليد قيمة HMAC مع مختلف مفاتيح المُستخدِم الممكنة. وفي حالة حدوث مطابقة، يمكن للمهاجم افتراض أنه تم اكتشاف كلمة السر. فالعملية المكونة من خطوتين والتي تم ذكرها تؤدي إلى زيادة كبيرة في الوقت الذي قد يستغرقه مثل هذا الهجوم.

ميزة أخرى لهذا الأسلوب هو أنه يعزل مفاتيح المُستخدِم عن نوعية نظام إدارة الشبكة (NMS). فلا يحتاج نظام إدارة الشبكة إلى تخزين قيم مفاتيح المُستخدِم. بدلاً من ذلك، يتم توليد مفتاح المُستخدِم من كلمة السر عند الحاجة. ويسرد [BLUM97b] الاعتبارات التالية التي تحفّز استخدام طريقة كلمة السر بشكلٍ مستقل عن نظام إدارة الشبكة:

- إذا أُريد للمفتاح أن يُخزّن بدلاً من أن يتم توليده من كلمة السر، فإن أحد البدائل هو الاحتفاظ بمستودع مركزي للمفاتيح السرية. ولكن هذا يؤثر سلباً على الاعتمادية الكلية للنظام كما يمكن أن يجعل عملية اكتشاف الأخطاء وإصلاحها أمراً مستحيلاً إذا كان المستودع نفسه لا يمكن الوصول إليه عند الحاجة.
- ومن جهة أخرى، إذا تم الحفاظ على عدة نسخ من المستودع، فإن هذا يهدد الأمن الشامل بتوفير مواطن اقتحام أكثر للمهاجمين المحتملين.
- في حالة استخدام مستودع مركزي أو عِدة نسخ منه، يجب المحافظة عليه في مواقع آمنة. قد يقلل ذلك من فرصة إقامة "موقع متقدم" أثناء التصدي للهجوم (أي، التدخل للإصلاح عندما تتعطل قطاعات غير متوقعة من الشبكة أو عندما لا يمكن الوصول إليها لفترات من الزمن لا يمكن التنبؤ بها).

ويمكن استخدام كلمة سر واحدة لتوليد مفتاح واحد يُستخدم لكلٍّ من عمليتي التشفير والتوثيق. الأسلوب الأكثر أماناً هو استخدام كلمتي سر، إحداهما لإنشاء مفتاح توثيق والأخرى لإنشاء مفتاح تشفير مختلف.

تم تعريف المفتاح المحلي في المرجع RFC 2574 على أنه مفتاح سري مشترك بين مُستخدِم ومحرك SNMP سيادي واحد. والهدف هو احتفاظ المُستخدِم بمفتاح واحد فقط (أو مفتاحين اثنين عند الحاجة لكلٍّ من التوثيق والخصوصية) ومن ثَمَّ يحتاج فقط لتذكُّر كلمة سر واحدة (أو اثنتين). تختلف المعلومات السرية الفعلية المشتركة بين مُستخدِم معين وكل محرك من محركات SNMP السيادية. ويُشار إلى العملية التي يتم بواسطتها تحويل مفتاح مُستخدِم واحد إلى عدة مفاتيح فريدة (واحد لكل محرك SNMP بعيد)، بعملية توطين المفتاح. يستعرض [BLUM97a] الدافع وراء هذه الاستراتيجية، والتي سنلخصها هنا.

يمكننا تحديد الأهداف الآتية لعملية إدارة المفاتيح:

- لكل نظام وكيل SNMP في شبكة موزعة مفتاحٌ فريدٌ خاصٌ بكل مُستخدِم مرخّص له بإدارة هذا الوكيل. إذا تم الترخيص لعدة مُستخدِمين كمديرين، يكون لدى الوكيل مفتاح توثيق فريد ومفتاح تشفير فريد لكل مُستخدِم. وهكذا، إذا تم اختراق مفتاح أحد المُستخدِمين، فلا يؤثر ذلك على مفاتيح المُستخدِمين الآخرين.
- تختلف مفاتيح مُستخدِم واحد بالنسبة للوكلاء المختلفين. وهكذا، إذا تعرض وكيل ما للاختراق، فإن مفاتيح المُستخدِم لهذا الوكيل فقط هي التي تتعرض للخطر بمعزلٍ عن مفاتيحه الأخرى مع الوكلاء الآخرين.
- يمكن إدارة الشبكة من أي نقطة فيها، بغض النظر عن توافر نظام إدارة الشبكة مسبق التهيئة (NMS). ويسمح ذلك للمُستخدِم بأداء وظائف إدارية من أي محطة إدارة. ويتم توفير هذه الإمكانية بواسطة خوارزمية "كلمة السر إلى مفتاح" المشروحة سابقاً.

يمكننا أيضاً تعريف الأمور الآتية كأشياء ينبغي تجنبها:

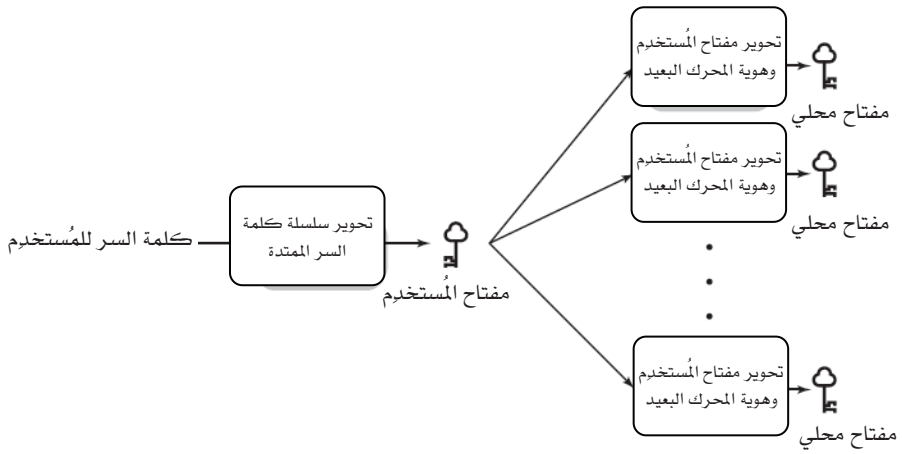
- على المُستخدِم تذكُّر (أو بمعنى آخر إدارة) عدد كبير من المفاتيح، وهو عدد يتنامى مع إضافة وكلاء جدد لإدارتهم.

- الخَصْم الذي يكتشف مفتاحاً لوكيل واحد يعد قادراً حينئذ على انتحال هوية أي وكيل آخر بالنسبة لأي مُستخدِم، أو أي مُستخدِم بالنسبة لأي وكيل آخر.

استجابةً للأهداف والاعتبارات السابقة، يتم باستخدام دالة أحادية الاتجاه غير قابلة للعكس (أي دالة تحوير آمنة) لتحويل مفتاح مُستخدِم واحد إلى عدة مفاتيح محلية مختلفة لمحركات موثَّقة مختلفة (وكلاء مختلفين).

ويتم هذا الإجراء كما يأتي:

- تُشكَّل السلسلة digest2 عن طريق سَلسَلة digest1 (تم وصفه سابقاً) مع قيمة snmpEngineID للمحرك السيادي مع digest1.
- إذا كان الهدف مفتاحاً مكوناً من 16 بايتاً، تُستخدم دالة التحوير MD5 مع digest2. وإذا كان المفتاح المرغوب بطول 20 بايتاً، تُستخدم دالة التحوير SHA-1 مع digest2. ويكون المُخرَج هو المفتاح المحلي للمُستخدِم.



الشكل 8-11: توطين المفاتيح.

يمكن بعدها تهيئة المفتاح المحلي الناتج على نظام الوكيل بطريقةٍ آمنة. ونظراً للطبيعة أحادية الاتجاه لدالتي MD5 وSHA-1، فإنه من غير المعقول (شبه المستحيل) لخَصْمٍ ما اكتشاف مفتاح المُستخدِم حتى لو نجح هذا الخَصْم في اكتشاف مفتاح محلي. يلخص الشكل 8-11 عملية توطين المفاتيح.

8-3-3 التحكم في الوصول المبنيّ على المشهد

التحكم في الوصول هو وظيفة أمنية تُؤدَّى على مستوى وحدة بيانات البروتوكول (PDU). تُعَرِّف وثيقة التحكم في الوصول آلياتٍ لتحديد ما إذا كان ينبغي السماح لكيان رئيس (principal) بعيد بالوصول لكائن مُدار في قاعدة معلومات إدارة (MIB) محلية. ويمكن تصور وتحديد آليات متعددة للتحكم في الوصول. تُعرِّف وثائق SNMPv3 نموذجاً للتحكم في الوصول المُعتمِد على المشهد (VACM). يستفيد VACM نفسه من MIB والتي تُعرِّف سياسة التحكم في الوصول لهذا الوكيل وتجعل من الممكن استخدام التهيئة عن بُعد.

يتسم نموذج VACM بخاصتين مهمتين:

- يحدِّد VACM ما إذا كان ينبغي السماح لكيان رئيس (principal) بعيد بالوصول إلى كائن مُدار في MIB محلية.
- يستفيد VACM من قاعدة معلومات إدارة MIB والتي:
  - تُعرِّف سياسة التحكم في الوصول إلى ذلك الوكيل.
  - تجعل التهيئة عن بُعد ممكنة الاستخدام.

❖ عناصر نموذج VACM

يُعَرِّف المرجع RFC 2575 خمسة عناصر تؤلِّف النموذج VACM: المجموعات (groups)، مستوى الأمن، السياقات (contexts)، ومشاهد MIB، وسياسة الوصول.

تم تعريف "المجموعة" كسلّة مكونة من صفر أو أكثر من الثنائي <securityModel, securityName> (أي <نموذج أمن، واسم أمن>) التي يمكن نيابةً

عنها الوصول إلى كائنات إدارة SNMP. يشير اسم الأمن securityName إلى كيان رئيس (principal) ، مع العلم أن جميع الرؤساء في مجموعة معيّنة لهم نفس حقوق الوصول. ويخصّص لكل مجموعة اسم فريد groupName. يُستخدم مفهوم المجموعة كأداة مفيدة لتصنيف المديرين فيما يتعلق بحقوق الوصول. فعلى سبيل المثال، قد يكون لجميع مديري المستوى الأعلى سلة واحدة من حقوق الوصول، بينما قد يكون لمديري المستوى المتوسط سلة مختلفة من حقوق الوصول.

يمكن لأي تشكيلة معيّنة من قيم securityModel وsecurityName أن تنتمي إلى مجموعة واحدة على الأكثر. أي أنه لهذا الوكيل ، لا يمكن أن يدرج أي كيان رئيس (principal) محدَّد له اتصالات محميَّة بنموذج أمن محدَّد (securityModel) إلا في مجموعة واحدة فقط.

تختلف حقوق الوصول لمجموعةٍ ما استناداً إلى مستوى الأمن للرسالة التي تحتوي على الطلب. على سبيل المثال، يمكن أن يسمح وكيلٌ ما بحق القراءة فقط لطلب مُرسَل في رسالة غير موثَّقة ولكن للسماح لها بحق الكتابة قد تحتاج إلى توثيق أولاً. وعلاوة على ذلك، في حالة بعض الكائنات الحساسة، قد يطلب الوكيل أن يتم إرسال الطلب والاستجابة له باستخدام خدمة الخصوصية.

يُعرَّف "سياق MIB" بمجموعة فرعية لبعض مثيلات (instances) الكائنات في الـ MIB المحلية باسم معين. توفر السياقات طريقة مفيدة لتجميع الكائنات في مجموعات لها سياسات مختلفة للوصول.

يتصل مفهوم السياق بعملية التحكم في الوصول. عندما تتفاعل محطة إدارة مع وكيل للوصول إلى معلومات الإدارة لدى الوكيل ، حينئذ يتم التفاعل بين الكيان الرئيس (principal) للإدارة ومحرك SNMP للوكيل، ويتم التعبير عن امتيازات التحكم في الوصول بمشهد MIB ينطبق على هذا الكيان الرئيس وذاك السياق. وتتمتع السياقات بالخصائص الرئيسة الآتية:

- قد يحتوي كيان SNMP – المُعرَّف بشكلٍ فريد بواسطة هوية محرك السياق contextEngineID – على أكثر من سياق واحد.

- قد يظهر الكائن أو مثيله في أكثر من سياق واحد.
- لتعريف مثيل كائن – في حال وجود سياقات متعددة – يجب تعريف اسم السياق contextName، وهوية محرك السياق contextEngineID الخاص به، بالإضافة إلى نوع الكائن ومثيله.

كثيراً ما نود تقييد وصول مجموعة معيّنة إلى كائنات بمجموعة فرعية في الوكيل. ولتحقيق هذا الهدف، يكون الوصول إلى سياق بواسطة مشهد MIB (VACM)، والذي يُعرِّف مجموعة معيّنة من الكائنات المُدارة (واختيارياً مثيلات كائنات محدَّدة). يستخدم VACM تقنية قوية ومرنة لتعريف مشاهد MIB، استناداً إلى مفاهيم مشهد الأشجار الفرعية ومشهد العائلات. ويُعرَّف مشهد MIB عن طريق مجموعة أو عائلة من الأشجار الفرعية، حيث يمكن إدراج أيٍّ من تلك الأشجار الفرعية أو استبعادها من المشهد.

يتم تنظيم الكائنات المُدارة في قاعدة بيانات محلية في تسلسل هرمي، أو شجري، استناداً إلى مُعرّفات الكائنات. تضم هذه القاعدة المحلية للبيانات مجموعة فرعية من جميع أنواع الكائنات المُعرَّفة وفقاً لمعيار الإنترنت الخاص ببنية معلومات الإدارة (Structure of Management Information (SMI)) ويتضمن مثيلات للكائنات التي تتفق مُعرّفاتها مع اصطلاحات SMI.

يتضمن SNMPv3 مفهوم الشجرة الفرعية. الشجرة الفرعية ببساطة هي عقدة في التسلسل الهرمي للتسمية والخاص بقاعدة معلومات الإدارة MIB بالإضافة إلى جميع العناصر المندرجة تحتها. ويمكن تعريف شجرة فرعية بشكلٍ اصطلاحي (رسمي) باعتبارها المجموعة التي تضم جميع الكائنات ومثيلات الكائنات التي لديها بادئة ASN.1 مشتركة في أسمائها. تُعدُّ البادئة المشتركة الأطول لجميع المثيلات في الشجرة الفرعية هي هوية الكائن (object identifier) للعقدة الأصل بتلك الشجرة الفرعية.

ترتبط ثلاثة مشاهد MIB مع كل مُدخَل في جدول وصول VACM (vacmAccessTable)، مشهد لكل وصول قراءة وكتابة وإخطار. ويتألَّف كل

مشهد MIB من مجموعة من مشاهد الشجرات الفرعية. ويتم تحديد كل مشهد شجرة فرعية في مشهد MIB من حيث كونه مضمَّناً أو مستبعداً. أي أن مشهد MIB إما يتضمن أو يستبعد كل مثيلات الكائنات الواردة في تلك الشجرة الفرعية. وبالإضافة لذلك، يتم تعريف قناع للمشهد لتقليل كمية معلومات التهيئة عندما يتطلب الأمر تحكماً دقيقاً وتفصيلياً للوصول (مثل التحكم في الوصول على مستوى مثيل الكائن).

يُمكِّن VACM من تهيئة محرك SNMP لفرض مجموعة معيّنة من حقوق الوصول، والتي تُشكِّل سياسة وصول. يعتمد تحديد الوصول على العوامل الآتية:

- يقدم الكيان الرئيس طلباً للوصول. يجعل VACM من الممكن للوكيل أن يخصّص امتيازات وصول مختلفة لمُستخدِمين مختلفين. فعلى سبيل المثال، قد تُمنح سلطات واسعة لتغيير عناصر في قاعدة معلومات الإدارة MIB المحلية لنظام مدير مسؤول عن تهيئة شبكة بأكملها، بينما قد يكون لأحد مديري المستويات الوسطى مسؤولية مراقبة الشبكة مع حق الوصول للقراءة فقط، وعلاوة على ذلك قد يكون الوصول مقصوراً على مجموعة فرعية فقط من MIB المحلية. وكما تمت مناقشته سابقاً، يتم تعيين الكيانات الرئيسة للمجموعات ويتم تحديد سياسة الوصول للمجموعة.
- مستوى الأمان الذي أُرسِل به الطلب في رسالة SNMP. حيث يطلب الوكيل عادةً استخدام التوثيق للرسائل التي تحتوي على طلب تعيين قيم (أي عملية كتابة).
- نموذج الأمن المُستخدَم في معالجة رسالة الطلب. إذا ما تم تنفيذ عدة نماذج للأمن في وكيلٍ ما، يمكن تهيئة الوكيل ليوفر مستويات مختلفة من الوصول للطلبات المُقدَّمة من رسائل تمت معالجتها بواسطة نماذج أمن مختلفة. فعلى سبيل المثال، قد تكون بعض العناصر قابلة للوصول إليها إذا كان طلب الرسالة قد أتى عبر نموذج أمن المُستخدِم (USM)، ولكن غير قابلة للوصول إليها إذا كان نموذج الأمن SNMPv1.
- سياق MIB للطلب.

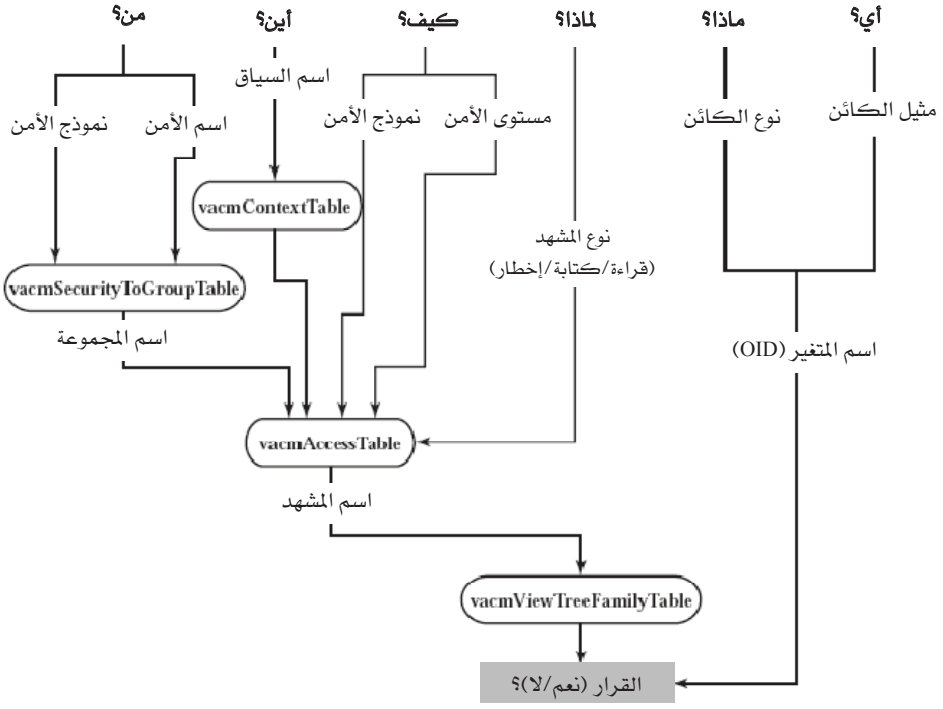
- مثيل الكائن المحدَّد المطلوب الوصول إليه. تمتلك بعض الكائنات معلومات أكثر خطورة أو حساسية مقارنةً بغيرها، ولذلك يجب أن تعتمد سياسة الوصول على المثيل المحدَّد المطلوب.
- **نوع** الوصول المطلوب (قراءة، كتابة، إخطار). فالقراءة والكتابة والإخطار عمليات إدارة مختلفة، ويمكن تطبيق سياسات مختلفة للتحكم في الوصول لكلٍّ منها.

❖ معالجة التحكم في الوصول

يقوم تطبيق SNMP باستدعاء نموذج التحكم في الوصول المبنيّ على المشهد (VACM) عن طريق isAccessAllowed، مع المُدخلات نموذج الأمن (securityModel) واسم الأمن (securityName) ومستوى الأمن (securityLevel) ونوع المشهد (viewType) واسم السياق (contextName) واسم المتغير (variableName). ونحتاج لجميع قيم هذه المُعامِلات لاتخاذ قرار التحكم في الوصول. أي أن النظام الفرعي للتحكم في الوصول قد تم تعريفه كأداة عالية المرونة لتهيئة التحكم في الوصول لدى الوكيل ، عن طريق تقسيم مكوِّنات قرار التحكم في الوصول إلى ستة متغيرات منفصلة.

يعرض الشكل 8-12 – المقتبس من شكل في مقالة طلب تعليقات RFC2575 – طريقة مفيدة للنظر إلى متغيرات الإدخال ويبيّن كيف أن الجداول المتنوعة في قاعدة معلومات الإدارة في نموذج VACM تؤثر في اتخاذ قرار التحكم في الوصول.

- مَنْ (who): وهو عبارة عن مزيج المُعامِلين securityModel وsecurityName، والذي يُعرِّف "مَنْ" لهذه العملية، كما يحدِّد هوية الكيان الرئيس الذي تكون اتصالاته محميّة بنموذج أمن (securityModel) معين. وينتمي هذه المزيج إلى مجموعة واحدة على الأكثر في محرك SNMP. أما الجدول vacmSecurityToGroupTable فيُعرِّف اسم المجموعة (groupName) لكل مزيجٍ من المُعامِلين securityModel وsecurityName.



الشكل 8-12: منطق نموذج التحكم في الوصول المبنيّ على المشهد (VACM).

- أين (where): يحدِّد اسم السياق (contextName) "أين" يمكن العثور على كائن الإدارة المطلوب. ويحتوي جدول سياق VACM (vacmContextTable) على قائمة بأسماء السياقات (contextNames) التي يتم التعرف عليها.
- كيف (how): وهو عبارة عن مزيج من المُعامِلين securityModel وsecurityLevel والذي يُعرِّف كيف تمت حماية الطلب الوارد أو PDU للإخطار. ويحدِّد المزيج الثلاثي "مَنْ"، و"أين"، و"كيف" صفاً واحداً على الأكثر يقابل هذا المزيج في جدول وصول VACM (vacmAccessTable).
- لماذا (why): يحدِّد مُعامل نوع المشهد (viewType) لماذا تم طلب الوصول لعملية قراءة أو كتابة أو إخطار. يحتوي الصف المُختار في جدول وصول VACM (vacmAccessTable) على اسم مشهد واحد (viewName) لقاعدة

معلومات الإدارة لكل نوع من هذه الأنواع الثلاثة من العملية، ويستخدم مُعامِل نوع المشهد (viewType) لاختيار اسم مشهد (viewName) محدَّد، والذي يختار بدوره مشهد MIB المناسب من جدول مشهد شجرة عائلة VACM (vacmViewTreeFamilyTable).

- ماذا (what): يُستخدم المُعامِل variableName كمُعرِّف كائن والذي تحدد بادئته نوعية هذا الكائن كما تحدد لاحقته مثيلاً معيناً. وتُشير نوعية الكائن إلى ماذا تكون نوعية معلومات الإدارة المطلوبة.
- ماهية (which): يشير مثيل الكائن إلى ماهية العنصر المحدَّد المطلوب من المعلومات.

وأخيراً، تتم مقارنة اسم المتغير (variableName) مع مشهد MIB الذي تم التوصل إليه. وإذا تطابق اسم المتغير مع عنصر ضمن مشهد MIB، يتم منح الوصول.

❖ الدوافع وراء مفاهيم VACM

يبدو أن المفاهيم التي تؤلِّف VACM تؤدِّي إلى تعريف معقد نوعاً ما للتحكم في الوصول. أما دوافع تقديم هذه المفاهيم فهي توضيح العلاقات المشاركة في معلومات إدارة الوصول والتقليل إلى أدنى حدّ من متطلبات التخزين والمعالجة لدى الوكيل. ولفهم تلك الدوافع، لاحظ أنه في SNMPv1 يُستخدَم مفهوم التجمُّع لتمثيل معلومات الأمن الآتية:

- هوية الكيان الطالب (محطة الإدارة).
- هوية الكيان المُنفِّذ (وكيل يمثل نفسه أو ينوب عن كيان مُفوِّض).
- هوية موقع معلومات الإدارة المراد الوصول إليها (وكيل أو كيان مُفوِّض).
- معلومات التوثيق.
- معلومات التحكم في الوصول (الترخيص بإجراء العملية المطلوبة).
- معلومات مشهد MIB.

وينتج عن دمج كل تلك المفاهيم في متغير واحد فقدان المرونة والفاعلية. يوفّر VACM نفس المعلومات الأمنية باستخدام متغيرات مميزة لكل عنصر. ويُعدُّ ذلك تحسناً جوهرياً مقارنةً بـ SNMPv1 . حيث يؤدي فصل هذه المفاهيم المتعددة إلى إمكانية تعيين كل متغير بشكلٍ مستقل.

4-8 توصيات للمطالعة

يتضمن المرجع [STAL99] دراسة شاملة ومفصَّلة عن SNMP وSNMPv2 وSNMPv3؛ كما يقدم نظرة عامة على تقنية إدارة الشبكة.

[STAL99] Stallings, W. *SNMP, SNMPv2, SNMPv3, and RMON 1 and 2*. Reading, MA: Addison-Wesley, 1999.

5-8 مصادر للمعلومات على الويب

- موقع ويب SNMPv3: تحتفظ به "جامعة براونشفايغ التقنية". وهو يوفر وصلات إلى المراجع ومسودات الإنترنت ونسخاً من التوضيحات والتغييرات المقترحة التي أُرسِلت بواسطة الفريق العامل ووصلات تشعبية إلى البائعين مع تطبيقات SNMPv3.
- موقع The Simple Web: تحتفظ به جامعة تفينتي. وهو مصدر جيد للمعلومات المتعلقة بـ SNMP، بما في ذلك المؤشرات إلى كثير من تطبيقات النطاق المفتوح وقوائم من الكتب والمقالات.

6-8 مصطلحات رئيسة

Access Policy	سياسة الوصول، سياسة النفاذ
Agent	وكيل
Community	التجمُّع
Community Name	اسم التجمُّع
Key Localization	توطين المفاتيح
Management Information Base (MIB)	قاعدة معلومات الإدارة
Management Station	محطة الإدارة
Message Processing Model	نموذج معالجة الرسالة
Network Management Proxy	مُفوَّض إدارة الشبكة
Simple Network Management Protocol (SNMP)	بروتوكول إدارة الشبكة البسيط
User Security Model	نموذج أمن المُستخدِم
View-based Access Control Model (VACM)	نموذج التحكم في الوصول المبنيّ على المشهد

8-7 أسئلة للمراجعة ومسائل

8-7-1 أسئلة المراجعة

8-1 ما الزوايا التي يمكن من جانبها اعتبار البنية الهيكلية لإدارة الشبكة متكاملة؟

8-2 ما العناصر الرئيسة في نموذج SNMP؟

8-3 ما المقصود بـ MIB؟

8-4 ما القدرات الأساسية أو الأوامر التي يتم توفيرها في SNMPv1؟

8-5 ما وظيفة مُفوَّض SNMP؟

8-6 وضّح باختصار مفهوم التجمُّع في SNMPv1.

8-7 ما العلاقة بين SNMPv1 وSNMPv2 وSNMPv3؟

8-8 ما التهديدات التي تم تصميم USM لمواجهتها؟

8-9 ما الفرق بين محرك سيادي (authoritative) ومحرك غير سيادي (nonauthoritative)؟

8-10 ما المقصود بتوطين المفاتيح؟

8-11 اسرد وعرّف باختصار العناصر التي تؤلّف VACM.

8-7-2 مسائل

8-1 يعرّف SNMPv1 نوع بيانات يُشار إليه كمقياس (gauge) ويتضمن الشرح الآتي دلالات هذا النوع:

> يُمثّل هذا النوع عدداً صحيحاً غير سالب، والذي قد يزيد أو ينقص،وتغلق (latches) قيمة المقياس عند القيمة القصوى، وقد حدد هذا المعيار القيمة القصوى للمقاييس بـ $2^{32} - 1 = 4294967295$.

وللأسف، لم يتم تعريف المقصود بالكلمة latches في المعيار، وأدى ذلك إلى تفسيرين مختلفين. أزال معيار SNMPv2 الالتباس بالتعريف الآتي:

> تتخذ قيمة المقياس الحد الأقصى عندما تكون المعلومات التي يجري نمذجتها أكبر من أو تساوي تلك القيمة القصوى؛ أما إذا انخفضت المعلومات التي يجري نمذجتها فيما بعد إلى أقل من قيمة الحد الأقصى، ينقص المقياس أيضاً.

a. ما التفسير البديل لذلك؟

b. ناقش إيجابيات وسلبيات كلا التفسيرين.

2-8 في SNMPv1، أي كائن يتم تعريفه في MIB يكون له فئة وصول يمكن تعيين قيمتها لأيٍّ من القيم الآتية: "القراءة فقط"، أو "القراءة والكتابة"، أو "الكتابة فقط"، أو "ممنوع الوصول". ويتم إنجاز القراءة عن طريق Get أو عن طريق Trap، ويتم إنجاز الكتابة عن طريق Set. وللكتابة فقط، قد يكون الكائن متاحاً لعمليتي Get وTrap، ولكن طريقة العمل هنا تعتمد على كيفية التنفيذ العملي. وتحدد فئة وصول MIB الحد الأقصى للوصول الذي يمكن أن يُسمح به لأحد الكائنات، ولكن في تسهيلات تجمُّع SNMPv1، يمكن لأسلوب الوصول أن يقصر هذا الوصول على تجمعات لها خصائص معيّنة. وفي الجدول الآتي قم بملء كل مُدخَل موضحاً الوصول المسموح به.

فئة وصول MIB	نمط وصول SNMP	
	قراءة فقط	قراءة وكتابة
قراءة فقط		
قراءة وكتابة		
كتابة فقط		
ممنوع الوصول		

3-8 a. يوضّح RFC 2574 أنه بالنسبة لمحرك غير سيادي (nonauthoritative)، يتم تعيين قيم msgAuthoritativeEngineBoots وmsgAuthoritativeEngineTime في ترويسة رسالة صادرة فقط إذا كانت الرسالة يتم توثيقها بواسطة المُستقبِل السيادي. لماذا يعدّ هذا التقييد ذا معنى؟

b. ومع ذلك، بالنسبة لرسالة استجابة من محرك سيادي، يتم تعيين قيم msgAuthoritativeEngineBoots وmsgAuthoritativeEngineTime دائماً في ترويسة الرسالة الصادرة. اشرح أسباب ذلك.

4-8 يحدِّد RFC 2574 أن مزامنة الساعة (تحديث الساعة المحلية بناءً على القيم الواردة) يحدث قبل التحقق من الإطار الزمني (التحقق من أن الرسالة الواردة في التوقيت المناسب). وهذا يعني أنه قد يقوم محرك غير سيادي بتحديث القيمة التقديرية لساعة

المحرك السيادي في حال كانت الرسالة المُستلَمة موثَّقة حتى إذا لم تكن الرسالة ضمن الإطار الزمني. كان هناك جدل مستمر حول ذلك في قائمة مراسلات SNMPv3 منذ صدور طلب التعليقات، ولكن يبدو أن الصياغة في المعيار لم تتغير حتى كتابة هذه السطور. ومن المفيد أن ننظر في الآثار المترتبة على ذلك. بافتراض التعريفات الآتية:

MAEB = msgAuthoritativeEngineBoots
MAET = msgAuthoritativeEngineTime
LRET = latestReceivedEngineTime
SET = local notion of snmpEngineTime for the remote authoritative engine
SEB = local notion of snmpEngineBoots for the remote authoritative engine

ثم لنفترض محركاً غير سيادي يستقبل رسالة بحيث

(MAEB = SEB) AND [LRET < MAET < (SET-150)]

ثم تتوافر الشروط لحصول تحديث الساعة، ومن ثَمَّ يتم التحديث:

SET := MAET; LRET := MAET

الآن، عندما نصل إلى اختبار الإطار الزمني، لدينا

(MAEB = SEB) AND (MAET = SET)

لذلك نعلن أن الرسالة في التوقيت المناسب. ومع ذلك، افترض أنه كان علينا أولاً القيام باختبار إطار الوقت. أكان علينا إعلان أن الرسالة في التوقيت المناسب أو ليست في التوقيت المناسب؟

5-8 في النسخة الأصلية المنشورة لمواصفات نموذج أمن المُستخدِم (USM) رقم RFC 2274، والتي تناقش مزامنة الساعة وإجراءات التحقق من الإطار الزمني، ورد التعليق الآتي: "لاحظ أن هذا الإجراء لا يسمح بالمزامنة التلقائية للوقت إذا كان محرك SNMP غير السيادي لديه حالة حقيقية من عدم التزامن أدت إلى أن يتأخر محرك SNMP سيادي أكثر من 150 ثانية عن محرك SNMP غير السيادي." وقد تم إسقاط هذا البيان من النسخة المعدلة رقم RFC 2574 بعد أن أوضح مؤلف هذا الكتاب لمجموعة العمل أن البيان ليس صحيحاً دائماً، وذكر المثال الذي جاء في المسألة 8-4 للتدليل على ذلك.

6-8 يفترض SNMPv3 أن هناك بعض الوسائل الآمنة لتسليم المفاتيح المحلية إلى أنظمة (وكيل) موثَّقة. تُعدُّ عملية التسليم الآمن هذه خارج نطاق SNMPv3؛ حيث يمكن أن يكون التسليم إما يدوياً أو بواسطة بروتوكول آمن آخر. وبمجرد أن يتم تسليم مفتاح أوَّليّ (أو زوج من المفاتيح للتوثيق والخصوصية) إلى وكيل ما، فإن SNMPv3 يوفر آلية لتحديث المفاتيح بشكلٍ آمن. ومن المرغوب فيه أن يتم تغيير المفاتيح من وقت لآخر لتعزيز

الأمن. يمكن لأحد المُستخدِمين الشروع في عملية تغيير المفتاح، عن طريق طلب كلمة سر جديدة وإدخالها. وبدلاً من ذلك، يمكن لنظام إدارة شبكة (NMS) بدء العملية بواسطة طلب كلمة سر جديدة. وفي كلتا الحالتين، يتم تحديث مفتاح المُستخدِم في NMS. حينئذ يمكن لنظام إدارة شبكة (NMS) اعتماد مفتاح محلي لكل وكيل اتصال.

يجب أن يتصل NMS بعد ذلك بشكلٍ آمن بكل وكيل ليحثه على تحديث مفتاحه المحلي. ومن الواضح أن NMS لا يمكنه ببساطة إرسال المفتاح في نص عادي عبر الشبكة. وهنا يوجد خياران:

- تشفير المفتاح الجديد باستخدام المفتاح القديم باعتباره مفتاح التشفير.
- استخدام نوع من الدوال أحادية الاتجاه لتوليد قيمة جديدة للمفتاح من قيمة المفتاح القديم. إجراء عملية أو-الحصرية (XOR) لهذه القيمة مع قيمة المفتاح الجديد وإرسال النتيجة إلى الوكيل. يستطيع الوكيل حينئذ إجراء عملية أو-الحصرية (XOR) لهذه النتيجة الواردة مع المفتاح القديم لإنتاج المفتاح الجديد.

يستخدم SNMPv3 نسخة معدلة من الطريقة الثانية. فما هي ميزة هذا النهج على الأول؟

7-8 ينطوي نهج SNMPv3 على استخدام كائن تغيير المفتاح (KeyChange) في نظام MIB المستهدف. ويقوم الكيان الرئيس البعيد أو نظام إدارة الشبكة (NMS) بضبط هذا الكائن، والذي يستخدم تلقائياً من قِبَل الوكيل لتحديث المفتاح المناظر. وتتم الخوارزمية على مرحلتين، تعمل واحدة منها في المحرك الطالب والأخرى في محرك الوكيل البعيد. تبدأ العملية عندما يرغب الطالب في تحديث مفتاح موجود (keyOld) إلى قيمة جديدة (keyNew). عندها يقوم الطالب بتنفيذ الخطوات الآتية:

1. توليد قيمة عشوائية إما من مولِّد أرقام شبه عشوائي أو مولِّد أرقام عشوائي حقيقي.
2. حساب

$$digest = Hash (keyOld \parallel random)$$

حيث دالة التحوير (Hash) هي إما MD5 أو SHA-1، بالاعتماد على ما إذا كان المطلوب مفتاح من 16 بايتاً أو 20 بايتاً، والرمز || يمثل الوصل.

3. حساب

$$delta = digest \oplus keyNew$$

حيث ⊕ هي عملية أو - الحصرية (XOR).

4. إرسال قيمة المعامل protocolKeyChange إلى الوكيل في أمر تعيين (Set) لتحديث مثيل كائن (KeyChange) في وكيل MIB.

ما الذي يجب أن يفعله الوكيل مع القيمة الواردة لتحديث المفتاح؟

8-8 الإجراء الأبسط من ذلك المبيَّن في المسألة السابقة يكون بإجراء عملية أو - الحصرية (XOR) بين قيمة المفتاح القديم (keyOld) وقيمته الجديدة (keyNew) وإرسال تلك القيمة. ثم يُجري المُستلِم عملية أو - الحصرية (XOR) بين القيمة التي تم تلقيها و(keyOld) لإنتاج المفتاح الجديد (keyNew). ولأن المهاجم لا يعرف (keyOld)، فإنه لا يستطيع استنتاج قيمة (keyNew). ما هي مزايا استخدام الرقم العشوائي ودالة التحوير الآمنة أحادية الاتجاه للمسألة 8-7 مقارنةً مع هذا النهج ؟

الباب الثالث

أمن نظام الحاسب

يتناول الباب الثالث المسائل الأمنية على مستوى النظام، بما في ذلك التهديد الذي يُمثله المتسللون والإجراءات المضادة للتسلل وللفيروسات وكذلك استخدام الجدران النارية والأنظمة الموثوقة.

خريطة الطريق للجزء الثالث

الفصل التاسع : المتسللون

يبحث الفصل التاسع مجموعة متنوعة من التهديدات المتعلقة بالخدمة أو الوصول إلى المعلومات التي تنجم عن قراصنة يستغلون نقاط الضعف في أنظمة الحاسب المبنية على الشبكات. يبدأ الفصل بمناقشةٍ لأنواع الهجمات التي يمكن أن يقوم بها المستخدمون غير المصرّح لهم، أو المتسللون، ويحلل الطرق المختلفة للوقاية من تلك الهجمات واكتشافها. كما يغطي هذا الفصل المسألة ذات الصلة والمتعلقة بإدارة كلمات السر.

الفصل العاشر: البرمجيات الخبيثة

يتناول الفصل العاشر التهديدات البرمجية للنظم، مع التركيز بشكلٍ خاص على الفيروسات والديدان. ويبدأ الفصل بمسح لمختلف أنواع البرامجيات الخبيثة، ويخص – بنظرة أعمق – طبيعة الفيروسات والديدان. بعد ذلك يتناول الفصل موضوع الإجراءات المضادة، وأخيراً يناقش الهجمات الموزّعة للحرمان من الخدمة.

الفصل الحادي عشر : الجدران النارية

يُعدُّ استخدام جدار ناري أحد الوسائل القياسية لحماية الأصول المحلية لنظام الحاسب من خطر التهديدات الخارجية. يتناول الفصل الحادي عشر مبادىء تصميم الجدار الناري ويستعرض تقنيات محددة في هذا الصدد. كما يغطي ذلك الفصل المسألة ذات الصلة والمتعلقة بالنظم الموثوقة.

9

الفصل التاسع

المتسللون (INTRUDERS)

محتويات الفصل:

9-1 المتسللون

9-1-1 أساليب التسلل

9-2 اكتشاف التسلل

9-2-1 سجلات المراجعة

9-2-2 اكتشاف الشذوذ الإحصائي

9-2-3 اكتشاف التسلل بناءً على قواعد

9-2-4 مغالطة المعدل الأساسي

9-2-5 الأنظمة الموزّعة لاكتشاف التسلل

9-2-6 المصايد الأمنية

9-2-7 صيغة تبادل البيانات لاكتشاف التسلل

9-3 إدارة كلمات السر

9-3-1 حماية كلمة السر

9-3-2 استراتيجيات اختيار كلمة السر

9-4 توصيات للمطالعة

9-5 مصادر للمعلومات على الويب

9-6 مصطلحات رئيسة

9-7 أسئلة للمراجعة ومسائل

الملحق 9-A: مغالطة المعدل الأساسي

اتفقوا على أن يتولى جراهام وضع الاختبار لتشارلز مابليدين. كان الاختبار أن يحصل دراجون على شفرة شتيرن – لا أكثر ولا أقل. فلو كان لديه الـ "in" الموجودة في "Utting" كما يدّعى فسيكون ذلك ممكناً، ولا يمكن أن يحول دون ذلك إلا الولاء لمحطة موسكو المركزية فقط. أما اذا حصل على مفتاح الشفرة فسيثبت ولاءه لمحطة لندن المركزية بدون أدنى شك.

— الحديث إلى رجال غرباء، روث رينديل

النقاط الرئيسة

- يُعدُّ التسلل إلى نظام أو شبكة حاسب واحداً من أخطر التهديدات لأمن أنظمة الحاسب.
- تم تطوير أنظمة اكتشاف التسلل لتوفير الإنذار المبكر عند حدوث تسلل كي يتسنى اتخاذ الإجراءات الدفاعية الكفيلة بمنع الضرر أو تقليله من جرّاء ذلك التسلل.
- يتضمن اكتشاف التسلل الكشف عن أنماط غير عادية من النشاط أو أنماط من النشاط معروف أنها ترتبط بعملية التسلل.
- تُعدُّ إدارة كلمات السر أحد العناصر المهمة لمنع التسلل، وذلك بهدف منع المستخدمين غير المرخّص لهم من الوصول إلى كلمات السر الخاصة بالمستخدمين الآخرين.

من المشكلات الأمنية الكبيرة لأنظمة الشبكات محاولات التسلل المعادية، أو على الأقل غير المرغوب فيها، سواءً من قِبَل مستخدمين أو من قِبَل برمجيات. أما التسلل من قِبَل المستخدمين فقد يأخذ شكل دخول غير مصرّح به (unauthorized logon) إلى أحد الأجهزة، أو – في حالة المُستخدِم المصرّح له – قد يتمثل في حصوله على امتيازات أو قيامه بأعمال تتجاوز تلك المرخّص له بها. أما التسلل من قِبَل البرمجيات فقد يتمثّل في صورة فيروس، أو دودة، أو حصان طروادة.

تتعلق كل تلك الهجمات بأمن الشبكة لأن الدخول إلى النظام يمكن أن يتم عن طريق الشبكة. ومع ذلك، لا تقتصر تلك الهجمات فقط على هجمات أساسها الشبكة. فقد يحاول مستخدم لديه إمكانية للوصول إلى محطة طرفية محلية التسلل إلى النظام دون اللجوء إلى شبكة وسيطة. كما يمكن إدخال فيروس أو حصان طروادة إلى النظام عن طريق قرص مرن. الدودة هي الظاهرة الوحيدة التي ترتبط فقط بالشبكات. وعليه، فإن التسلل إلى النظام يمثل المجال المشترك الذي يلتقي فيه أمن الشبكة مع أمن الحاسب.

نظراً لأن هذا الكتاب يركّز أساساً على أمن الشبكات، فلن نحاول القيام بتحليل شامل لا للهجمات ولا للإجراءات المضادة فيما يتعلق بالتسلل إلى النظام. وبدلاً من ذلك سنُلقي في هذا الباب نظرة عامة واسعة على تلك المخاوف.

يغطي هذا الفصل موضوع المتسللين. سنناقش أولاً طبيعة الهجوم ، ثم نتناول الاستراتيجيات المستخدمة لمنعه، أو اكتشافه إذا تعذر ذلك. بعدها نغطي الموضوع ذا الصلة والمتعلق بإدارة كلمات السر.

9-1 المتسللون

يشكّل المتسللون أحد أشد خطرين يهددان أمن أنظمة الحاسب (الخطر الآخر هو الفيروسات)، ويطلق عليهم بشكلٍ عام القراصنة أو مكسّرو الشفرات. في دراسة مبكرة ومهمة عن التسلل، حدد أندرسون [ANDE80] ثلاث فئات من المتسللين:

- المتنكّر (Masquerader): شخص غير مخوّل له استخدام نظام الحاسب ولكنه يخترق نظام التحكم في الدخول إلى النظام لاستغلال حساب مستخدم شرعي.
- مسيء التصرف (Misfeasor): مستخدم شرعي للنظام ولكنه يصل إلى بيانات أو برامج أو موارد غير مخوّلٍ الوصول إليها، أو أنه مخوّل له الوصول إليها ولكنه يسيء استخدام الصلاحيات المعطاة له.

- المُستخدِم السري (Clandestine user): شخص يستولي على وسائل التحكم الإشرافي على النظام ويستغل ذلك للتهرب من عمليات المراجعة والتحكم في الوصول أو لمنع عمليات جمع سجلات المراجعة.

من المرجّح أن يكون المتنكر شخصاً من خارج النظام، بينما يكون مسيء التصرف عموماً من داخله، أما المُستخدِم السري فقد يكون من الخارج أو من الداخل.

تتفاوت هجمات التسلل من الحميدة إلى الخطيرة. ففي الناحية الحميدة، يوجد كثير من الأشخاص الذين يرغبون ببساطة في مجرد استكشاف شبكات الإنترنت ليرَوا ماذا يقبع على الطرف الآخر هناك. أما في الناحية الخطيرة فهناك أشخاص يحاولون قراءة بيانات خصوصية، أو إجراء تعديلات غير مصرّح بها على البيانات، أو عرقلة عمل النظام.

لقد انتشر الحديث عن تهديدات التسلل، لا سيما إثر حادث قرصان ويلي "Wily Hacker" الشهير في عامي 1986 و1987، والذي قام بتوثيقه كليف ستول [STOL88, 89]. وقد شهد عام 1990 حملة وطنية على قراصنة الحاسب المحظورين تضمنت اعتقالات، وتهماً جنائية، ومحاكمة شهيرة لعدة جلسات، والعديد من الاعترافات، ومصادرة كميات كبيرة من البيانات ومعدات الحاسب [STER92]. وعندئذٍ اعتقد كثيرٌ من الناس أن المشكلة قد تم السيطرة عليها.

في واقع الأمر، لم تتم السيطرة على المشكلة. كمثال واحد على ذلك، أفاد فريق من مختبرات بيل (Bell Labs) [BELL93, BELL92] بوقوع هجمات متكررة وبإصرار على أنظمة الحاسب لديها عبر شبكة الإنترنت على مدى فترة زمنية طويلة ومن مصادر متنوعة. في فترة إعداد تلك التقارير، واجهت مجموعة بيل ما يأتي:

- محاولات لنسخ ملف كلمة السر (والذي سنناقشه لاحقاً)، بمعدّل تجاوز مرة كل يومين.

- نداء إجراء مشبوه عن بُعد (remote procedure call (RPC))، بمعدّل تجاوز مرة كل أسبوع.
- محاولات للاتصال بأجهزة حاسب وهمية تُستخدَم كطُعم (bait)، على الأقل مرة كل أسبوعين.

قد تكون هجمات المتسللين الحميدة مقبولة، ولكنها تستهلك موارد النظام وقد تؤدي إلى إبطاء أداء النظام للمستخدمين الشرعيين. ومع ذلك، لا توجد وسيلة لكي نعرف مسبقاً ما إذا كان هجوم المتسلل سيكون حميداً أو خبيثاً. ومن ثم فحتى بالنسبة للأنظمة التي لا تتضمن موارد حساسة بشكلٍ خاص، هناك مايبرر محاولة السيطرة على تلك المشكلة.

من الأمثلة التي توضّح بشكلٍ كبير حجم التهديد المحتمل، ما حدث في جامعة Texas A&M [SAFF93]. ففي أغسطس 1992 تم إبلاغ مركز الحاسب بالجامعة أن أحد الأجهزة يُستخدم في الهجوم على أجهزة حاسب في مكان آخر عبر الإنترنت. ومن خلال مراقبة النشاط علِم موظفو مركز الحاسب بوجود مجموعة من المتسللين الخارجيين يقومون بتشغيل برامج لفك شفرات كلمات السر على عددٍ من أجهزة الحاسب (يضم الموقع ما مجموعه 12,000 جهاز موصلة فيما بينها بشبكات). وقام المركز بفصل الأجهزة المتضررة وسد الفجوات الأمنية المعروفة ثم استأنف العمل كالمعتاد. وبعد بضعة أيام، اكتشف أحد مديري النظام المحليين أن هجوم التسلل قد استُؤنف. واتضح عندئذٍ أن الهجوم كان أكثر تعقيدا مما كان يُظنّ في البداية. وتم العثور على ملفات تحتوي على المئات من كلمات السر التي تم الاستحواذ عليها، بما في ذلك بعض الكلمات الخاصة بخوادم رئيسة من المُفترض أنها آمنة. وبالإضافة إلى ذلك، كان القراصنة قد اتخذوا من أحد الأجهزة المحلية لوحة إعلانات (bulletin board) خاصة بهم يستخدمونها للاتصال فيما بينهم ولمناقشة الأساليب التي يستخدمونها والتقدم الذي يحرزونه.

تبيّن من تحليل ذلك الهجوم أنه كان هناك بالفعل مستويان من القراصنة. تضمّن المستوى الأعلى المستخدمين المحترفين الذين كانوا على معرفة دقيقة

بالتقنية والأساليب المستخدمة، في حين تضمّن المستوى الأدنى "جنود المشاة" الذين اقتصر دورهم فقط على استخدام برامج تكسير الشفرات دون الحاجة إلى فهم عميق لكيفية عملها. ولقد جمع فريق العمل هذا أخطر سلاحيْن في ترسانة المتسللين: معرفة متطورة بكيفية التسلل من جهة، والاستعداد لقضاء الساعات الطويلة في محاولات "لفتح مقابض الأبواب" بحثاً عن نقاط الضعف من جهة أخرى.

كان من نتائج الوعي المتزايد بمشكلة التسلل إنشاء عدد من فرق الاستجابة لطوارئ الحاسب (computer emergency response teams (CERTs)). تقوم تلك المشاريع التعاونية بجمع المعلومات عن نقاط الضعف في الأنظمة ونشرها بين مديري النظم. وللأسف، يمكن للقراصنة أيضاً الوصول إلى تقارير CERT. وفي حادث جامعة Texas A&M، أظهرت التحاليل التي أُجريت لاحقاً أن القراصنة قاموا بتطوير برامج لاختبار الأجهزة التي تعرضت للهجوم بحثاً عن كل نقاط الضعف تقريبا التي سبق وأُعلن عنها في تقارير CERT. في حال عدم الاستجابة على الفور للتعليمات التي يوصي بها تقرير CERT، حتى ولو على جهازٍ واحدٍ، فإنه يصبح عُرضةً لمثل تلك الهجمات.

بالإضافة إلى تشغيل برامجَ لتكسير كلمات السر، حاول المتسللون تعديل برمجيات التحكم في الدخول (login software) لتمكينهم من الاستيلاء على كلمات السر الخاصة بالمستخدمين أثناء دخولهم إلى النظام. وقد مكّنهم ذلك من حشد مجموعة مُبهرة من كلمات السر التي تمكنوا من الاستيلاء عليها وجعلوها متاحةً على لوحة إعلانات أقاموها على أحد الأجهزة ضمن النظام الضحية.

سنُلقي في هذا الباب نظرةً على الأساليب المستخدمة للتسلل، ثم ننظر في الطرق المختلفة المستخدمة لاكتشاف التسلسل، وأخيراً نتناول أساليب منع التسلسل المبنية على كلمة السر.

9-1-1 أساليب التسلل

يهدف المتسلل إلى الدخول على نظام أو توسيع نطاق الامتيازات المصرّح له بها على النظام. وعموماً، يتطلب ذلك حصول المتسلل على معلومات من المُفترض أن تكون محميَّة. وفي بعض الحالات، تأخذ تلك المعلومات شكل كلمة السر لمستخدم. وبمعرفة كلمة السر لمستخدم شرعي، يتمكّن المتسلل من الدخول على النظام وممارسة جميع الامتيازات الممنوحة لذلك المستخدم. عادةً، يجب أن يحتفظ النظام بملف يربط مابين كل كلمة سر والمُستخدِم الشرعي الذي يستخدمها. إذا كان هذا الملف مُخزّناً بدون حماية، فسيكون من السهل الوصول إليه ومعرفة كلمات السر منه. ويمكن حماية ملف كلمات السر بإحدى طريقتين:

- استخدام دالة أحادية الاتجاه: يخزّن النظام فقط قيمة دالة مبنية على كلمة سر المستخدِم. وعندما يقوم المُستخدِم بإدخال كلمة السر للدخول على النظام، يُحول النظام كلمة السر تلك باستخدام نفس الدالة ويقارن النتيجة بالقيمة المخزنة. عملياً، يستخدم النظام عادةً دالة باتجاه واحد، (أي غير انعكاسية (not reversible) تُستخدم فيه كلمة السر لتوليد مفتاح للدالة ويتم إنتاج مُخرج له طول ثابت.
- التحكم في الوصول للملف: تُقصر صلاحية الوصول إلى ملف كلمات السر على واحد أو عدد قليل جداً من الحسابات على النظام.

إذا استُخدمت إحدى الطريقتين أعلاه أو كلاهما، فسيتطلب الأمر من متسلل حريص بذل بعض الجهد لمعرفة كلمات السر. على أساس دراسة الأدبيات ومقابلات أُجريت مع هواة اكتشاف كلمات السر، يذكر [ALVA90] الأساليب التالية لمعرفة كلمات السر:

1. جرّب استخدام كلمات السر الافتراضية (default) للحسابات القياسية التي تُشحن مع النظام. كثيرٌ من مديري الأنظمة لا يكلّفون أنفسهم عناء تغيير تلك الإعدادات الافتراضية.

2. جرّب بشكلٍ استقصائي جميع كلمات السر القصيرة التي تتكوّن من واحد إلى ثلاثة أحرف.
3. جرّب الكلمات المستخدمة في قاموس النظام على شبكة الإنترنت أو في قائمة كلمات السر المُحتملة. وتوجد أمثلة للكلمات من النوع الأخير متاحة على لوحات الإعلانات الخاصة بالقراصنة.
4. اجمع معلومات عن المستخدمين، كأسمائهم بالكامل، وأسماء الزوج/الزوجة والأطفال، والصور في مكاتبهم، والكتب المتعلقة بالهوايات في مكاتبهم.
5. جرّب استخدام أرقام الهواتف، وأرقام الضمان الاجتماعي، وأرقام الغرف الخاصة بالمستخدمين.
6. جرّب جميع أرقام لوحات السيارات المشروعة في تلك الولاية.
7. استخدم حصان طروادة (سيأتي وصفه في الفصل العاشر) للالتفاف حول القيود المفروضة على الوصول.
8. تنصّت على الخط الواصل بين مستخدم عن بعد والنظام المضيف.

الأساليب الستة الأول هي طرق مختلفة لتخمين كلمة السر. وإذا كان على المتسلل التحقق من صحة تخمينه بمحاولة الدخول يدوياً على النظام، فهذا أمر مُضنٍ ويمكن إحباطه بسهولة. فعلى سبيل المثال، يمكن للنظام ببساطة رفض أي دخول على النظام بعد ثلاث محاولات لإدخال كلمة السر، مما يتطلب من المتسلل إعادة الاتصال بالمضيف لمعاودة المحاولة من جديد. وفي ظل تلك الظروف، ليس من العملي محاولة أكثر من حفنة من كلمات السر. ومع ذلك، فإنه من غير المرجّح أن يحاول المتسلل تلك الأساليب الساذجة. فعلى سبيل المثال، إذا تمكن متسلل بمستوى منخفض من الامتيازات من الوصول إلى ملف مُشفَّر لكلمات السر، فقد تتضمن خطته الاستيلاء على ذلك الملف، ثم استخدام آلية التشفير لذلك النظام وأخذ الوقت الكافي لاكتشاف كلمة سر توفر له امتيازات أكبر.

هجمات تخمين كلمات السر ممكنة، بل قد تكون فعّالة للغاية، إذا أمكن القيام بتجربة عدد كبير من التخمينات آلياً والتحقق من كل تخمين بدون الكشف عن عملية التخمين. ولاحقاً في هذا الفصل، سنورد الكثير عن إحباط هجمات التخمين.

قد يكون من الصعوبة بمكان التمكن من إحباط الأسلوب السابع للهجوم (استخدام حصان طروادة). ويتضمن [ALVA90] مثالاً لبرنامج يمكنه الالتفاف حول القيود المفروضة على الوصول. حيث قام مستخدمٌ ذو امتيازات منخفضة بتطوير برنامج ألعاب ودعا مشغّل النظام لاستخدامه للتسلية في وقت فراغه. وكان البرنامج فعلاً يشغّل لعبة، ولكن في الخلفية كان يتضمن أيضاً كوداً لنسخ ملف كلمات السر الذي كان في هذه الحالة غير مُشفَّر ولكن الوصول إليه كان محمياً. ونظراً لأن برنامج اللعبة كان يتم تشغيله في نمط الامتيازات العالية التي يتمتع بها مشغل النظام، فقد كان بمقدوره الوصول إلى ملف كلمات السر.

يتعلق الهجوم الثامن في القائمة، التنصت على الخط، بمسألة الأمن المادي. ويمكن مواجهته باستخدام أساليب تشفير الوصلة التي تناولناها في الفصل الثاني.

هناك أساليب أخرى للتسلل لا تحتاج لمعرفة كلمة سر. فبوسع المتسللين الدخول على نظام باستخدام هجمات كهجوم فيضان المخزن المؤقت (buffer overflows) على برنامج يعمل بمستوى معين من الامتيازات. كما يمكن للمتسلل محاولة ترقية مستوى الامتيازات المتاحة له بنفس الطريقة.

ننتقل الآن إلى مناقشة النوعين الأساسين من الإجراءات المضادة: الاكتشاف والوقاية. ويعني الاكتشاف العلم بالهجوم، سواءً قبل نجاحه أو بعده. أما الوقاية فهي الغاية الأمنية صعبة التحقيق والتي تمثل تحدياً قائماً في جميع الأوقات. وتكمن الصعوبة هنا في حقيقة أنه في حين يتعين على المدافع محاولة إحباط كل الهجمات الممكنة، يكون للمهاجم الحرية في محاولة للعثور على الحلقة الأضعف في سلسلة الدفاعات ومن ثم الهجوم منها.

9-2 اكتشاف التسلل

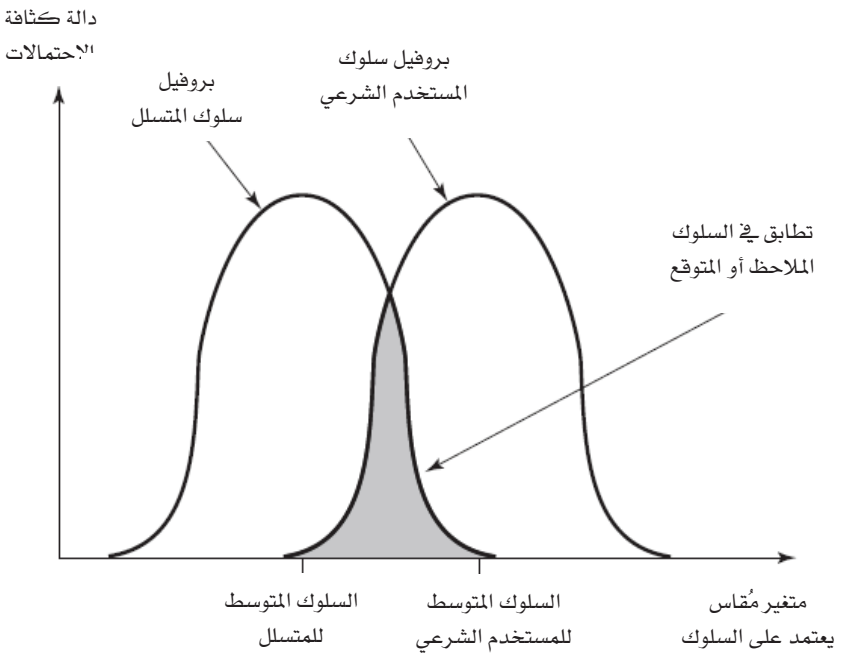
حتى أفضل نظام لمنع التسلل لا بد أن يصاب حتماً بالفشل. ويمثل اكتشاف التسلل خط الدفاع الثاني عن النظام. وكان هذا الموضوع المحور الذي دار حوله كثير من البحوث في السنوات الأخيرة، وكان الدافع وراء ذلك عدة اعتبارات نذكر منها ما يأتي:

1. إذا أمكن اكتشاف التسلل بالسرعة الكافية، فسيتسنى التعرف على المتسلل وطرده من النظام قبل أن يتمكن من إلحاق أي ضرر أو الاستيلاء على أية بيانات أو العبث بها. حتى إذا لم يتم الاكتشاف بسرعة كافية للحيلولة دون حدوث التسلل، فكلما كان الاكتشاف سريعاً، أمكن تقليل الضرر الناجم واستعادة عمل النظام بصورة طبيعية بسرعة أكبر.
2. يمكن أن يكون النظام الفعّال لاكتشاف التسلل بمثابة رادع يسهم في منع عمليات التسلل.
3. يسمح نظام اكتشاف التسلل بجمع المعلومات عن أساليب التسلل التي يمكن أن تُستخدم بدورها لتعزيز قدرة النظام على منع التسلل.

يعتمد اكتشاف التسلل على افتراض أن سلوك المتسلل يختلف عن سلوك المُستخدِم الشرعي من عدة وجوه يمكن تحديدها بشكلٍ كمّي. وبطبيعة الحال، لا يمكننا أن نتوقع أن يكون هناك حد فاصل واضح يميز بالضبط وبشكلٍ قطعي بين هجوم متسلل والاستخدام الطبيعي للموارد من قِبَل مستخدم شرعي. أي أنه يتعين علينا أن نتوقع بعض التماثل.

يوضّح الشكل 9-1، بأسلوبٍ مجرد للغاية، طبيعة المهمة التي تواجه مصممو نظم االكتشاف التسلل. فرغم أن البروفيل (الشكل العام) للسلوك النمطي للمتسلل يختلف عنه للمستخدم الشرعي، فهناك قدرٌ من الشبه بين هذين السلوكين. ومن ثم ، فإن تعريفاً فضفاضاً لسلوك المتسلل سيكون قادراً على كشف عددٍ أكبر من المتسللين، ولكنه في المقابل سيؤدي إلى عدد من الإنذارات الكاذبة أو ما يسمى

بالأخطاء الإيجابية الكاذبة (false positives) حيث سيُصنَّف عددٌ أكبر من المستخدمين الشرعيين على أنهم متسللون. وعلى الجانب الآخر، فإن محاولة الحد من الإنذارات الكاذبة عن طريق تضييق تعريف سلوك المتسلل ستؤدي إلى زيادة عدد المتسللين الذين يفشل النظام في التعرف عليهم كمتسللين ويصنفهم كمستخدمين شرعيين أو ما يسمى بالأخطاء السلبية الكاذبة (false negatives) بطريق الخطأ. ومن ثم تتطلب ممارسة اكتشاف التسلل قدراً من الموازنة والحنكة.



الشكل 9-1 بروفيلات لسلوك المتسللين وسلوك المستخدمين الشرعيين.

في دراسة لأندرسون [ANDE80] افترض فيها أنه يمكن، وبقدر معقول من الثقة، التمييز بين المتنكّر (masquerader) والمُستخدِم الشرعي. فبملاحظة التاريخ الماضي يمكن استنباط أنماط سلوك المستخدمين الشرعيين ، ومن ثم اكتشاف أي انحراف كبير عن تلك الأنماط. وقد ألمح أندرسون إلى أن مهمة كشف مسيء التصرف (misfeasor) (المُستخدِم الشرعي الذي يتصرف بطريقة غير مشروعة) تُعدُّ أكثر صعوبة، حيث إن الفرق بين السلوك الطبيعي والسلوك غير الطبيعي في هذه الحالة قد يكون ضئيلاً. وقد استنتج أندرسون أن مثل تلك الانتهاكات لن يتسنّى اكتشافها من خلال البحث عن السلوك الشاذ فقط. ومع ذلك، فقد يتسنّى اكتشاف سلوك مسيء التصرف عن طريق تعريف ذكي لمجموعة الشروط التي توحي بالاستخدام غير المشروع. وأخيرا فقد خَلُصت الدراسة إلى أن اكتشاف المُستخدِم السري يتجاوز إمكانات أساليب البحث الآلي البحتة. ومن الجدير بالذكر أن هذه الملاحظات التي قُدِّمت في عام 1980 لا تزال صحيحة اليوم.

حدَّد [PORR92] الأساليب الآتية لاكتشاف التسلل:

1. اكتشاف الشذوذ الإحصائي (Statistical anomaly detection): يتضمن ذلك جمع البيانات المتعلقة بسلوك المستخدمين الشرعيين على مدى فترة من الزمن. ثم إجراء اختبارات إحصائية على السلوك الملاحَظ لتحديدإمكانية الجزم ، بدرجة عالية من الثقة، أن ذلك السلوك ليس بسلوك مستخدم شرعي. هناك نوعان من هذا الأسلوب:

   a. الاكتشاف بالعتبة (Threshold detection): يتضمن ذلك تحديد عتبات، لاتعتمد على المستخدم، لمعدلات تكرار وقوع الأحداث المختلفة.

   b. الاكتشاف المبني على البروفيل (Profile based detection): يتم تحضير بروفيل لنشاط كل مستخدم واستخدامه لاكتشاف أي تغيّرات في سلوك حسابات الأفراد.

2. الاكتشاف بناءً على قواعد (Rule-based detection): يتضمن ذلك محاولة التوصل إلى مجموعة من القواعد التي يمكن استخدامها لتقرير ما إذا كان السلوك الملاحَظ صادراً عن متسلل. هناك نوعان من هذا الأسلوب:

a. اكتشاف الشذوذ: تُصمم القواعد بحيث تكشف عن الانحراف الحاصل عن أنماط الاستخدام السابقة.

b. التعرف على نوع الاختراق الناتج: يُستخدم نظام خبير (expert system) للبحث عن السلوكيات المشبوهة.

وباختصار، يمكن القول بأن الطرق الإحصائية تسعى لتحديد السلوك الطبيعي أو المتوقع، في حين تحاول الطرق القائمة على استخدام قواعد تعريف السلوك السليم.

من حيث أنواع المهاجمين المذكورة آنفا، يُلاحظ أن أساليب اكتشاف الشذوذ الإحصائي تُعدُّ فعّالة ضد فئة المتكّرين، حيث يُستبعد أن يقلد هؤلاء أنماط السلوك للحسابات التي استحوذوا عليها. ومن ناحية أخرى، فإن تلك الأساليب قد لا تكون قادرة على التعامل مع مسيئي التصرف. وتكون الأساليب المبنية على استخدام قواعد أكثر قدرةً على كشف هجمات هؤلاء، وذلك بالتعرف على أحداث وسلاسل زمنية معينة قد تؤدي - عند أخذها في السياق الملائم - للكشف عن اختراقات من ذلك النوع. وفي واقع الأمر، قد يجمع نظام اكتشاف التسلل مابين هذين الأسلوبين الأساسيين ليكون فعَّالا ضد تشكيلة واسعة من الهجمات.

9-2-1 سجلات المراجعة (Audit Records)

تُعدُ سجلات المراجعة أداة أساسية لاكتشاف التسلل، حيث يُحتفظ بسجلات عن النشاطات التي يقوم بها المستخدمون على النظام، و تُستخدم تلك السجلات كمُدخل إلى نظام اكتشاف التسلل. في هذا الصدد هناك خطتان أساسيتان:

- سجلات المراجعة المحلية: تتضمن كل أنظمة التشغيل تقريباً التي تدعم عدة مستخدمين برمجيات محاسبة تقوم بجمع معلومات عن نشاط المستخدم. من مزايا استخدام هذه المعلومات أننا لن نكون بحاجة إلى

إضافة برمجيات خاصة لجمع المعلومات. وتكمن المشكلة هنا في أن سجلات المراجعة المحلية قد لا تشمل المعلومات المطلوبة، أو قد لا توفرها بشكلٍ مناسب.

- سجلات المراجعة الخاصة باكتشاف التسلل: يمكن ترتيب آلية لجمع المعلومات وتوفير سجلات مراجعة تتضمّن فقط المعلومات المطلوب إدخالها إلى نظام اكتشاف التسلل. ومن مزايا هذه الطريقة أنه يمكن تصميم تلك الآلية بحيث تكون مستقلة عن المُوَرِّد ومن ثم تركيبها على عدة أنظمة مختلفة. ومن عيوب هذه الطريقة التعقيدات الناتجة عن وجود برنامجيْن للمحاسبة يعملان في الواقع جنباً إلى جنب على كل نظامٍ للحاسب. من الأمثلة الجيدة لأنظمة سجلات المراجعة الخاصة باكتشاف التسلل ذلك النظام الذي قامت بتطويره دوروثي دينج [DENN87]. يتضمن كل سجل مراجعة الحقول الآتية:

- الفاعل (Subject): وهو منشئ الإجراء. وعادةً يكون الفاعل أحد مستخدمي محطة طرفية (terminal user)، أو قد يكون عملية تنفّذ نيابةً عن مستخدم أو عن مجموعة من المستخدمين. تنجم كل الأنشطة عن أوامر يُصدرها الفاعلون. ويمكن تقسيم الفاعلين إلى فئات مختلفة من حيث وصولهم إلى النظام، وقد تتداخل تلك الفئات.

- الفعل (الإجراء) (Action): العملية التي يقوم بها الفاعل أو التي يتم إجراؤها على المفعول به، مثلاً كالدخول على النظام، أوالقراءة، أوالقيام بإدخال/إخراج بيانات أو التنفيذ.

- المفعول به (Object): مُتلقّي الفعل. ومن أمثلة ذلك الملفات والبرامج والرسائل والسجلات والمحطات الطرفية والطابعات، وهياكل البيانات من إنشاء المُستخدِم أو البرامج. عندما يكون متلقّي الفعل هو فاعل آخر، كما في حالة تلقّي رسالة بريد إلكتروني، فإن ذلك الفاعل يُعدُّ في هذه الحالة مفعولاً به. يمكن تصنيف المفعول بهم حسب النوع. قد تختلف درجة خشونة المفعول به (object granularity) حسب نوع المفعول به والبيئة. فمثلاً، قد تتم

مراجعة الإجراءات المتعلقة بقاعدة بيانات على مستوى قاعدة البيانات ككل أو على مستوى السجلات فيها.

- حالة الاستثناء (Exception-Condition): تشير إلى حالة الاستثناء، إن وجدت، التي يتم إعلانها عند العودة في نهاية الإجراء.
- استخدام الموارد: قائمة تضم مجموعة عناصر كمّية يمثل كل عنصر فيها الكمية المستخدمة من بعض موارد النظام (مثلاً، عدد السطور التي تم طباعتها أو عرضها على الشاشة، وعدد السجلات التي تم كتابتها أو قراءتها، ووقت المعالج الذي تم استهلاكه، ووحدات الإدخال/الإخراج التي تم استخدامها، والوقت المنقضي أثناء الجلسة).
- خاتم الوقت (Time-Stamp): خاتم بقيمة فريدة لوقت وتاريخ وقوع الفعل.

تتألف معظم عمليات المستخدمين من عدة أفعال أولية بسيطة. فعلى سبيل المثال، ينطوي أمر "نسخ ملف" على تنفيذ أمر المستخدم، والذي يتضمن تحصيل المصادقة على الوصول للملف والقيام بإعداد النسخة، والقراءة من ملف، بالإضافة إلى الكتابة إلى ملف آخر. لنأخذ في الاعتبار الأمر الآتي:

COPY GAME.EXE TO <Library>GAME.EXE

والصادر من Smith لنسخ الملف التنفيذي GAME من الدليل الحالي (current directory) إلى دليل آخر هو <Library>. قد يتم توليد سجلات المراجعة الآتية:

Smith	execute	<Library> COPY.EXE	0	CPU = 00002	11058721678

Smith	read	<Smith> GAME.EXE	0	RECORDS = 0	11058721679

Smith	execute	<Library> COPY.EXE	write-viol	RECORDS = 0	11058721680

في هذه الحالة، يتم إجهاض عملية نسخ الملف نظراً لأن Smith غير مصرّح له بالكتابة في الدليل <Library>.

هناك ثلاث مزايا لتحليل عمليات المُستخدِم إلى أفعال أولية:

1. لمّا كان المفعول به يمثل الكيانات المفترض حمايتها في النظام، فإن استخدام أفعالٍ أولية يسمح بمراجعة جميع التصرفات التي يمكن أن تؤثر على المفعول به. وهكذا، يمكن للنظام الكشف عن محاولات التلاعب بضوابط الوصول (access controls) (بملاحظة أي مظاهر غير طبيعية في أحوال الاستثناء (exception-conditions) التي يتم الإبلاغ عنها عند العودة من الإجراء، كما يمكنه الكشف عن النجاح في محاولة التخريب بملاحظة مظاهر غير طبيعية في مجموعة كيانات المفعول به التي أصبح بوسع الفاعل الوصول لها.
2. يؤدي استخدام سجلات المراجعة من نوع مفعول به واحد، وفعل واحد (Single-object, single-action) إلى تبسيط النموذج وطريقة التنفيذ.
3. إن الهيكل البسيط والمنتظم لسجلات المراجعة الخاصة باكتشاف التسلل قد يجعل من السهل نسبيا الحصول على تلك المعلومات، أو على الأقل جزءٍ منها، عن طريق مطابقة سجلات المراجعة المحلية الموجودة مباشرةً مع سجلات المراجعة الخاصة باكتشاف التسلل.

9-2-2 اكتشاف الشذوذ الإحصائي

كما ذكرنا آنفاً تنقسم أساليب اكتشاف الشذوذ الإحصائي إلى قسمين أساسيين: أنظمة الاكتشاف بالعتبة (threshold)، والأنظمة المبنية على البروفيل. يتضمن اكتشاف العتبة حساب عدد مرات وقوع حدث من نوع معين خلال فترة زمنية محددة. وإذا تجاوز هذا العدد الحد المقبول المُتوقع حدوثه في الوضع الطبيعي، فإننا نفترض وقوع تسلل.

يُعدُّ تحليل العتبات في حد ذاته طريقةً تقريبيةً وغير فعّالة حتى لاكتشاف هجمات على درجة متواضعة من التعقيد. وينبغي تحديد قيم كلٍّ من العتبة والفترة الزمنية؛ الأمر الذي يؤدي على الأرجح إلى كثير من الأخطاء الإيجابية الكاذبة أو كثير من الأخطاء السلبية الكاذبة نظراً للتباين الشديد عبر المستخدمين. ومع ذلك، قد يفيد استخدام عددٍ من الأنظمة البسيطة للاكتشاف بالعتبة عند توظيفها جنباً إلى جنب مع أساليب أخرى أكثر تطوراً.

يركز اكتشاف الشذوذ المبني على استخدام البروفيل على تحديد خواصٍ تميّز سلوك الأفراد أو مجموعات المستخدمين ذات الصلة واكتشاف الانحرافات الكبيرة عن تلك الخواص بعد ذلك. وقد يتألف البروفيل من مجموعة من المتغيرات، بحيث إن الانحراف في متغير واحد منها فقط قد لا يكون كافيا في حد ذاته لتوليد إنذار بالخطر.

تعتمد هذه الطريقة على تحليل سجلات المراجعة، حيث توفر تلك السجلات مدخلات لعملية اكتشاف التسلل من ناحيتين. أولاً، يجب على مصمم النظام تحديد عدد من المقاييس الكمّية التي يمكن استخدامها لقياس سلوك المستخدم. يمكن عن طريق تحليل سجلات المراجعة تلك على مدى فترة من الزمن تعيين الملامح التي تميّز النشاط المعتاد للمستخدم ومن ثم البروفيل الخاص به. أي أن سجلات المراجعة تُستخدم أولاً في تحديد السلوك النمطي. ثانياً، تُستخدم سجلات المراجعة الحالية بعد ذلك كمُدخلات لاكتشاف التسلل، حيث يقوم نموذج اكتشاف التسلل بتحليل سجلات المراجعة تلك لتحديد مدى الانحراف عن السلوك المعتاد. من أمثلة المقاييس المفيدة للاستخدام في اكتشاف التسلل بناءً على البروفيل ما يأتي:

- عدّاد (counter): وهو عدد صحيح غير سالب يمكن فقط أن يزداد ولكن لا ينقص إلى أن يتم تصفيره من قِبَل الإدارة. عادةً ما يتم حساب قيمة العداد لأنواع معينة من الأحداث على مدى فترة محددة من الزمن. ومن الأمثلة على ذلك عدد مرات محاولة الدخول على النظام من قِبَل مستخدم ما خلال

ساعة، وعدد مرات تنفيذ أمر بعينه خلال جلسة واحدة للمستخدم، وعدد مرات الإخفاق في إدخال كلمة السر خلال دقيقة.

- مقياس (gauge): وهو عدد صحيح غير سالب يمكن أن يزداد أو ينقص، ويُستخدم عادةً لقياس القيمة الحالية لكيانٍ ما. من الأمثلة على ذلك عدد الوصلات المنطقية المخصصة لأحد تطبيقات المستخدم، وعدد الرسائل الصادرة من إحدى عمليات المُستخدِم والتي تنتظر المعالجة.
- موقّت الفترة (Interval timer): وهو طول الفترة الزمنية بين حدثين مرتبطين. من الأمثلة على ذلك طول الفترة الزمنية بين محاولتين متتاليتين للدخول على حساب معيّن على النظام.
- استخدام الموارد (Resource utilization): وهي الكمية المُستَهلَكة من الموارد خلال فترة محددة. من الأمثلة على ذلك عدد الصفحات التي يطبعها مستخدم خلال جلسة، والزمن الكلي الذي يستغرقه تنفيذ برنامج.

باستخدام تلك المقاييس العامة، يمكن القيام بعدة اختبارات لتحديد ما إذا كان النشاط الحالي يقع في نطاق الحدود المقبولة. يسرد [DENN87] الطرق الآتية التي يمكن اتخاذها في هذا الصدد:

- المتوسط والانحراف المعياري.
- النموذج متعدد المتغيرات.
- عملية ماركوف.
- السلسلة الزمنية.
- النموذج التشغيلي.

أبسط اختبار إحصائي يمكن استخدامه هو قياس المتوسط والانحراف المعياري لمتغيرٍ ما على مدى فترة زمنية معينة. يعطي ذلك انطباعاً عن السلوك المتوسط ومدى التغير فيه مع الزمن. يصلح أسلوب المتوسط والانحراف المعياري

للاستخدام مع تشكيلة كبيرة من العدادات والموقّتات ومقاييس استخدام الموارد. غير أن هذه التدابير عادةً ما تكون غير كافية بمفردها لأغراض اكتشاف التسلل.

يعتمد النموذج متعدد المتغيرات على الارتباط بين اثنين أو أكثر من المتغيرات. يمكن توصيف سلوك المتسلل بثقة أكبر عن طريق النظر في علاقات الارتباط تلك (على سبيل المثال، الارتباط بين الوقت المستغل من المعالج واستخدام الموارد، أو معدل الدخول على النظام والوقت المنقضي في الجلسة).

يُستخدم نموذج عملية ماركوف في تحديد قيم احتمالات الانتقال بين مختلف حالات النظام. فمثلاً، يمكن استخدام هذا النموذج لدراسة الانتقالات بين أوامر بعينها.

يركز نموذج السلسلة زمنية على الفترات الزمنية، آخذاً في الاعتبار تسلسل الأحداث التي يمكن أن تقع بشكلٍ أسرع من المعتاد أو أبطأ من المعتاد. يمكن استخدام تشكيلة من الاختبارات الإحصائية لتوصيف التوقيت غير الطبيعي.

وأخيراً، يعتمد النموذج التشغيلي على تقدير ما يُعدّ غير طبيعي، وليس على القيام بالتحليل الآلي لسجلات المراجعة السابقة. وعادةً ما يتم وضع حدود ثابتة ويتم الاشتباه في حدوث تسلل عند ملاحظة سلوك خارج تلك الحدود. وتعمل هذه الطريقة بشكلٍ جيد عندما يكون من الممكن استنتاج سلوك المتسلل من خلال أنواع محددة من النشاط. على سبيل المثال، فإن عدداً كبيراً من محاولات الدخول خلال فترة قصيرة قد ينُم عن محاولة تسلل.

وكمثال على استخدام تلك المقاييس والنماذج المختلفة، يبيّن الجدول 9-1 التدابير المختلفة التي تم اختبارها أو أخذها في الاعتبار ضمن نظام اكتشاف التسلل (IDES) في معهد أبحاث ستانفورد (Stanford Research Institute (SRI)) [DENN87, JAVI91, LUNT88].

الجدول 9-1 المقاييس التي يمكن استخدامها لاكتشاف التسلل.

المقياس	النموذج	أنواع التسلل التي يمكن اكتشافها
أنشطة الدخول على النظام والجلسة		
معدل الدخول على النظام باليوم والوقت	المتوسط والانحراف المعياري	يُحتمل أن يحاول المتسللون الدخول على النظام خارج ساعات الدوام الرسمي
معدل الدخول على النظام في مواضع مختلفة	المتوسط والانحراف المعياري	يُحتمل أن يحاول المتسللون الدخول على النظام من موضع يندر أن يستعمله مستخدم بعينه أولا يستعمله على الإطلاق
الوقت المنقضي منذ آخر دخول على النظام	تشغيلي	اختراق على حسابٍ "ميِّت" (غير فعّال)
الوقت المنقضي أثناء كل جلسة	المتوسط والانحراف المعياري	قد ينم الانحراف الملحوظ عن وجود متنكر
كمية البيانات المُخرَجة إلى موضع	المتوسط والانحراف المعياري	قد ينم إرسال كمية كبيرة من البيانات إلى مواضع بعيدة عن تسريب بيانات حساسة
استخدام الموارد أثناء الجلسة	المتوسط والانحراف المعياري	قد تنم مستويات غير طبيعية من وقت المعالج أو عمليات الإدخال/الإخراج عن وجود متسلل
فشل إدخال كلمة السر أثناء محاولة الدخول على النظام	تشغيلي	محاولة للدخول على النظام عن طريق تخمين كلمة سر
فشل محاولة الدخول على النظام من محطات طرفية معينة	تشغيلي	محاولة اختراق
نشاطات تنفيذ أوامر أو برامج		
معدل التنفيذ	المتوسط والانحراف المعياري	قد يؤدي إلى كشف متسللين يُحتمل أن يحرصوا على استخدام أوامر مختلفة، أو كشف نجاح محاولة اختراق بواسطة مستخدم مشروع تمكّن من التوصل لصلاحية استخدام أوامر ذات امتيازات عالية.
استخدام موارد البرنامج	المتوسط والانحراف المعياري	قد تنم قيمة غير طبيعية عن حقن فيروس أو حصان طروادة في النظام، بما يصحب ذلك من آثار جانبية تتمثل في زيادة معدلات الإدخال/الإخراج واستخدام المعالج.
حجب تنفيذ أوامر	تشغيلي	قد يؤدي لكشف محاولة اختراق بواسطة مستخدم يسعى لتحصيل امتيازات أعلى.

تابع الجدول 9-1 المقاييس التي يمكن استخدم لاكتشاف التسلل.

المقياس	النموذج	أنواع التسلل التي يمكن اكتشافها
نشاطات الوصول إلى ملفات		
معدل عمليات القراءة والكتابة والإنشاء والحذف	المتوسط والانحراف المعياري	قد تنم المظاهر غير الطبيعية في عمليات الوصول للقراءة والكتابة بواسطة المستخدمين عن أعمال تنكّر أو تصفح.
السجلات التي يتم قراءتها وكتابتها	المتوسط والانحراف المعياري	قد تنم المظاهر غير الطبيعية عن محاولة للحصول على بيانات حساسة عن طريق الاستدلال والتجميع.
عدد مرات الفشل في عمليات القراءة والكتابة والإنشاء والحذف	تشغيلي	قد تكشف مستخدمين يصرّون على محاولة الوصول إلى ملفات غير مرخّصة لهم.

تكمن الميزة الأساسية لاستخدام البروفيلات الإحصائية في أننا لا نحتاج لمعرفة مسبقة بالثغرات الأمنية، حيث يقوم البرنامج بنفسه بتعلم ما يُعد سلوكاً "طبيعياً" ثم يبحث عن الانحرافات عن هذا السلوك. ولا تعتمد هذه الطريقة على خواص ونقاط ضعف متعلقة بنظام حاسب بعينه، ومن ثَمَّ، يسهل نقلها واستخدامها على أنواع مختلفة من أنظمة الحاسب.

9-2-3 اكتشاف التسلل بناءً على قواعد (Rule-based Intrusion Detection)

تقوم الأساليب المبنية على قواعد باكتشاف التسلل عن طريق مراقبة الأحداث التي تقع في النظام وتطبيق مجموعة من القواعد للتوصل إلى قرار بشأن ما إذا كان نمط معين من النشاط مشبوهاً. بشكلٍ عامٍ جداً، يمكننا تصنيف كل الطرق المستخدمة هنا إلى طرق تكتشف الشذوذ أو طرق تتعرف على الاختراق، رغم وجود بعض التطابق بين هذين الصنفين.

يشبه اكتشاف الشذوذ بناءً على قواعد اكتشاف الشذوذ الإحصائي من حيث الطريقة التي يستخدمها ونقاط القوة التي يمتاز بها. ففي الأسلوب الذي يعتمد

على قواعد، يتم تحليل سجلات المراجعة التاريخية لتحديد أنماط الاستخدام والقيام آلياً بتوليد قواعد تصف تلك الأنماط. وقد تمثل تلك القواعد أنماط السلوك الماضية للمستخدمين والبرامج والامتيازات والشرائح الزمنية والمحطات الطرفية، وما إلى ذلك. بعد ذلك يتم ملاحظة السلوك الحالي، وتُطابَق كل معاملة إزاء تلك المجموعة من القواعد لتحديد ما إذا كانت تلك المعاملة تتفق مع أي نمط سلوكي تمت ملاحظته تاريخياً.

وكما هو الحال مع اكتشاف الشذوذ الإحصائي، فإن اكتشاف الشذوذ بناءً على قواعد لا يتطلب معرفة نقاط الضعف الأمنية في النظام. فالخطة تقوم على مراقبة السلوك في الماضي، وتفترض في الواقع أن المستقبل سيكون كالماضي. ولكي تكون هذه الطريقة فعّالة، ينبغي أن تضم قاعدة البيانات عدداً كبيراً من القواعد. فعلى سبيل المثال، يستخدم النظام الذي وصفه [VACC89] قاعدة بيانات تضم مابين 10^4 و 10^6 قاعدة.

تتبع نظم التعرف على الاختراق بناءً على قواعد أسلوباً مختلفا للغاية لكشف التسلل يعتمد على تقنية الأنظمة الخبيرة. وتتلخص السمة الرئيسة لتلك النظم في استخدام قواعد للتعرف على اختراقات معروفة أو اختراقات تستغل نقاط ضعف معروفة. ويمكن أيضاً وضع قواعد للتعرف على سلوك مشبوه، حتى لو كان هذا السلوك يقع ضمن حدود أنماط الاستخدام المعتادة. وعادةً ما تكون القواعد المستخدمة في تلك النظم خاصة بنظام حاسب ونظام تشغيل بعينه. كما توضع تلك القواعد من قِبَل "خبراء" آدميين وليس عن طريق التحليل الآلي لسجلات المراجعة. تعتمد الطريقة المعتادة على إجراء مقابلات مع مديري الأنظمة والمحللين الأمنيين لجمع مجموعة من سيناريوهات الاختراق المعروفة والأحداث الرئيسة التي تهدد أمن النظام المستهدف[1]. وهكذا، فإن قوة هذه الطريقة تعتمد على مهارات المشاركين في وضع تلك القواعد.

[1] بل قد تتسع دائرة تلك المقابلات لتشمل القراصنة التائبين أو الذين لم يتوبوا بعد المستعدين لتقاسم خبراتهم نظير أجر [FREE93].

يوجد مثال بسيط لأنواع القواعد التي يمكن استخدامها في هذا الصدد في NIDX، وهو نظام تم تطويره سابقاً ويستخدم قواعد استدلال لامنهجية يمكن استعمالها لتعيين درجة الشك في الأنشطة المختلفة للمستخدمين [BAUE88]. من أمثلة قواعد الاستدلال تلك ما يأتي:

1. لا ينبغي أن يقوم المستخدمون بقراءة الملفات الموجودة في الأدلة الشخصية للمستخدمين الآخرين.
2. لا ينبغي أن يقوم المستخدمون بالكتابة في ملفات المستخدمين الآخرين.
3. المستخدمون الذين يدخلون على النظام بعد ساعات الدوام غالباً ما يصلون إلى نفس الملفات التي استخدموها في السابق.
4. المستخدمون عامةً لا يفتحون أجهزة أقراص التخزين مباشرة ولكنهم يعتمدون في ذلك على برامج خدمة (utilities) ذات مستوى أعلى يوفرها نظام التشغيل.
5. لا ينبغي للمستخدمين الدخول أكثر من مرة في نفس الوقت على نفس النظام.
6. المستخدمون لايسعون للحصول على نسخ من برامج النظام.

يمثل النظام المُستخدَم في IDES للتعرف على الاختراق الاستراتيجية المتبعة عادةً في هذا المجال، حيث يتم فحص سجلات المراجعة بمجرد إنشائها، ومطابقتها بالقواعد الموجودة في قاعدة البيانات. عند وُجود تطابق، فإن معدل الاشتباه في ذلك المُستخدِم يزداد. وفي حال حدوث تطابق مع عدد كافٍ من تلك القواعد، فإن معدل الاشتباه المتزايد قد يجتاز العتبة المحددة مما يؤدي إلى الإبلاغ عن اكتشاف شذوذ عن السلوك الطبيعي.

الجدول 9-2 إجراءات USTAT في مقابل أنواع الأحداث في نظام SunOS.

إجراء USTAT	نوع حدث SunOS
اقرأ (Read)	open_r, open_rc, open_rtc, open_rwc, open_rwtc, open_rt, open_rw, open_rwt
اكتب (Write)	truncate, ftruncate, creat, open_rtc, open_rwc, open_rwtc, open_rt, open_rw, open_rwt, open_w, open_wt, open_wc, open_wct
أنشئ (Create)	mkdir, creat, open_rc, open_rtc, open_rwc, open_rwtc, open_wc, open_wtc, mknod
احذف (Delete)	rmdir, unlink
نفّذ (Execute)	exec, execve
اخرج (Exit)	exit
عدّل المالك (Modify_Owner)	chown, fchown
عدّل التصريح (Modify_Perm)	chmod, fchmod
غيّر الاسم (Rename)	Rename
وصلة صلبة (Hardlink)	link

تعتمد طريقة عمل نظام IDES على فحص سجلات المراجعة. وتُعدُّ قلة المرونة من إحدى نقاط الضعف في هذه الطريقة. فلكل سيناريو اختراق، هناك عدد من التسلسلات البديلة لسجلات المراجعة، تختلف فيما بينها اختلافاتٍ بسيطة أو بشكلٍ غير ملحوظ. وقد يكون من الصعب تضمين تلك الاختلافات في قواعد صريحة. تتلخص إحدى الطرق الأخرى في وضع نموذج على مستوى أعلى بحيث يكون مستقلاً عن أية سجلات مراجعة محددة. ومن الأمثلة على ذلك نموذج انتقال الحالة المعروف بـ USTAT [ILGU93]. يتعامل USTAT مع الإجراءات العامة وليس مع الأفعال التفصيلية المحددة التي تسجلها آلية المراجعة على نظام التشغيل يونيكس. وقد تم تنفيذ USTAT على نظام SunOS الذي يوفر سجلات مراجعة

تشمل 239 حدثاً. تم اختيار 28 حدثٍ فقط من بين تلك الأحداث ليقوم معالج قَبْلي بمقابلتها بعشرة إجرءات عامة، كما بالجدول 9-2. باستخدام تلك الإجراءات العشر فقط مع المتغيرات المرتبطة بكل إجراء أمكن تطوير مخطط لانتقال حالة (state transition diagram) يصف النشاط المشبوه. نظراً لأن عدد الأحداث المختلفة التي يمكن تسجيلها في سجلات المراجعة تقابل عدداً أقل من الإجراءات المختلفة ، فإن عملية إنشاء القواعد تكون أبسط. وعلاوةً على ذلك، يمكن تعديل نموذج مخطط انتقال الحالة لاستيعاب سلوكيات التسلل الجديدة التي يتعلمها النظام.

9-2-4 مغالطة المعدل الأساسي

لكي يكون لنظام اكتشاف التسلل فائدة عملية، ينبغي أن يكون بمقدوره الكشف عن نسبة كبيرة من المتسللين مع الحفاظ على معدل الإنذارات الكاذبة عند مستوى مقبول. إذا كان النظام يكتشف فقط نسبة متواضعة من محاولات التسلل الفعلية فإنه بهذا يعطي إحساساً زائفاً بالأمان. وفي المقابل، إذا كان هذا النظام في كثير من الأحيان يصدر إنذارا بينما لا توجد محاولة تسلل في الواقع (أي إنذار كاذب) فإما أن يبدأ مديرو نظام الحاسب في تجاهل الإنذارات التي يصدرها النظام أو أن وقتاً طويلاً سيضيع في تحليل إنذارات كاذبة.

للأسف، فإنه بسبب طبيعة الاحتمالات التي ينطوي عليها النظام، فإنه من الصعب جداً تلبية المعايير المطلوبة لمعدّلات عالية للاكتشاف مع معدّلات منخفضة للإنذارات الكاذبة في آن واحد. وعموماً، إذا كان العدد الفعلي للاختراقات قليلاً مقارنةً بعدد الاستخدامات المشروعة للنظام، فإن معدل الانذارات الكاذبة سيكون عالياً ما لم يكن الاختبار المُستخدَم للوصول إلى قرار على درجة عالية من التمييز. وتشير دراسة عن أنظمة اكتشاف التسلل الحالية وردت في [AXEL00] إلى أن النظم الحالية لم تتغلب على مشكلة مغالطة المعدل الأساسي. انظر الملحق A-9 للاطلاع على استعراض موجز للخلفية الرياضية لتلك المشكلة.

9-2-5 الأنظمة الموزّعة لاكتشاف التسلل

حتى وقت قريب، انصبّ العمل في مجال اكتشاف التسلل على تطوير أنظمة قائمة بذاتها تعمل على نظام حاسب واحد. غير أن المؤسسات تحتاج عادةً إلى الدفاع عن مجموعة موزّعة من الأنظمة المضيفة ترتبط فيما بينها بشبكة إنترنت أو شبكة محلية. رغم أنه من الممكن توفير ذلك الدفاع عن طريق أنظمة اكتشاف تسلل قائمة بذاتها على كل مضيف، فإن الدفاع سيكون أكثر فعالية إذا تم التنسيق والتعاون بين تلك الأنظمة عبر الشبكة.

يوضّح بوراس المسائل الرئيسة الآتية فيما يتعلق بتصميم نظام موزّع لاكتشاف التسلل [PORR92]:

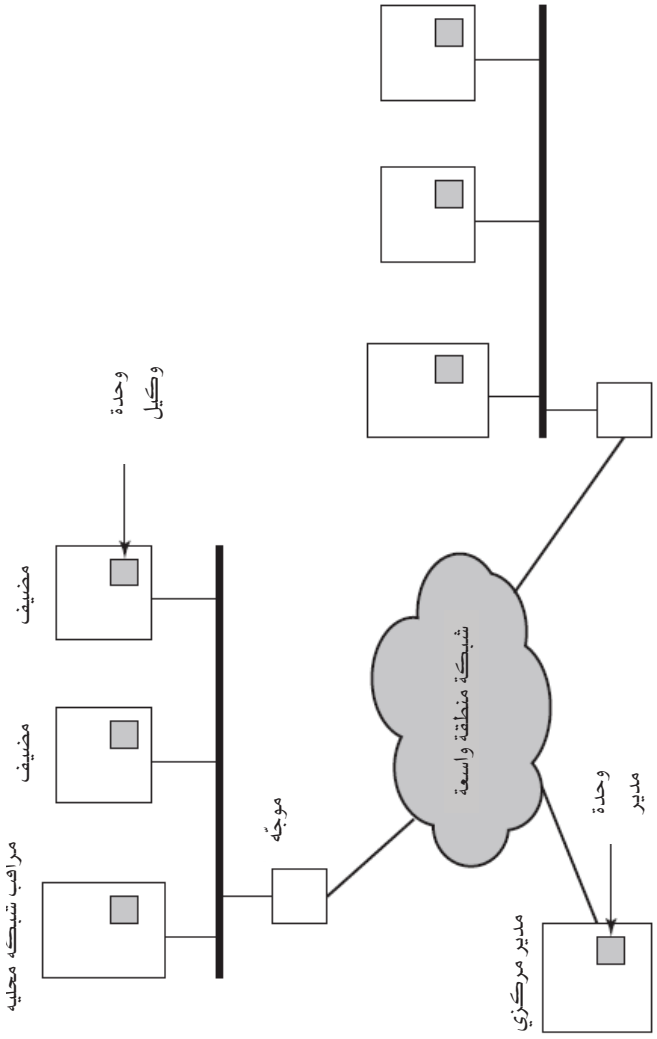
- قد يحتاج النظام الموزّع لاكتشاف التسلل إلى التعامل مع صيغ مختلفة من سجلات المراجعة. وفي بيئة غير متجانسة، سوف تستخدم نظم الحاسب المختلفة أنظمة محلية مختلفة لجمع معلومات المراجعة، وإذا كانت تستخدم أنظمة لاكتشاف التسلل، فقد تستخدم صيغاً مختلفة لسجلات المراجعة المتعلقة بالأمن.
- ستُستخدم عقدة أو أكثر في الشبكة كنقاط لجمع البيانات من الأنظمة الموجودة على الشبكة وتحليلها. وعليه سيتعين إرسال بيانات مراجعة خام أو موجزة عبر الشبكة. ومن ثَمَّ، سيتطلب الأمر ضمان سلامة وسرية تلك البيانات. سلامة البيانات مطلوبة لمنع المتسلل من إخفاء نشاطه عن طريق تغيير بيانات المراجعة المُرسَلة. أما السرية فمطلوبة لأن بيانات المراجعة المرسلة قد تكون ثمينة.
- يمكن استخدام بنية معمارية مركزية أو لامركزية. في حالة البنية المركزية، تكون هناك نقطة مركزية واحدة لجمع كل بيانات المراجعة وتحليلها. رغم أن هذا يُسهِّل مهمة التنسيق واستخلاص العلاقات المشتركة بين التقارير الواردة، فإنه قد يمثل عنق زجاجة كما يمكن أن يؤدي إلى حدوث اختناق أو إلى احتمال تعطّل للنظام بتعطل نقطة واحدة فيه (single

(point of failure). في البنية اللامركزية يوجد أكثر من مركز لتحليل البيانات، ولكن يتعين على تلك المراكز تنسيق أنشطتها وتبادل المعلومات فيما بينها.

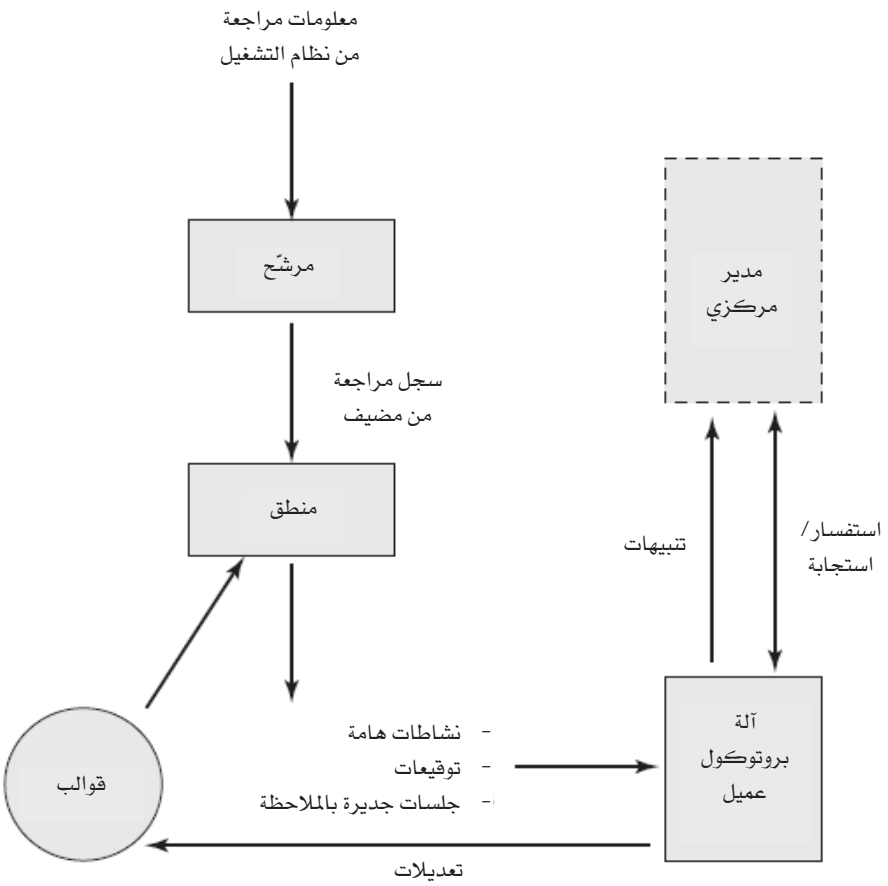
من الأمثلة الجيدة على الأنظمة الموزّعة لاكتشاف التسلل ذلك النظام الذي تم تطويره في جامعة كاليفورنيا في ديفيز [HEBE92, SNAP91]. ويوضّح الشكل 9-2 البنية المعمارية الكلية لذلك النظام، والتي تتألف من ثلاثة عناصر رئيسة هي:

- وحدة وكيل المضيف: وحدة لجمع بيانات المراجعة يجري تشغيلها كعملية بخلفية النظام الذي تجري مراقَبته. تقوم الوحدة بجمع البيانات عن الأحداث المتعلقة بالأمن بالمضيف وإرسالها إلى المدير المركزي.
- وحدة مراقبة الشبكة المحلية: تعمل بنفس طريقة وحدة وكيل المضيف إلا أنها تقوم بتحليل حركة مرور البيانات على الشبكة المحلية وإرسال النتائج إلى المدير المركزي.
- وحدة المدير المركزي: تستقبل التقارير الواردة من وحدات وكلاء مراقبة الشبكة المحلية ووكلاء وعمليات المضيفين وتقوم بالربط بينها بهدف اكتشاف التسلل.

صُمم النظام ليكون مستقلاً عن أيٍّ من أنظمة التشغيل أو أنظمة المراجعة. ويوضّح الشكل 9-3 [SNAP91] النهج العام المتبع. ويقوم الوكيل بالتقاط كل سجل مراجعة ينتجه نظام جمع بيانات المراجعة المحلي ويُستخدم مرشّح للاحتفاظ فقط بالسجلات التي لها علاقة بالأمن. يتم بعد ذلك وضع تلك السجلات في صيغة قياسية موحّدة تُعرف بسجل المراجعة للمضيف ((host audit record (HAR)).



الشكل 9-2 البنية المعمارية لنظام موزّع لاكتشاف التسلل.



الشكل 9-3 البنية المعمارية لوكيل.

تقوم وحدة منطقية مبنية على قالب (template-driven) بتحليل السجلات بحثاً عن أي نشاط مشبوه. في أدنى مستوى، يقوم الوكيل بعملية مسح للسجلات بحثاً عن أحداث بارزة لها أهميتها بصرف النظر عن أي أحداث سابقة. ومن أمثلة ذلك: الفشل في الوصول إلى ملفات، والوصول إلى ملفات النظام، وتغيير ضوابط التحكم في الوصول إلى ملف. وفي أعلى مستوى، يبحث الوكيل عن تسلسلات أحداث،

كأنماط الهجوم المعروفة (التوقيعات). وأخيراً، يبحث الوكيل عن سلوكيات شاذة لمستخدم بعينه على أساس البروفيل التاريخي لذلك المستخدم، كعدد البرامج التي ينفذها، وعدد الملفات التي يصل إليها، وما إلى ذلك.

عند اكتشاف أي نشاط مشبوه، يتم إرسال تنبيه إلى المدير المركزي. يستخدم المدير المركزي نظاماً خبيراً (expert system) يمكّنه من استخلاص النتائج من البيانات الواردة. كما يمكن للمدير أيضاً أن يطلب من مختلف الوكلاء إرسال نسخ من سجلات المراجعة HAR للربط ما بينها وبين السجلات الواردة من وكلاء آخرين.

يقوم وكيل الشبكة المحلية أيضاً بتزويد المدير المركزي بالمعلومات. كما يقوم الوكيل بمراجعة التوصيلات بين الأنظمة المضيفة والخدمات المستخدمة وحجم حركة المرور. ويقوم أيضاً بالبحث عن الأحداث المهمة، كالتغيرات المفاجئة في حِمل الشبكة، واستخدام خدمات متعلقة بالأمن، والأنشطة على الشبكة مثل عمليات الدخول عن بُعد (rlogin).

تُعدُّ البنى المعمارية المبينة في الشكلين 9-2 و9-3 عامةً جداً ومرنة جداً. وهي تمثل الأساس لطريقة مستقلة عن نوع أجهزة الحاسب، ويمكن التوسع فيها ابتداءً من أنظمة قائمة بذاتها لاكتشاف التسلل إلى نظام قادر على الربط مابين الأنشطة على عدد من المواقع والشبكات للكشف عن الأنشطة المشبوهة والتي بدون ذلك كانت ستبقى مخفية.

9-2-6 المصايد الأمنية (Honeypots)

المصايد الأمنية هي وسيلة حديثة نسبياً في مجال تقنيات اكشاف التسلل. تعمل أنظمة المصايد الأمنية كفخاخ بهدف جذب اهتمام المهاجم المحتمل بعيداً عن النظم الأساسية ذات الأهمية المحورية. يتم تصميم تلك المصايد الأمنية لتقوم بما يأتي:

- صرف اهتمام المهاجم بحيث لا يصل إلى النظم المحورية.

- جمع المعلومات عن نشاط المهاجم.
- تشجيع المهاجم على البقاء على النظام لفترة كافية ليتسنى للإداريين التعامل معه.

تمتلئ تلك النظم ببيانات ملفقة صمّمت لتبدو وكأنها معلومات قيّمة ولكنها غير متاحة فلا يمكن الوصول إليها من قِبَل المستخدمين الشرعيين للنظام. وعليه فإن أي وصول إلى المصيدة الأمنية يكون مثار شبهة. يُزوّد النظام بأجهزة مراقبة حساسة ومسجّلات للأحداث للكشف عن أي زيارة للمصيدة وجمع المعلومات عن المهاجم. ونظراً لأن النظام مصمم بحيث يُظهر أي هجوم على المصيدة الأمنية كهجوم ناجح، فإنه سيكون لدى الإداريين الوقت الكافي للتعبئة وتسجيل الهجوم ومتابعة مساره دون الكشف عن النظم الفعلية/الفعّالة.

تضمّنت الجهود الأولية استخدام مصيدة أمنية تتألف من نظام حاسب واحد له عنوان IP مصمم لجذب القراصنة. وقد ركّزت البحوث التي أُجريت مؤخراً في هذا الصدد على بناء شبكات بأكملها من المصايد الأمنية لتبدو كما لوكانت شبكة مؤسسة، وقد تتضمن بيانات وحركة مرور حقيقية أو محاكاة. بمجرد أن يصبح القراصنة داخل الشبكة، يمكن للإداريين مراقبة سلوكهم بالتفصيل وحشد الدفاعات الناجعة ضدهم.

9-2-7 صيغة تبادل البيانات لاكتشاف التسلل

لتيسير تطوير نظم موزّعة لاكتشاف التسلل يمكنها العمل عبر تشكيلة كبيرة من البيئات، ينبغي وضع المعايير اللازمة لدعم العمل المشترك عبر الأنظمة المختلفة. تركّز مجموعة العمل الخاصة باكتشاف التسلل ضمن فريق مهام هندسة الإنترنت (IETF) على تطوير تلك المعايير. والغرض من تشكيل مجموعة العمل تلك هو تحديد صيغ وإجراءات تبادل البيانات للمساعدة في تقاسم المعلومات التي تهم أنظمة الكشف والاستجابة لمحاولات التسلل وكذلك النظم الإدارية التي قد تحتاج للتفاعل مع تلك الأنظمة. تتضمن النتائج المطلوبة من مجموعة العمل ما يأتي:

1. إعداد وثيقة متطلبات تصف الاحتياجات الوظيفية عالية المستوى للاتصال فيما بين أنظمة اكتشاف التسلل، وكذلك متطلبات الاتصال بين أنظمة اكتشاف التسلل ونظم الإدارة، بما في ذلك الأسباب المنطقية لتلك المتطلبات. ولتوضيح تلك المتطلبات سيتم استخدام سيناريوهات متعددة.
2. إعداد مواصفات للغة موحّدة لأغراض اكتشاف التسلل، والتي تصف صيغ البيانات اللازمة لتلبية الاحتياجات المطلوبة.
3. وثيقة إطار تحدد أفضل البروتوكولات الموجودة حالياً لاستخدامها في الاتصال بين أنظمة اكتشاف التسلل وتصف علاقة صيغ البيانات المقترحة بتلك البروتوكولات.

حتى وقت كتابة هذا الجزء، فإن كل تلك الوثائق وصلت إلى مرحلة مسودة وثيقة إنترنت.

9-3 إدارة كلمات السر

9-3-1 حماية كلمة السر

يُعدُّ نظام كلمة السر خط الدفاع الأول ضد المتسللين. حيث تتطلب جميع الأنظمة لعدة مستخدمين تقريباً أن يقوم المُستخدِم بإدخال اسم (أو هوية) المُستخدِم (ID)، وأيضاً بإدخال كلمة سر. والغرض من كلمة السر هو توثيق هوية الشخص للدخول على النظام. وبدورها، توفر هوية المُستخدِم الأمن من الجوانب الآتية:

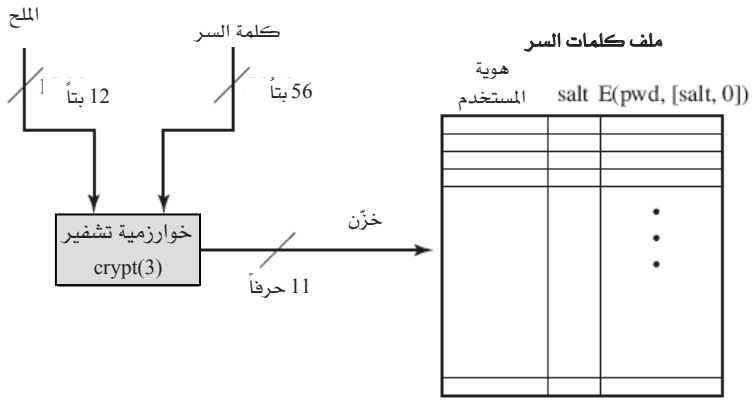
- تُبيّن الهوية ما إذا كان المُستخدِم مرخّصاً له بالدخول على النظام. في بعض النظم، يُسمح بالدخول فقط للأشخاص الذين لديهم فعلاً هوية مخزّنة على النظام.
- تحدّد الهوية الامتيازات الممنوحة للمستخدم. فقد يتمتع قلة من المستخدمين بمرتبة إشرافية (مستخدم متميز) تمكّنهم من قراءة الملفات وأداء المهام التي تتمتع بحماية خاصة من قِبَل نظام التشغيل. وقد توفر بعض الأنظمة

حسابات للضيوف أو حسابات مجهولة الهوية يتمتع مستخدموها بامتيازات محدودة مقارنةً بغيرهم.

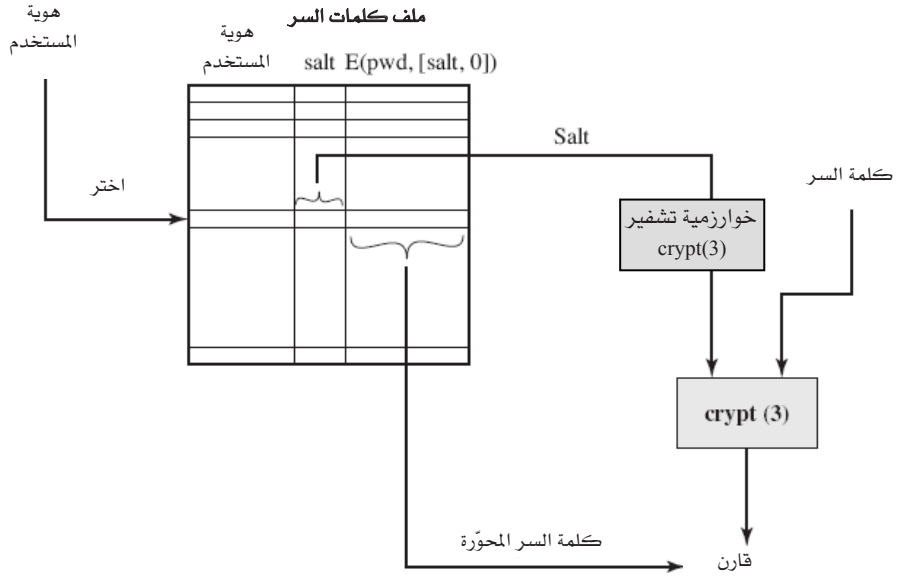
- تُستخدم الهوية فيما يُعرف بالتحكم التقديري في الدخول على النظام (discretionary access control). فعلى سبيل المثال، من خلال نشر قائمة بهويات المستخدمين الآخرين، يمكن لمستخدم أن يسمح لبعضهم بقراءة الملفات التي يملكها .

❖ نقاط الضعف في كلمات السر:

لكي نفهم طبيعة التهديد الذي تتعرض له النظم القائمة على كلمة السر، نأخذ بعين الاعتبار طريقة تستخدم بكثرة على نظام التشغيل يونيكس، حيث لاتخزّن كلمات السر واضحةً أبداً. بدلاً من ذلك، يُستخدم الإجراء التالي (انظر الشكل 9-4 (a)). يختار كل مستخدم كلمة سر تتألف من أحرف قابلة للطباعة بطول أقصى يبلغ ثمانية أحرف. يتم تحويل تلك الكلمة إلى قيمة من 56 بتاً (باستعمال كود آسكي بـ 7 بتات) لتُستخدم كمُفتاح ل لبرنامج التشفير المعروف باسم crypt(3) المبني على خوارزمية DES. يتم تعديل DES باستخدام قيمة تُعرف بالملح طولها 12 بتاً، وعادةً ماتعتمد تلك القيمة على الوقت الذي تم فيه تخصيص كلمة السر للمستخدم. يتم تشغيل خوارزمية DES بإدخال كتلة بيانات تتألف من 64 بتاً كلها أصفار، ويُستخدم ناتج هذه العملية كمُدخل لعملية التشفير التالية. تُكرّر هذه العملية لإتمام عدد كلي من التشفيرات يبلغ 25 تشفيرة، ويُعامل الناتج الذي يتألف من 64 بتاً على أساس أنه يمثل تسلسلاً من 11 حرفاً. بعد ذلك تُخزن كلمة السر المحوّرة (hashed password)، مع نسخة غير مُشفَّرة من قيمة الملح، في ملف كلمات السر مقابل الهوية المناظرة للمستخدم. تم إثبات أن هذه الطريقة آمنة ضد مجموعة متنوعة من هجمات تحليل الشفرة [WAGN00].



(a) تخزين كلمة سر جديدة



(b) التحقق من كلمة سر

الشكل 9-4 منظومة كلمة السر في نظام التشغيل يونيكس.

تخدم قيمة الملح ثلاثة أغراض:

- تمنع ظهور كلمات سر مكررة في ملف كلمات السر. فحتى لو اختار مستخدمان نفس كلمة السر، فإن كلمة السر ستخصص لهما في وقتين مختلفين. ومن ثم فإن كلمة السر "الموسّعة" والمخزنة في الملف ستكون مختلفة لكلٍ منهما.
- تؤدي عملياً إلى زيادة طول كلمة السر للمستخدم بحرفين دون أن يحتاج المُستخدِم إلى تذكّر حرفين إضافيين. وعليه، فإن ذلك يؤدي إلى زيادة عدد كلمات السر الممكنة 4096 مرة، مما يزيد من صعوبة تخمين كلمة السر.
- تمنع استخدام عتاد تنفيذ خوارزمية DES والذي يؤدي، لو استخدم، إلى تقليل صعوبة القيام بهجوم استقصائي لتخمين كلمة السر.

عندما يحاول المُستخدِم الدخول على نظام التشغيل يونيكس فإنه يقوم بإدخال هوية المُستخدِم وكلمة السر. يستخدم نظام التشغيل الهوية كمؤشر إلى كلمة السر في الملف ويسترجع القيمة غير المُشفَّرة للملح وكذلك كلمة السر المُشفَّرة. تُستخدم قيمة الملح وكلمة السر التي أدخلها المُستخدِم كمُدخل لروتين التشفير. إذا تطابقت نتيجة التشفير مع القيمة المخزنة في الملف اعتُبرت كلمة السر مقبولة.

صُمم روتين التشفير بحيث يثبّط من محاولات التخمين. فتنفيذ خوارزمية DES بالبرمجيات أبطأ منه باستخدام العتاد، كما أن تكرار عملية التشفير 25 مرة يضاعف الوقت اللازم للعملية 25 مرة. ومع ذلك فقد طرأ تغييران منذ التصميم الأصلي لتلك الخوارزمية. أولاً، أدت الإصدارات الأحدث من الخوارزمية إلى تنفيذها بشكلٍ أسرع. على سبيل المثال، تمكنت دودة الإنترنت التي نورد وصفاً لها في الفصل العاشر من القيام بتخمين بضع مئات من كلمات السر على الخط مباشرةً في فترة زمنية محدودة باستخدام خوارزمية تشفير أكثر كفاءةً بكثير من النسخة المعيارية المخزنة على أنظمة يونيكس التي كانت عرضةً للهجوم. وثانيا، مع التزايد المضطرد في سرعة أداء أجهزة الحاسب، فإن أي خوارزميات تنفذ بواسطة برمجيات يتم إنجازها أيضاً بشكلٍ أسرع من ذي قبل.

مما تقدم يتضح أن هناك تهديدين لمنظومة كلمة السر على نظام التشغيل يونيكس. أولاً، يمكن للمستخدم الدخول على جهاز عن طريق حساب ضيف أو بأي وسيلة أخرى، ثم يقوم بتشغيل برنامج لتخمين كلمة السر، والذي يُعرف باسم مكسّر كلمات السر (password cracker) على ذلك الجهاز. وسيكون بوسع المهاجم تجربة المئات وربما الآلاف من كلمات السر الممكنة باستخدام كمية ضئيلة من موارد النظام. بالإضافة إلى ذلك، إذا تمكن الخَصْم من الحصول على نسخة من ملف كلمات السر، فسيمكنه تشغيل برنامج للتكسير على رِسْلِه في جهاز آخر. وبهذه الطريقة سيتمكّن الخَصْم من تجربة عدة آلاف من كلمات السر الممكنة خلال فترة معقولة.

كمثال على ذلك، تم الإبلاغ عن مكسّر لكلمات السر على شبكة الإنترنت في أغسطس 1993 [MADS93]. وباستخدام جهاز حاسب بمعالجات من إنتاج شركة Thinking Machines Corporation على التوازي، أمكن تحقيق أداء بلغ 1560 تشفيرة في الثانية لكل وحدة متجه (vector unit). باستخدام أربع وحدات متجه على كل عقدة (وتلك ترتيبة قياسية) يتم إنجاز 800,000 تشفيرة في الثانية على جهاز يضم 128 عقدة (وهو جهاز متواضع الحجم) أو 6.4 مليون تشفيرة في الثانية على جهاز يضم 1024 عقدة.

حتى معدلات التخمين الهائلة تلك لا تجعل من الممكن لمهاجم استخدام الأسلوب الاستقصائي الغبي الذي يجرّب كل تباديل الأحرف الممكنة لاكتشاف كلمة سر. بدلاً من ذلك يعتمد مكسّرو كلمات السر على حقيقة أن بعض الناس يختارون كلمات سر يمكن تخمينها بسهولة.

بعض المستخدمين، عندما يُسمح لهم باختيار كلمة سر خاصة بهم، يختارون كلمة قصيرة للغاية. ويبيّن الجدول 9-3 نتائج دراسة أجريت في جامعة بوردو. راقبت الدراسة خيارات تغيير كلمة السر على 54 جهاز حاسب تمثل حوالي 7,000 حساب مستخدم. تألفت 3% من كلمات السر من 3 أحرف أو أقل. وبوسع مهاجم بدء هجومه بأن يقوم باختبار استقصائي لكل كلمات السر الممكنة بطول 3 أحرف أو

أقل. ويتمثل أحد الحلول البسيطة لتلك المشكلة في أن يرفض النظام اختيار كلمة سر تكون أقل من ستة أحرف مثلاً، أو حتى أن يشترط أن تتكون جميع كلمات السر من ثمانية أحرف بالضبط. ومعظم المستخدمين لن يتضجّر من مثل تلك القيود.

إن طول كلمة السر ليس سوى جزء واحد من المشكلة، فكثير من الناس عندما يُسمح لهم باختيار كلمة سر خاصة بهم، يختارون كلمة سر يمكن تخمينها، كاسم الشخص، أو اسم الشارع، أو كلمة عامة من كلمات القاموس، مما يجعل تكسير كلمة السر مهمة سهلة وبسيطة. فكل مايحتاجه مكسّر كلمات السر هو أن يقوم ببساطة باختبار ملف كلمات السر بحثاً عن قوائم من كلمات السر المحتملة. ولأن كثير من الناس يستخدمون كلمات سر يسهل تخمينها، فإن هذه الاستراتيجية سوف تنجح على جميع النظم.

الجدول 9-3 أطوال كلمة المرور التي تم ملاحظتها في دراسة [SPAF92a].

الطول	العدد	الكسر من المجموع الكلي
1	55	.004
2	87	.006
3	212	.02
4	449	.03
5	1260	.09
6	3035	.22
7	2917	.21
8	5772	.42
المجموع	**13787**	**1.0**

من التجارب التي أكدت فعالية التخمين ما ورد في [KLEI90]. جمع المؤلف، من مصادر مختلفة، ملفات لكلمات السر على نظام يونيكس تتضمن ما يقرب من 14,000 كلمة سر مُشفَّرة. ويبيّن الجدول 9-4 النتيجة، والتي يصفها المؤلف، ومعه حق، بأنها مخيفة. فقد أمكن تخمين الربع من مجموع كلمات السر التي اشتملت عليها الدراسة، وذلك باتباع الاستراتيجية الآتية:

1. جرّب اسم المستخدم، والأحرف الأولى، واسم الحساب، وغير ذلك من المعلومات الشخصية. وقد تم تجربة ما مجموعه 130 تبديلة مختلفة لكل مستخدم.
2. جرّب كلمات من مختلف القواميس. قام المؤلف بتجميع قاموس يضم أكثر من 60,000 كلمة موجودة في عدة مصادر من بينها قاموس النظام نفسه على الإنترنت، وغيره من القوائم المختلفة كما هو مبين في الجدول.
3. جرّب تعديلات مختلفة على الكلمات في الخطوة 2. يشمل ذلك تغيير الحرف الأول إلى حرف كبير (uppercase)، وقلب الكلمة كلها إلى أحرف كبيرة، وعكس ترتيب أحرف الكلمة، وتغيير الحرف "o" إلى الرقم صفر "0"، وهكذا. أضافت تلك التبديلات مليون كلمة أخرى إلى القائمة.
4. تبديلات أخرى تتعلق بجعل الأحرف كبيرة على الكلمات في الخطوة 2 التي لم تؤخذ في الاعتبار في الخطوة 3. أضاف ذلك ما يقرب من 2 مليون كلمة أخرى إلى القائمة.

وهكذا اشتملت التجربة على ما يقرب من 3 مليون كلمة. باستخدام أسرع أجهزة الحاسب من نوع Thinking Machines المذكورة أعلاه، فإن الوقت اللازم لتشفير كل تلك الكلمات باستخدام كل الاحتمالات الممكنة لقيمة الملح سيكون أقل من ساعة. جديرٌ بالذكر أن مثل هذا البحث الدقيق يمكن أن يؤدي إلى نجاح بنسبة 25%، في حين أن النجاح في تخمين كلمة سر واحدة فقط قد يكفي لتحصيل مجموعة كبيرة من الامتيازات على النظام المهاجَم.

❖ **التحكم في الوصول:**

من طرق إحباط الهجوم على كلمة السر منع الخَصْم من الوصول إلى ملف كلمات السر. فإذا جُعل الجزء الخاص بكلمات السر المُشفَّرة من الملف بحيث يمكن الوصول إليه فقط من قِبَل مستخدم متميّز، فلن يتمكن الخَصْم من قراءتها دون أن يعرف بالفعل كلمة السر الخاصة بذلك المستخدم. يشير [SPAF92a] إلى عدد من أوجه القصور في تلك الاستراتيجية:

- يتعرض عدد من النظم – بما في ذلك معظم أنظمة يونيكس – لعمليات اقتحام غير متوقعة. فبمجرد نجاح المهاجم في النفاذ إلى النظام، فإنه يود الحصول على مجموعة من كلمات السر ليستطيع استخدام حسابات مختلفة لجلسات دخول مختلفة وذلك لتقليل خطر انكشافه. أو قد يرغب مستخدم له حساب في استعمال حساب مستخدم آخر للوصول إلى بيانات متميّزة أو لتخريب النظام.
- قد يؤدي حادث يتعلق بالحماية إلى جعل ملف كلمة السر قابلاً للقراءة، ومن ثم تعريض كل الحسابات للخطر.
- بعض المستخدمين لهم حسابات على أجهزة أخرى ضمن مجالات أخرى للحماية وهم يستخدمون نفس كلمة السر لتلك الحسابات. وعليه، فإذا أمكن قراءة كلمات السر من قِبَل أي شخص على أحد الأجهزة، فقد يُصبح جهاز آخر في مكان آخر عُرضةً للخطر.

وهكذا، فإن الاستراتيجية الأكثر فعالية هي إرغام المستخدمين على اختيار كلمات سر يصعب تخمينها.

9-3-2 استراتيجيات اختيار كلمة السر

الدرس الذي يمكن استخلاصه من التجارب التي وصفناها آنفاً (انظر الجدولين 9-3 و 9-4) هو أن المستخدمين إذا تُركوا وشأنهم، فإن كثيراً منهم يختارون كلمة سر قصيرة جداً أو من السهل للغاية تخمينها. وعلى النقيض، إذا تم

تخصيص كلمات سر للمستخدمين تتكون من ثمانية أحرف قابلة للطباعة تُختار عشوائيا فإن كسر كلمات السر يصبح ضرباً من المستحيل عملياً. ولكن في المقابل سيُصبح من المستحيل بنفس القدر تقريباً على معظم المستخدمين تذكُّر كلمات السر الخاصة بهم. لحسن الحظ، حتى لو قمنا بالحد من فضاء كلمة السر لسلاسل أحرف يسهل حفظها بشكلٍ معقول، فإن ذلك الفضاء يبقى كبيراً جداً بحيث لن يسمح بعملية التكسير في الواقع العملي. وهدفنا إذن هو القضاء على كلمات السر سهلة التخمين وفي الوقت نفسه السماح للمستخدم باختيار كلمة سر يسهل تذكرها.

الأساليب الأربعة الأساسية المستخدمة في هذا الصدد هي:

- توعية المستخدم.
- توليد كلمات السر بواسطة الحاسب.
- الفحص التفاعلي لكلمة السر.
- الفحص الاستباقي لكلمة السر.

يمكن أن يُخطَر المستخدمون بأهمية استخدام كلمات سر صعبة التخمين، وأن يتم تزويدهم بمبادئ توجيهية لمساعدتهم في اختيار كلمات سر قوية. ليس من المرجّح أن تنجح مثل تلك الاستراتيجية لتوعية المُستخدِم في معظم المنشآت، ولاسيما حيث يوجد عددٌ كبير من المستخدمين وتكثر عمليات التجديد فيهم. كثير من المستخدمين سيتجاهلون ببساطة المبادئ التوجيهية. البعض الآخر قد لا تتوفر لديه الحكمة الكافية لاختيار كلمة سر قوية. فعلى سبيل المثال، هناك عدد من المستخدمين يعتقدون، وهذا خطأ، أن عكس ترتيب أحرف كلمة السر أو قلب الحرف الأخير منها إلى حرف كبير يجعل كلمة السر غير قابلة للتخمين.

الجدول 9-4 كلمات السر التي أمكن تخمينها من عينة تضم 13,797 حساباً [KLEI90].

نوع كلمة السر	حيز البحث	عدد التطابقات	النسبة المئوية لكلمات السر التي أمكن مطابقتها	نسبة الكلفة/الفائدة[a]
اسم المستخدم/الحساب	130	368	2.7%	2.830
تسلسل أحرف	866	22	0.2%	0.025
أعداد	427	9	0.1%	0.021
صيني	392	56	0.4%	0.143
أسماء أماكن	628	82	0.6%	0.131
أسماء مشهورة	2239	548	4.0%	0.245
أسماء إناث	4280	161	1.2%	0.038
أسماء ذكور	2866	140	1.0%	0.049
أسماء غير مشهورة	4955	130	0.9%	0.026
خرافات وأساطير	1246	66	0.5%	0.053
شيكسبيريات	473	11	0.1%	0.023
مصطلحات رياضة	238	32	0.2%	0.134
خيال علمي	691	59	0.4%	0.085
أفلام ونجوم سينما	99	12	0.1%	0.121
أفلام كرتونية	92	9	0.1%	0.098
مشاهير	290	55	0.4%	0.190
تعبيرات وأنماط	933	253	1.8%	0.271
ألقاب	33	9	0.1%	0.273
مصطلحات بيولوجية	58	1	0.0%	0.017
قاموس النظام	19683	1027	7.4%	0.052
أسماء أجهزة	909	132	1.0%	0.015
اختصارات	14	2	0.0%	0.143
إنجيل الملك جيمس	7525	83	0.6%	0.011
كلمات منوّعة	3212	54	0.4%	0.017
لغة اليديش	56	0	0.0%	0.000
كويكبات	2407	19	0.1%	0.007
المجموع	62727	3340	24.2%	0.053

[a] تُحسب بقسمة عدد التطابقات على حيز البحث. كلما زاد عدد الكلمات التي يتطلب الأمر تجربتها للحصول على تطابق، قلّت نسبة الكلفة/الفائدة.

استخدام كلمات سر من توليد الحاسب له مشاكله أيضاً. إذا كانت كلمات السر عشوائية تماما في طبيعتها، فلن يستطيع المستخدمون تذكّرها. حتى لو كانت كلمة السر يمكن نطقها، فقد يجد المُستخدِم صعوبة في تذكرها، ومن ثَمَّ سيميل إلى كتابتها للاحتفاظ بها. بشكلٍ عام، ارتبطت كلمات السر المولّدة بواسطة الحاسب بتاريخ سيء من حيث قبول المستخدمين لها. تُعرّف الوثيقة FIPS PUB 181 واحداً من أفضل المولدات الآلية لكلمات السر تصميماً. لا تحتوي الوثيقة فقط على وصفٍ للطريقة ولكن تحتوي أيضاً على سردٍ كاملٍ لكود برنامج الخوارزمية بلغة C. تقوم الخوارزمية بتوليد كلمات عن طريق تشكيل مقاطع قابلة للنطق ونظمها في سلسلة لتكوين كلمة السر. وتستخدم الخوارزمية مولد أعداد عشوائية لإنتاج تدفق عشوائي من الأحرف لبناء المقاطع والكلمات.

في استراتيجية الفحص التفاعلي لكلمة السر يقوم النظام دوريا بتشغيل برنامجه الخاص لتكسير كلمات السر بحثاً عن كلمات السر التي يمكن تخمينها. فيقوم النظام بإلغاء كلمات السر التي أمكن تخمينها وإخطار المُستخدِم بذلك. ولهذا الأسلوب عددٌ من المثالب. فهو أولاً يستنزف كثيراً من الموارد لإنجاز المهمة على مايرام. ولأن خَصْماً عنيداً تمكّن من سرقة ملف كلمة السر سيكون بوسعه تكريس وقت المعالج بأكمله للمهمة لساعات أو حتى لأيام، فإن أسلوب الفحص التفاعلي لكلمة السر يعاني من قصور واضح. وعلاوةً على ذلك، فستبقى كلمات السر الضعيفة تمثل نقاط ضعف في النظام إلى أن يكتشفها الفاحص التفاعلي لكلمة السر.

يُعدُّ أسلوب الفحص الاستباقي لكلمة السر من أفضل الاستراتيجيات الواعدة لتحسين أمن كلمات السر. وفي هذه الطريقة، يُسمح للمستخدم باختيار كلمة السر الخاصة به. غير أنه في وقت الاختيار يقوم النظام بفحص كلمة السر لتحديد ما إذا كانت مقبولة، وإلا فإنه يرفضها. ويقوم أسلوب الفحص هذا على الفلسفة التي مؤداها أنه بتوجيهٍ كافٍ من النظام، يمكن للمستخدمين اختيار كلمات سر يسهل تذكرها من فضاء كلمات السر الواسع نسبياً التي يُستبعد تخمينها في هجوم يعتمد على كلمات القاموس.

تكمن الفكرة في الفحص الاستباقي لكلمة السر في تحقيق التوازن بين قوة كلمة السر ومدى تَقبُّل المستخدمين لها. فإذا كان النظام يرفض كثيراً من كلمات السر، فسيشتكي المستخدمون من أنه من الصعب جداً اختيار كلمة سر. وفي المقابل إذا كان النظام يستخدم خوارزمية بسيطة لتحديد ما هو مقبول، فسيوفر ذلك توجيهاً لقراصنة تكسير كلمات السر لمراجعة أساليب التخمين التي يستخدمونها. فيما تبقى من هذا الجزء الفرعي، سنتناول بعض الطرق الممكنة للفحص الاستباقي لكلمة السر.

الطريقة الأولى هي مجرد نظام بسيط لفرض بعض القواعد. فعلى سبيل المثال، يمكن فرض تطبيق القواعد الآتية:

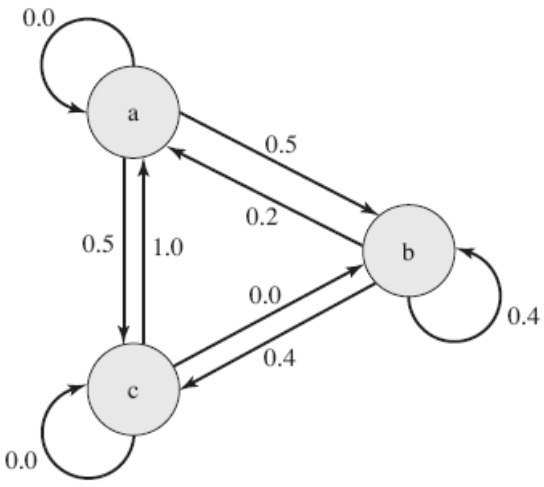
- يجب أن لا يقل طول كلمة السر عن ثمانية أحرف.
- في أول ثمانية أحرف، يجب أن تتضمن كلمة السر واحداً على الأقل من كلٍّ من: الأحرف الكبيرة، والأحرف الصغيرة، والأرقام (من 0 إلى 9)، وعلامات الترقيم.

يمكن أن يصحب تلك القواعد بعض التوجيه للمستخدم. فرغم أن هذه الطريقة تُعدُّ أفضل من مجرد توعية المستخدم، فقد لا تكون كافية لإحباط جهود تكسير كلمات السر. حيث يلفت هذا الأسلوب انتباه قراصنة تكسير كلمات السر إلى الكلمات التي لا ينبغي أن يجربوها، ولكنه مع ذلك قد يسمح بالقيام بتكسير كلمات السر.

من الإجراءات الممكنة أيضاً تجميع قاموس كبير من كلمات السر "السيئة". وعندما يختار مستخدم كلمة سر يقوم النظام بالتأكد من أنها لا توجد في القائمة الممنوعة. هناك مشكلتان في هذه الطريقة:

- الحيز: يتعين أن يكون القاموس كبيراً جداً لتكون الطريقة فعالة. فعلى سبيل المثال، يحتاج القاموس المُستخدَم في دراسة بوردو [SPAF92a] لأكثر من 30 ميجابايت لتخزينه.

- الوقت: الوقت اللازم للبحث خلال قاموس كبير سيكون كبيراً في حد ذاته. وعلاوةً على ذلك، فلفحص التباديل المحتملة لكلمات القاموس، إما أن يتم تضمين تلك التباديل في القاموس، الأمر الذي سيجعله ضخماً بحق، أو أن تكون عملية البحث مصحوبةً بعمليات معالجة مما يستهلك وقتاً إضافياً طويلاً.



حيث: $M = \{3, \{a, b, c\}, T, 1\}$

$$T = \begin{bmatrix} 0.0 & 0.5 & 0.5 \\ 0.2 & 0.4 & 0.4 \\ 1.0 & 0.0 & 0.0 \end{bmatrix}$$

مثال لسلسلة أحرف يُحتمل أن تنتمى لتلك اللغة: abbcacaba

مثال لسلسلة أحرف يُحتمل ألا تنتمي لتلك اللغة: aacccbaaa

الشكل 9-5 مثال لنموذج ماركوف.

هناك أسلوبان واعدان لتطوير نظامٍ للفحص الاستباقي لكلمة السر يقوم على رفض الكلمات الموجودة على قائمة معينة. ويعتمد أحد هذين الأسلوبين على تطوير نموذج ماركوف لتوليد كلمات السر التي يمكن تخمينها [DAVI93]. ويوضّح الشكل 9-5 مثالاً مبسطاً لذلك النموذج والذي يبيّن لغةً تتألف من أبجدية مكونة من ثلاثة أحرف. وتُمثل حالة النظام في أي وقت بآخر حرفٍ تم توليده. وتمثل القيمة المرتبطة بالانتقال من حالة إلى أخرى احتمال كون الحرف التالي يتبع الحرف الحالي في اللغة التي يمثلها النموذج. فكما بالشكل، وبافتراض أن الحرف الحالي هو a فإن احتمال كون الحرف التالي هو b، يساوي 0.5.

بشكلٍ عام، يُمثّل نموذج ماركوف برباعية $[m, A, \mathbf{T}, k]$، حيث m هي عدد الحالات في النموذج، وA هو فضاء (مجموعة set) تلك الحالات، و$\mathbf{T}$ مصفوفة احتمالات الانتقال، و k درجة النموذج . لنموذج من الدرجة k، يعتمد احتمال حدوث انتقال إلى حرف معين على الأحرف الـ k الأخيرة التي تم توليدها. ويبيّن الشكل 9-5 نموذجاً بسيطاً من الدرجة الأولى ($k = 1$).

تناول مؤلفو الدراسة تطوير نموذج من الدرجة الثانية واستخدامه. في البداية يتم إنشاء قاموس من كلمات السر التي يمكن تخمينها. بعد ذلك يتم حساب مصفوفة احتمالات الانتقال كالآتي:

1. قم بتعيين مصفوفة التردد $\mathbf{f}$، حيث $\mathbf{f}(i, j, k)$ هو عدد مرات تكرار ثلاثي الأحرف الذي يشمل الأحرف i وj وk. فعلى سبيل المثال تتضمن كلمة السر parsnips ثلاثيات الأحرف: par، ars، rsn، sni، nip، ips.
2. لكل ثنائي أحرف ij، احسب $\mathbf{f}(i, j, \infty)$ كالعدد الإجمالي لثلاثيات الأحرف التي تبدأ بـ ij. على سبيل المثال، سيمثّل $\mathbf{f}(a, b, \infty)$ العدد الإجمالي لثلاثيات الأحرف: aba ، abb، abc، وهكذا.
3. احسب عناصر $\mathbf{T}$ كالآتي:

$$\mathbf{T}(i, j, k) = \frac{\mathbf{f}(i, j, k)}{\mathbf{f}(i, j, \infty)}$$

والنتيجة هي نموذج يعكس هيكل الكلمات في القاموس. وباستخدام هذا النموذج، فإن السؤال: "هل كلمة السر هذه سيئة؟" ويتحول إلى السؤال: "هل قام نموذج ماركوف هذا بتوليد سلسلة الأحرف تلك (كلمة السر)؟" ولكلمة سر بعينها، يمكن تعيين كل احتمالات الانتقال لكل ثلاثيات الأحرف التي تتضمنها. وبعد ذلك يمكن استخدام بعض الاختبارات الإحصائية القياسية لتحديد ما إذا كان من المرجّح أو من غير المرجّح خضوع كلمة السر تلك لذلك النموذج. ويتم رفض كلمات السر التي يُحتمل توليدها من النموذج. ويذكر مؤلفو الدراسة أنهم حصلوا على نتائج جيدة باستخدام نموذج من الدرجة الثانية. واستطاع النظام الذي طوروه اكتشاف كل الكلمات غير المناسبة في قاموسهم ولم يستبعد كثيراً من كلمات السر الجيدة، مما ينبئ عن أنه سيكون مقبولاً لدى المستخدمين.

أما سبافورد [SPAF92a, SPAF92b] فقد استخدم طريقة مختلفة تماماً تعتمد على استخدام مرشّح بلوم [BLOO70]. في البداية سنشرح طريقة عمل مرشح بلوم. يتألف مرشح بلوم من الدرجة k من عدة دوال تحوير (hash functions) مستقلة عددها k، ولنرمز لها بـ $H_1(x), H_2(x),\ldots, H_k(x)$، حيث تقوم كل دالة بتحويل كل كلمة سر إلى قيمة محوّرة تتراوح مابين 0 و N-1، أي أن:

$$H_i(X_j) = y \qquad 1 \le i \le k; \qquad 1 \le j \le D; \qquad 0 \le y \le N - 1$$

حيث:

X_j = كلمة السر رقم j في قاموس كلمات السر.

D = عدد الكلمات في قاموس كلمات السر.

بعد ذلك يتم تنفيذ الإجراء الآتي على القاموس:

1. يتم تعريف جدول تحوير (hash table) يتكون من N بت، مع وضع كل البتات في البداية عند القيمة 0.
2. لكل كلمة سر، احسب قيم دوال التحوير الـ k لها، وضَع البتات المناظرة لها في جدول التحوير عند القيمة 1. ومن ثَمَّ فإذا كانت قيمة الدالة $H_i(X_j)$

عند (i, j) تساوي 67، فإن البت 67 في جدول التحوير يتم وضعه عند القيمة 1. إذا كانت قيمة ذلك البت 1 بالفعل فإنه يُترك عند القيمة 1.

عندما تَرِد كلمة مرور جديدة للفاحص، يقوم بحساب قيم دوال التحوير لها والتي عددها k. إذا كانت كل البتات المناظرة لقيم تلك الدوال في جدول التحوير لها القيمة 1 فإن كلمة السر تلك يتم رفضها. غير أنه سيكون هناك أيضاً بعض الخطأ الإيجابي الكاذب (false positives) (أي كلمات سر ليست في القاموس ولكنها تؤدي إلى تطابق في جدول التحوير). ولتوضيح ذلك، لنأخذ في الاعتبار نظاماً يستخدم دالَّتَي تحوير. لنفترض أن كلمتي السر *undertaker* و *hulkhogan* موجودتان في القاموس، في حين أن كلمة السر xG%#jj98 ليست في القاموس. لنفترض أيضاً ما يأتي:

H_1(undertaker) = 25	H_1(hulkhogan) = 83	H_1(xG%#jj98) = 665
H_2(undertaker) = 998	H_2(hulkhogan) = 665	H_2(xG%#jj98) = 998

إذا ما قُدمت كلمة السر xG%#jj98 إلى النظام فإنها ستُرفض، رغم أنها ليست في القاموس. وإذا وُجد عدد كبير من الأخطاء الإيجابية الكاذبة تلك سيكون من الصعب على المستخدمين اختيار كلمة السر. ولذلك، نود تصميم دوال التحوير بحيث تقلل الأخطاء الإيجابية الكاذبة لأقل حد ممكن. ويمكننا إثبات أن احتمال حدوث خطأ إيجابي كاذب يُعبّر عنه تقريباً كما يأتي:

$$P \approx (1 - e^{kD/N})^k = (1 - e^{k/R})^k$$

أو ما يكافئ

$$R \approx \frac{-k}{\ln(1 - P^{1/k})}$$

حيث:

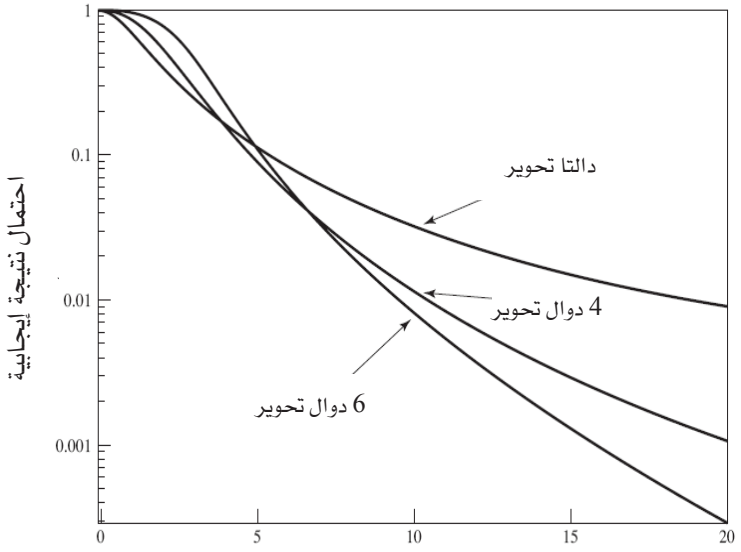
k = عدد دوال التحوير.

N = عدد البتات في جدول التحوير.

D = عدد الكلمات في القاموس.

$R = N/D$ = النسبة بين طول جدول التحوير (بتاً) إلى حجم القاموس (كلمة).

يمثل الشكل 9-6 الاحتمال P مرسوماً كدالة في R عند قيم مختلفة من k. لنفترض أن لدينا قاموساً يضم مليون كلمة ونود أن يكون احتمال رفض كلمة سر غير موجودة في القاموس هو 0.01. إذا اخترنا 6 دوال تحوير تكون النسبة المطلوبة هي $R = 9.6$. وعليه فإننا نحتاج إلى جدول تحوير طوله 9.6×10^6 بتاً، أي حوالي 1.2 ميجابايت من التخزين. وفي المقابل، يتطلب تخزين القاموس كله حيزاً في حدود 8 ميجابايت. وهكذا، فإننا نحقق ضغطاً للبيانات بنسبة تبلغ حوالى 7. وعلاوة على ذلك، ينطوي فحص كلمة السر على حساب ست دوال تحوير بشكلٍ مباشر ولا يعتمد ذلك على حجم القاموس، بينما مع استخدام القاموس الكامل، يحتاج الأمر إلى كمية كبيرة من البحث [2].



الشكل 9-6 أداء مرشّح بلوم.

[2] يتضمن نموذج ماركوف ونموذج مرشح بلوم استخدام أساليب الاحتمالات. في حالة نموذج ماركوف، هناك احتمال قليل أن بعض كلمات السر في القاموس لن يتم اكتشافها واحتمال قليل أن بعض كلمات السر التي ليست في القاموس سيتم رفضها. في حالة مرشح بلوم، هناك احتمال قليل أن بعض كلمات السر التي ليست في القاموس سيتم رفضها. مرة أخرى نرى أن استخدام أساليب الاحتمالات يبسط الحل (على سبيل المثال، انظر الحاشية 1 في الفصل الخامس).

4-9 توصيات للمطالعة

يتضمن كلٌّ من [BACE00] و[PROC01] معالجة شاملة لموضوع اكتشاف التسلل. ومن المقالات المسحية عن الموضوع والتي تُعدُّ مفيدة رغم إيجازها: [KENT00] و[MCHU00]. أما [NING04] فيتضمن مسحاً لجوانب التقدم الذي حدث مؤخراً في تقنيات اكتشاف التسلل. كما يُعدُّ [HONE01] مرجعاً أساسياً عن المصايد الأمنية، حيث يقدم تحليلاً مفصلاً عن أدوات القرصنة وطرقها.

[BACE00] Bace, R. *Intrusion Detection.* Indianapolis, IN: Macmillan Technical Publishing, 2000.

[BACE01] Bace, R., and Mell, P. *Intrusion Detection Systems.* NIST Special Publication SP 800-31, November 2000.

[HONE01] The Honeynet Project. *Know Your Enemy: Revealing the Security Tools, Tactics, and Motives of the Blackhat Community.* Reading, MA: Addison-Wesley, 2001.

[KENT00] Kent, S. "On the Trail of Intrusions into Information Systems," *IEEE Spectrum*, December 2000.

[MCHU00] McHugh, J.; Christie, A.; and Allen, J. "The Role of Intrusion Detection Systems." *IEEE Software*, September/October 2000.

[NING04] Ning, P., et al. "Techniques and Tools for Analyzing Intrusion Alerts." *ACM Transactions on Information and System Security*, May 2004.

[PROC01] Proctor, P., *The Practical Intrusion Detection Handbook.* Upper Saddle River, NJ: Prentice Hall, 2001.

5-9 مصادر للمعلومات على الويب

نوصي بالمواقع الآتية على الويب:

- CERT Coordination Center (مركز سيرت للتنسيق): منظمة نشأت عن فريق الاستجابة لطوارىء الحاسب الذي شكّلته وكالة مشاريع بحوث الدفاع المتقدمة DARPA. يوفر الموقع معلومات جيدة عن تهديدات أمن الإنترنت، ونقاط الضعف، وإحصائيات عن عمليات الهجوم.
- Honeynet Project (مشروع المصايد الأمنية): مشروع بحثي لدراسة أساليب الهجوم التي يستخدمها القراصنة وتطوير منتجات للمصايد الأمنية.
- Honeypots (المصايد الأمنية): مجموعة جيدة من الأوراق البحثية والمقالات الفنية.
- Intrusion Detection Working Group (مجموعة العمل الخاصة باكتشاف التسلل): يضم الموقع جميع الوثائق من إعداد تلك المجموعة.

6-9 مصطلحات رئيسة

audit record	سجل مراجعة
base-rate fallacy	مغالطة المعدل الأساسي
Bayes' Theorem	نظرية بايز
intruder	المتسلل
intrusion detection	اكتشاف التسلل، اكتشاف الاختراق الامني
intrusion detection exchange format	صيغة تبادل البيانات لاكتشاف التسلل
password	كلمة السر، كلمة المرور
rule-based intrusion detection	اكتشاف التسلل بناءً على قواعد
salt	الملح
statistical anomaly detection	اكتشاف الشذوذ الإحصائي
honeypot	مصيدة أمنية

7-9 أسئلة للمراجعة ومسائل

1-7-9 أسئلة للمراجعة

1-9 اذكر وعرّف بإيجاز ثلاث فئات من المتسللين.

2-9 اذكر اثنين من الأساليب المستخدمة بكثرة لحماية ملف كلمات السر.

3-9 اذكر ثلاث فوائد لاستخدام نظم لاكتشاف التسلل.

4-9 ما الفرق بين اكتشاف الشذوذ الإحصائي واكتشاف التسلل بناءً على قواعد؟

5-9 أذكر بعض المقاييس المفيدة في اكتشاف التسلل بناءً على البروفيل.

6-9 ما الفرق بين اكتشاف الشذوذ الإحصائي بناءً على قواعد و التعرف على الاختراق بناءً على قواعد؟

7-9 ما المقصود بالمصيدة الأمنية؟

8-9 ما المقصود بالملح في سياق إدارة كلمات السر بنظام التشغيل يونيكس؟

9-9 اذكر وعرّف بإيجاز أربعة أساليب تُستخدَم لتجنب استعمال كلمات سر يمكن تخمينها.

2-7-9 مسائل

1-9 تسببت سيارة أجرة في حادث مرور مميت ليلاً ثم هربت بعده. وتعمل بالمدينة شركتان لسيارات الأجرة: شركة السيارات الخضراء وشركة السيارات الزرقاء. تنامى إلى علمك المعلومات الآتية:

- 85% من سيارات الأجرة في المدينة خضراء و15% زرقاء.
- ذكر شاهد عيان أن سيارة الأجرة المتورطة في الحادث كانت زرقاء. اختبرت المحكمة موثوقية شهادة الشاهد تحت نفس الظروف السائدة ليلة الحادث واستنتجت أنه كان بوسعه التعرف على اللون الصحيح للسيارة في 80%

من الحالات. ما احتمال أن تكون السيارة المتورطة في الحادث زرقاء وليست خضراء؟.

2-9 بافتراض أن كلمات السر تتألف من تباديل من أربعة أحرف يتم اختيارها من بين 26 حرفاً أبجدياً، وأن الخَصْم قادرٌ على تجربة كلمات السر بمعدل كلمة سر واحدة كل ثانية.

a. فإذا افترضنا أيضاً عدم وجود رد فعل لما يقوم به الخَصْم حتى ينتهي من كل محاولة، ما الوقت المتوقع لاكتشاف كلمة السر الصحيحة؟
b. على افتراض وجود تغذية راجعة على شكل مؤشر يخبر الخَصْم عند إدخال حرف غير صحيح أثناء المحاولة، ما الوقت المتوقع لاكتشاف كلمة السر الصحيحة؟

3-9 بافتراض عناصر مصدر طول كلٍّ منها k خانة يتم مقابلتها بشكلٍ منتظم بعناصر هدف طول كلٍّ منها p خانة. وإذا كانت كل خانة يمكن أن تأخذ قيمة واحدة من بين r قيمة، فإن عدد عناصر المصدر يكون r^k وعدد عناصر الهدف يكون r^p. فإذا عُلم أن عنصر المصدر x_i يقابل عنصر الهدف y_j:

a. ما احتمال أن يقوم خَصْم باختيار عنصر المصدر الصحيح في محاولة واحدة؟
b. ما احتمال أن يقوم الخَصْم باختيار عنصر مصدر مختلف x_k (حيث $x_i \neq x_k$) يُنتج نفس عنصر الهدف y_j ؟
c. ما احتمال أن يقوم خَصْم باختيار عنصر الهدف الصحيح في محاولة واحدة؟

4-9 يقوم مولد كلمات سر باختيار قطعتين عشوائيا لتوليد كلمة سر تتألف من 6 أحرف. CVC (حرف ساكن - حرف متحرك - حرف ساكن) حيث V = <a, e, i, o, u> وC = مكمّل V.

a. ما هو العدد الكلي لكلمات السر الممكنة؟
b. ما هو احتمال أن يتمكن خَصْم من تخمين كلمة سر بشكلٍ صحيح؟

5-9 بافتراض أن كلمات السر تقتصر على استخدام 95 حرفاً بصيغة آسكي قابلة للطباعة، وأن جميع كلمات السر تتألف من 10 أحرف. إذا فرضنا استخدام مكسّر

كلمات سر يقوم بالتشفير بمعدل 6.4 مليون تشفيرة/ ثانية. كم من الوقت يستغرق تجريب كل كلمات السر الممكنة بشكلٍ استقصائي على نظام يونيكس؟

6-9 نظراً للمخاطر المعروفة لمنظومة كلمات السر على نظام التشغيل يونيكس، توصي وثائق نظام التشغيل SunOS-4.0 بإزالة ملف كلمات السر والاستعاضة عنه بملف متاح للقراءة من قِبَل للجمهور يسمّى etc/publickey/. يتألف المُدخل الخاص بالمُستخدِم A في ذلك الملف من معرّف المُستخدِم (ID_A)، والمفتاح العام للمستخدم (PU_a)، والمفتاح الخاص المناظر (PR_a). يتم تشفير ذلك المفتاح الخاص باستخدام خوارزمية DES بمفتاح مستمد من كلمة سر الدخول الخاصة بالمُستخدِم (P_a). وعندما يحاول المُستخدِم A الدخول على النظام، يقوم النظام بإزالة تشفير $E[P_a, PR_a]$ للحصول على المفتاح الخاص للمستخدم (PR_a).

a. يتحقق النظام بعد ذلك من أن كلمة السر التي تم إدخالها صحيحة. كيف؟

b. كيف يمكن لخَصْم الهجوم على هذا النظام؟

7-9 التشفير المُستخدَم لكلمات السر في نظام يونيكس أحادي الاتجاه، ولايمكن عكسه. وعليه، فما مدى دقة القول بأن العملية، في الواقع، تتضمن الحصول على كود تحوير (hash code) لكلمة السر وليس تشفيراً لها؟

8-9 ذكرنا أن استخدام الملح في منظومة كلمة السر على نظام يونيكس يزيد من صعوبة تخمين كلمة السر بمعامل قدره 4096. غير أن قيمة الملح تخزّن واضحةً غير مُشفَّرة في نفس المُدخل مع كلمة السر المناظرة بعد تشفيرها. وعليه، فإن هذين الحرفين يكونان معروفين للمهاجم ولذا لن يحتاج إلى تخمينهما. لماذا أكدنا على أن استخدام الملح يزيد الأمن؟

9-9 إذا كنت قد نجحت في الإجابة على المسألة السابقة وفهمت أهمية استخدام الملح، فهذا سؤال آخر لك. أليس من الممكن إحباط كل هجمات تكسير كلمات السر تماماً عن طريق زيادة كبيرة في حجم الملح، ليصبح، مثلاً 24 أو 48 بتاً؟

10-9 في مرشح بلوم الذي تناولناه في الجزء 9-3 إذا افترضنا أن:

k = عدد دوال التحوير

N = عدد البتات في جدول التحوير

D = عدد الكلمات في القاموس

a. بيّن أن النسبة المتوقعة للبتات التي لها القيمة 0 في جدول التحوير يمكن التعبير عنها على النحو الآتي:

$$\phi = (1 - \frac{k}{N})^D$$

b. بيّن أن احتمال قبول كلمة مُدخلة بطريق الخطأ كأنها كلمة موجودة بالقاموس هو:

$$P = (1 - \phi)^k$$

c. بيّن أن التعبير السابق يمكن تقريبه إلي:

$$P = (1 - e^{-kD/N})^k$$

11-9 قم بتصميم نظامٍ للتحكم في الوصول إلى الملفات يمنح مستخدمين معينين صلاحية القراءة والكتابة في ملف بناءً على معايير الترخيص التي أقامها النظام. ينبغي أن تأخذ التعليمات الشكل:

READ (F, User A): محاولة المُستخدِم A قراءة الملف F

WRITE (F, User A): محاولة المُستخدِم A تخزين نسخة (ربّما معدّلة) من الملف F

لكل ملف سجل ترويسة يتضمن امتيازات الترخيص، أي قائمة بالمستخدمين المرخّص لهم بقراءة الملف وكتابته. وينبغي تشفير الملف بمفتاحٍ غير متاحٍ للمستخدمين ولكنه معروف فقط للنظام.

الملحق A-9

مغالطة المعدل الأساسي

نبدأ باستعراض بعض النتائج المهمة من نظرية الاحتمالات، ثم نقوم بتوضيح مغالطة المعدل الأساسي.

❖ الاحتمالات المشروطة والاستقلال

كثيراً ما نحتاج لمعرفة احتمال مشروط على وقوع حدث بعينه. يؤدي الشرط إلى إزالة بعض النتائج من فضاء العينة. على سبيل المثال، ما هو احتمال الحصول على المجموع 8 عند رمي زهرتين من زهر النرد إذا علمنا أن العدد على وجه زهرة واحدة منهما على الأقل هو عدد زوجي؟ ويمكننا أن نفكّر في الأمر على النحو الآتي: نظراً لأن إحدى الزهرتين زوجية والمجموع زوجي فينبغي أن تكون الزهرة الأخرى أيضاً زوجية. وعليه، فهناك ثلاث نتائج ناجحة لكلٍّ منها نفس القدر من الاحتمال: (2، 6)، (4، 4) (6، 2)؛ وذلك من بين عدد من الاحتمالات يبلغ: [36 - (عدد الأحداث التي تعطي وجهين فرديين)] = 3×3-36 = 27، ومن ثم يكون الاحتمال المطلوب حسابه = 3/27 = 1/9.

ويعرّف الاحتمال المشروط لحدث A رسمياً بافتراض أن حدثاً B قد وقع، ونرمز له بالرمز Pr[A/B]، كما يأتي:

$$\Pr[A \mid B] = \frac{\Pr[AB]}{\Pr[B]}$$

على افتراض أن Pr[B] ليست صفراً.

في مثالنا أعلاه، الحدث A = {المجموع = 8} ، والحدث B = {على الأقل زهرة واحدة تعطي عدداً زوجياً}. تمثل الكمية $\Pr[AB]$ احتمال وقوع الحدثين معاً، أي المجموع = 8 وتعطي زهرة واحدة على الأقل عدداً زوجيا. كما رأينا أعلاه هناك 3 حالات تحقق هذين الشرطين، ومن ثَمَّ فإن $\Pr[AB] = 3/36 = 1/12$. بعد قليل من التفكير سترى أن $\Pr[B] = 3/4$. الآن يمكننا حساب

$$\Pr[A \mid B] = \frac{1/12}{3/4} = \frac{1}{9}$$

وهذا يتفق مع أسلوبنا السابق في الحل.

يُطلق على الحدثين A وB مستقلين إذا كان $\Pr[AB] = \Pr[A]\times\Pr[B]$. ويمكن بسهولة إثبات أنه إذا كان A وB مستقلين فإن $\Pr[A|B] = \Pr[A]$ و$\Pr[B|A] = \Pr[B]$.

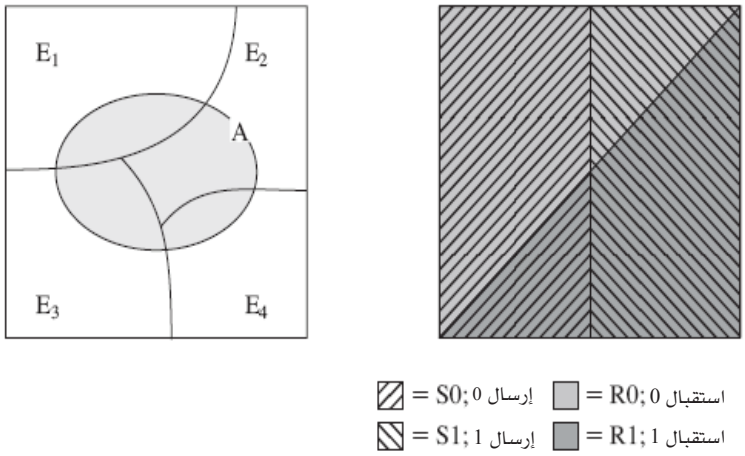
❖ نظرية بايز (Bayes Theorem)

من أهم النتائج لنظرية الاحتمالات ما يعرف بنظرية بايز. نحتاج أولاً لكتابة معادلة الاحتمال الكلي. لنفترض أن لدينا مجموعة من الأحداث المتنافية فيما بينها (mutually exclusive events): $E_1, E_2, \ldots, E_n$ بحيث إن الاتحاد (union) بين تلك الأحداث يغطي كل النواتج الممكنة. بافتراض حدث اعتباطي A، فإنه يمكننا إثبات أن:

$$\Pr[A] = \sum_{i=1}^{n} \Pr[A \mid E_i]\Pr[E_i] \qquad (1\text{-}9)$$

ويمكن التعبير عن نظرية بايز على النحو الآتي:

$$\Pr[E_i \mid A] = \frac{\Pr[A \mid E_i]\Pr[E_i]}{\Pr[A]} = \frac{\Pr[A \mid E_i]\Pr[E_i]}{\sum_{j=1}^{n} \Pr[A \mid E_j]\Pr[E_j]} \qquad (2\text{-}9)$$



(a) شكل لتوضيح المفهوم (b) مثال

الشكل 9-7 مفهوم الاحتمال الكلي ونظرية بايز.

يوضّح الشكل 9-7 (a) مفهوم الاحتمال الكلي ونظرية بايز. تُستخدم نظرية بايز لحساب "الاحتمالات البَعدية" (posterior odds)، أي احتمال أن شيئاً ما واقعٌ بالفعل في ضوء أدلة تؤيد ذلك. فعلى سبيل المثال، لنفترض أننا نرسل سلسلة من البتات بقيم 1 أو 0 على خط لنقل البيانات يعاني من وجود ضجيج (noise). ليكن S0 هو الحدث الذي يمثل إرسال 0 وS1 هو الحدث الذي يمثل إرسال 1 في وقت معيّن. بالمثل، ليكن R0 وR1 هما الحدثان اللذان يمثلان استقبال 0 أو 1 عند المُستقبل، على التوالي. افترض أن الاحتمالات عند المصدر هي: p = Pr[S1] وp -1= Pr[S0]. الآن نراقب الخط لتحديد مدى تكرار حدوث خطأ عند إرسال 1 وعند إرسال 0، ونقوم بحساب الاحتمالات الآتية: p_a = Pr[R0|S1] و p_b = Pr[R1|S0]. إذا تم استقبال 0، فإنه يمكننا حساب الاحتمال المشروط للخطأ، أي الاحتمال المشروط لإرسالنا 1 واستقبالنا 0. باستخدام نظرية بايز:

$$\Pr[S1 \mid R0] = \frac{\Pr[R0 \mid S1]\Pr[S1]}{\Pr[R0 \mid S1]\Pr[S1] + \Pr[R0 \mid S0]\Pr[S0]}$$
$$= \frac{P_a P}{P_a P + (1 - P_b)(1 - P)}$$

يوضّح الشكل 9-7 (b) المعادلة السابقة. ويتم تمثيل فضاء العينة في الشكل بمربع طول ضلعه الوحدة. يمثّل إرسال 0 (S0) بنصف المربع ويمثل إرسال 1 (S1) بالنصف الآخر، وعليه يكون Pr[S0] = Pr[S1] = 0.5. بالمثل، يمثّل استقبال 0 (R0) بنصف المربع ويمثل استقبال 1 (R1) بالنصف الآخر، وعليه يكون Pr[R0] = Pr[R1] = 0.5. في المساحة التي تمثل S0، لاحظ أن ربع تلك المساحة ينتمي للمساحة التي تمثل R1، ومن ثم فإن Pr[R1|S0] = 0.25. يمكن استنتاج بقية الاحتمالات المشروطة من الشكل بنفس الطريقة.

❖ مغالطة المعدل الأساسي

خذ في الاعتبار السيناريو الآتي. أُجرِي لمريض الاختبار الخاص بتشخيص مرضٍ معين، وكانت نتيجة الاختبار إيجابية (أي تشير إلى أن المريض لديه ذلك المرض). تتوافر لديك المعلومات الآتية:

- تبلغ دقة الاختبار 87% (أي اذا كان المريض مصابا بالمرض، فإنه في 87% من الوقت سيشير الاختبار إلى النتيجة الصحيحة، وإذا كان المريض ليس لديه المرض، فإنه في 87% من الوقت سيشير الاختبار أيضاً إلى النتيجة الصحيحة).
- ينتشر ذلك المرض في السكان بنسبة 1%.

إذا وجدنا نتيجة الاختبار إيجابية، فما هو احتمال أن المريض ليس لديه ذلك المرض؟ أي ما هو احتمال أن هذا إنذار كاذب؟ نحن بحاجة إلى نظرية بايز للحصول على الجواب الصحيح:

$$\Pr[well \mid positive] = \frac{\Pr[positive \mid well]\Pr[well]}{\Pr[positive \mid disease]\Pr[disease] + \Pr[positive \mid well]\Pr[well]}$$
$$= \frac{(0.13)(0.99)}{(0.87)(0.01) + (0.13)(0.99)} = 0.937$$

أي أنه في الغالبية العظمى من الحالات، عند اكتشاف حالة مرض باستخدام ذلك الاختبار، تكون النتيجة إنذار كاذب.

عُرضت هذه المشكلة، والتي استُخدمت في الدراسة [PIAT91]، على عدد من الأشخاص. أعطى معظم المشاركين الجواب 13%. أعطت الغالبية العظمى، بما في ذلك عدد من الأطباء، نسبة أقل من 50%. علّق كثيرٌ من الأطباء الذين أعطوا إجابة خاطئة بقولهم: "إذا كنتم على حق، فليس هناك مبرر لإجراء اختبارات سريرية!" يرجع السبب في أن معظم الناس لا يتوصلون إلى الإجابة الصحيحة إلى أنهم لا يأخذون في الاعتبار المعدل الأساسي لوقوع المرض (معدل انتشاره في السكان). يُعرف هذا الخطأ بمغالطة المعدل الأساسي.

كيف يمكن حل هذه المشكلة؟ إذا افترضنا أنه يمكنك تحسين معدل النتيجة الصحيحة للاختبار (سواءً كانت سلباً أو إيجاباً) ليصبح 99.9%. أي افترض القيم الجديدة Pr[postive|disease] = 0.999 وPr[negative|well] = 0.999. بإدخال الرقم الأول في المعادلة أعلاه، نحصل على Pr[well|positive] = 0.09، ومن ثم يقل معدل الإنذارات الكاذبة إلى 9%. هذا أفضل بكثير، ولكنه لا يزال غير مثالي. علاوةً على ذلك ، ومرة أخرى بافتراض 99.9% لدقة الاختبار، افترض الآن أن حالات الإصابة بالمرض تبلغ حالة واحدة فقط في كل 10000 نسمة، أي أن Pr[disease] = 0.0001. في هذه الحالة سنحصل على معدل إنذارات كاذبة يبلغ 91%. في الواقع العملي، وجد [AXEL00] أن الاحتمالات المرتبطة بأنظمة اكتشاف التسلل تُنتج معدلات مرتفعة بشكلٍ غير مُرضٍ للإنذارات الكاذبة.

10

الفصل العاشر

البرمجيات الخبيثة

محتويات الفصل:

10-1 الفيروسات والتهديدات المتعلقة بها

10-2 الإجراءات المضادة للفيروسات

10-3 هجمات حجب الخدمة الموزَّعة

10-4 توصيات للمطالعة

10-5 مصادر للمعلومات على الويب

10-6 مصطلحات رئيسة

10-7 أسئلة للمراجعة ومسائل

"ما هو مفهوم الدفاع؟ تفادي ضربة. ما الخاصية المميزة له؟ انتظار الضربة."
ـ من كتاب "عن الحرب" ، لكارل فون كلوزويز.

النقاط الرئيسة

- البرمجيات الخبيثة هي البرامج التي تُضمَّن أو تُدخَل عمداً في نظامٍ ما لغرض ضار.
- الفيروس هو قطعة من البرمجيات التي يمكن أن تصيب البرامج الأخرى عن طريق تعديلها؛ يشمل التعديل إدخال نسخة من برنامج الفيروس التي يمكن بدورها أن تصيب برامج أخرى.
- الدودة هي برنامج يمكن أن يتكاثر (يضاعف نفسه) ويرسل نسخاً منه من حاسب إلى آخر عبر توصيلات الشبكة. قد تنشط الدودة عند وصولها للحاسب لمضاعفة نفسها وإعادة الانتشار من جديد. بالإضافة إلى الانتشار، عادةً ما تقوم الدودة ببعض العمليات غير المرغوبة.
- حجب الخدمة (DoS) هو هجوم يهدف إلى محاولة منع مستخدمين شرعيين لخدمة معينة من استخدام تلك الخدمة.
- يُشن هجوم حجب الخدمة الموزَّع من عدة مصادر متعاونة فيما بينها.

سنناقش في هذا الفصل البرمجيات الخبيثة (malware)، وخصوصاً الفيروسات والديدان.

10-1 الفيروسات والتهديدات المتعلقة بها

لعل أكثر أنواع التهديدات لأنظمة الحاسب تطوراً هذه الأيام يتمثل في البرامج التي تستغل نقاط الضعف في أنظمة الحاسب. ويشمل هذا السياق كلاً من برامج التطبيقات والبرامج الخدمية كبرامج التحرير (editors) والمترجمات

(compilers). وسنبدأ هذا الجزء بنظرة عامة على مدى التهديد الذي تمثله البرمجيات الخبيثة. وسنركز في باقي الجزء على الفيروسات والديدان.

10-1-1 البرمجيات الخبيثة

تُشكِّل المصطلحات في هذا المجال مشكلة نظراً لعدم وجود اتفاق عام على كل المصطلحات، ولتداخل بعض التصنيفات. يعطي الجدول 10-1 (والمستمد من [SZOR05]) دليلاً مفيداً لمصطلحات البرمجيات الخبيثة.

يمكن تقسيم البرمجيات الخبيثة إلى صنفين: صنف يحتاج إلى برنامج مضيف، وآخر مستقل. الصنف الأول هو في الحقيقة أجزاء برامج (fragments of programs) لا يمكنها التواجد بشكل مستقل عن برنامج تطبيق فعلي معين، أو برنامج خدمي أو برنامج نظام. ومن أمثلة تلك البرمجيات الخبيثة: الفيروسات (viruses)، والقنابل المنطقية (logic bombs)، والمداخل الخلفية (backdoors). الصنف الآخر هو برامج مكتفية ذاتياً ويمكن جدولتها (scheduling) وتشغيلها من قِبَل نظام التشغيل. من أمثلة ذلك: برامج الزومبي (zombies) والديدان (worms).

يمكن أيضاً أن نُفرِّق بين تهديدات البرامج التي لا تتكاثر (أي تتضاعف بنسخ نفسها) وتلك التي تتكاثر. النوع الأول هو برامج أو أجزاء برامج تحتاج لمثير لكي تنشط؛ كالقنابل المنطقية، والمداخل الخلفية، وبرامج الزومبي. أما النوع الآخر فيشمل أجزاء برامج أو برامج مستقلة قد تُنتج عند تنفيذها نسخة أو أكثر من نفسها لكي تنشط لاحقاً على نفس النظام أو على نظام آخر؛ كالفيروسات والديدان.

في بقيّة هذا الجزء الفرعي، سنستعرض بسرعة بعض تلك الأصناف الرئيسة من البرمجيات الخبيثة باستثناء الفيروسات والديدان والتي سوف نغطيها بتفصيل أكثر لاحقاً في هذا الجزء.

الجدول 10-1: مصطلحات البرمجيات الخبيثة.

المصطلح	الوصف
الفيروس (Virus)	يُلحق نفسه ببرنامج وينشر نسخاً من نفسه للبرامج الأخرى.
الدودة (Worm)	برنامج ينشر نسخاً من نفسه (ينتشر) من حاسب لآخر.
القنبلة المنطقية (Logic bomb)	تُثار عند توفر ظروف معينة.
حصان طروادة (Trojan horse)	برنامج يتضمن وظائف أخرى غير متوقعة.
مدخل خلفي (باب المصيدة) (Backdoor (trapdoor))	تعديل بالبرنامج يسمح بالوصول غير المخوّل للوظائف.
استغلالات (Exploits)	كود محدد بنقطة ضعف وحيدة أو مجموعة من نقاط الضعف.
برامج تنزيل (Downloaders)	برامج تقوم بتثبيت عناصر أخرى على الجهاز محل الهجوم. تُرسَل عادةً عبر البريد الإلكتروني.
برنامج للوصول التلقائي للجذر (Auto-rooter)	أدوات للمهاجم تُستخدم في اختراق أجهزة جديدة عن بُعد.
علبة أدوات (لتوليد فيروسات) (Kit (virus generator))	مجموعة من الأدوات لتوليد فيروسات جديدة بشكل آلي.
برامج إرسال رسائل الدعاية (Spammer programs)	تُستخدم لإرسال كم كبير من رسائل االبريد الإلكتروني غير المرغوبة.
برامج الفيضان (Flooders)	تُستخدم لمهاجمة أنظمة حاسب متصلة بالشبكة بكم هائل من حركة مرور البيانات لتنفيذ هجوم حجب الخدمة (DoS).
مسجلات ضربات المفاتيح (Keyloggers)	تُستخدم لتسجيل الضربات على لوحة المفاتيح بجهاز مصاب.
الجذور الخفية (Rootkit)	مجموعة من الأدوات التي يستخدمها المهاجم بعد أن يكون قد تَمكّن من اختراق نظام حاسب وكَسَب وصولاً للمستوى الجذري (root-level access) (أي مستوى مدير النظام ومن ثَمَّ يكون له صلاحيات أكثر).
برنامج الزومبي (Zombie)	برنامج يُفعَّل على جهاز مصاب لشن هجوم على أجهزة أخرى.

❖ المداخل الخلفية (Backdoors):

يُعرَف المدخل الخلفي أيضاً بباب المصيدة (trapdoor)، وهو عبارة عن نقطة دخول سرية للبرنامج يمكن استغلالها من قِبَل شخصٍ ما – على دارية بها – للدخول بدون المرور عبر الإجراءات الأمنية المعتادة للوصول. استخدم المبرمجون المداخل الخلفية بشكل شرعي لعدة سنوات من أجل تنقيح البرامج واختبارها. ويتم ذلك عادةً عندما يطور المبرمج تطبيقاً له إجراء توثيق، أو تهيئة طويلة، تتطلب من المستخدم إدخال عدد من القيم المختلفة لتشغيل التطبيق. ولإجراء عملية تنقيح (debugging) للبرنامج، قد يرغب المُبرمج في كسب امتيازات خاصة أو تفادي إجراءات التوثيق أو التهيئة. وقد يرغب المبرمج أيضاً في ضمان وجود طريقة لتنشيط البرنامج في حالة حدوث خطأ ما في إجراء التوثيق المُبيَّت (built-in) كجزء من التطبيق. والمدخل الخلفي هو تعليمات بالبرنامج تتعرف على تسلسل معيّن من المدخلات أو تُثار بتشغيلها بواسطة اسم مستخدم معين أو عبر تسلسل معيّن غير محتمل من الأحداث.

تُشكِّل المداخل الخلفية تهديدات أمنية عندما يستخدمها مبرمجون عديمو الضمير لكسب وصول غير مُخوَّل لهم. وكان المدخل الخلفي الفكرة الأساسية لنقطة الضعف التي جُسِّدت في فيلم "مناورات حربية" (War Games). من الأمثلة الأخرى أنه أثناء تطوير نظام التشغيل مولتكس (Multics)، أُجريت اختبارات الاختراق بواسطة "فريق النمر" التابع للقوّات الجوية (لمحاكاة الخصوم).

تضمَّنت إحدى الخطط (التكتيكات) التي استُخدِمت إرسال رسالة مزيفة لتحديث نظام التشغيل إلى موقع يستخدم نظام مولتكس. وتضمَّنت رسالة التحديث حصان طروادة – سنصفه لاحقاً – يمكن تنشيطه من قِبَل مدخل خلفي يسمح لفريق النمر بالتمكن من الدخول. ونُفِّذ التهديد بشكلٍ جيد جداً، لدرجة أن مُطوِّري نظام مولتكس أنفسهم لم يتمكنوا من العثور عليه، حتى بعد علمهم بوجوده [ENGE80].

من الصعب تنفيذ أساليب تحكم في المداخل الخلفية بنظام التشغيل. يجب أن تركّز إجراءات الأمن على عملية تطوير البرنامج والأنشطة المختلفة لتحديث البرمجيات.

❖ القنبلة المنطقية (Logic Bomb):

تُعدُّ القنبلة المنطقية أحد أقدم أنواع تهديدات البرامج، فهي تسبق الفيروسات والديدان. إن القنبلة المنطقية هي تعليمات مُتضمَّنة في برنامج شرعي، ومحدد لها وقت "انفجار" عندما تتحقق ظروف معينة. ومن أمثلة تلك الظروف التي يمكن أن تستخدم كمثيرات لانفجار القنبلة المنطقية: وجود بعض الملفات أو عدم وجودها، أو حلول يوم معيّن من الأسبوع أو تاريخ معين، أو تشغيل التطبيق من قِبَل مستخدم معين. عند إثارة القنبلة قد تقوم بتعديل أو حذف بيانات أو ملفات بكاملها، أو تتسبب في توقّف الجهاز، أو إلحاق أي ضرر آخر. من الأمثلة المثيرة على كيفية استخدام القنابل المنطقية قضية تيم لويد (Tim Lloyd) التي أُدين فيها بتثبيت قنبلة منطقية كلَّفت صاحب العمل (شركة أوميجا للهندسة) أكثر من 10 مليون دولار، وأحبطت الخطة الاستراتيجية لتطوير الشركة، وأدَّت إلى تسريح 80 من العاملين بالشركة [GAUD00]. في النهاية حُكِم على لويد بـ 41 شهراً في السجن وبدفع 2 مليون دولار تعويضات.

❖ أحصنة طروادة (Trojan Horses):

حصان طروادة هو برنامج أو إجراء مفيد (أو مفيد في ظاهره) يتضمن تعليمات مخفية تقوم عند تنفيذها ببعض الأعمال غير المرغوبة أو الضارّة.

يمكن أن تُستخدَم برامج أحصنة طروادة للقيام بشكلٍ غير مباشر بعمليات لا يمكن لمستخدم غير مخوّل له أن ينجزها بشكلٍ مباشر. فعلى سبيل المثال، للوصول إلى ملفات مستخدم آخر على نظام مشترك، يمكن أن ينشئ

المستخدم برنامج حصان طروادة يغيّر عند تنفيذه صلاحيات الوصول للملفات والمخوَّلة للمستخدم الذي يقوم بتنفيذ البرنامج بحيث يمكن لأي مستخدم قراءة تلك الملفات. ويمكن لمنشئ حصان طروادة أن يغري المستخدمين الآخرين بتشغيل البرنامج وذلك بوضعه في مجلد مُشترك (بنظام الملفات) ويسمّيه بحيث يبدو برنامجاً مفيداً. ومثال لذلك برنامج يزعم أنه يسرد ملفات المستخدم بصيغة مرغوبة. بعد أن يُشغّل مستخدم آخر ذلك البرنامج الذي يتضمن حصان طروادة، يمكن لمؤلف البرنامج عندئذ الوصول للمعلومات الموجودة في ملفات ذلك المستخدم. ومن الأمثلة لبرنامج حصان طروادة يصعب اكتشافه برنامج مترجم (compiler) مُعدَّل لإدخال تعليمات إضافية إلى بعض البرامج، كبرنامج الدخول على النظام، أثناء القيام بترجمتها [THOM84]. تنشئ تلك التعليمات مدخلاً خلفياً في برنامج الدخول على النظام مما يسمح لمؤلف تلك التعليمات بالدخول للنظام باستخدام كلمة سر خاصة. ولا يمكن أبداً اكتشاف حصان طروادة هذا بقراءة تعليمات كود المصدر (source code) لبرنامج الدخول.

يُعدُّ تدمير البيانات من الدوافع الأخرى الشائعة لاستخدام حصان طروادة. يظهر البرنامج كما لو كان يؤدّي وظيفة مفيدة (مثلاً برنامج آلة حاسبة)، لكنّه قد يقوم أيضاً خلسةً وبهدوء بحذف ملفات المستخدم. على سبيل المثال كان المدير التنفيذي لشركة CBS ضحيةً لحصان طروادة دمّر كل المعلومات الموجودة بذاكرة حاسبه [TIME90]. كان حصان طروادة هذا مزروعاً في إجراء رسومات (graphics routine) معروضٍ على لوحة إعلانات إلكترونية (electronic bulletin board).

❖ برنامج زومبي (Zombie):

هو برنامج يسيطر خلسةً على حاسب آخر متصل بالإنترنت، وبعد ذلك يستخدم ذلك الحاسب لشنّ هجمات يصعب تتبّعها إلى مُنشئ الزومبي. يُستخدم الزومبي في هجمات حجب الخدمة (DoS)، عادةً ضد مواقع ويب مستهدفة. يتم

زراعة الزومبي على مئات الحاسبات التي تنتمي لجهات أخرى لا تكون محل اشتباه، وبعد ذلك يُستخدَم البرنامج لغمر موقع الويب المستهدف بحركة مرور شديدة عبر الإنترنت. يناقش الجزء 10-3 الزومبي ضمن سياق هجمات حجب الخدمة.

10-1-2 طبيعة الفيروسات

الفيروس هو جزء من برنامج يمكن أن "يصيب" برامج أخرى بتعديلها؛ يشمل التعديل تضمين نسخة من برنامج الفيروس يمكن بدورها أن تواصل عملها بإصابة برامج أخرى.

الفيروسات الحيوية (biological viruses) (أي التي تصيب الكائن الحي) هي فضلات صغيرة من الشفرة الجينية – DNA أو RNA – يمكن أن تسيطر على الوحدات الوظيفية لخلية حيّة وتخدعها لعمل آلاف النُسخ المطابقة للفيروس الأصلي. مثل نظيره الحيوي، يحمل فيروس الحاسب في تعليماته وصفةً لإنتاج نُسخ تامة من نفسه. وبشكلٍ اعتيادي يكون الفيروس مُتضمَّناً في برنامج على حاسب. بعد ذلك عندما يتصل الحاسب المصاب بجزء غير مصاب من البرمجيات تعبر نسخة جديدة من الفيروس إلى البرنامج الجديد. وهكذا، يمكن أن تنتشر العدوى من حاسب إلى حاسب من قِبَل المستخدمين الغافلين من خلال تبادل الأقراص أو إرسال البرامج إلى أحدهما الآخر عبر شبكة. في بيئة الشبكة، توفر القدرة للوصول للتطبيقات ولخدمات النظام على حاسبات أخرى مزرعة مثالية لانتشار الفيروس.

يمكن أن يعمل الفيروس أي شيء يمكن أن تعمله البرامج الأخرى. ويكمن الفرق الوحيد في أنه يربط نفسه ببرنامج آخر ويُنفَّذ خلسةً عند تشغيل البرنامج المضيف. وأثناء تنفيذ الفيروس يمكن أن يؤدي أي عملية كمحو الملفات والبرامج. وأثناء فترة العمر للفيروس الاعتيادي يمر بالمراحل الأربع الآتية:

- مرحلة الخمول (dormant phase): وفيها يكون الفيروس خاملاً إلى أن يتم تنشيطه في النهاية بحدثٍ ما؛ كحلول تاريخ معين، أو وجود برنامج أو ملف آخر، أو تجاوز سعة تخزين القرص حداً معيناً. ويلاحظ أنه ليست كل الفيروسات تمر بهذه المرحلة.
- مرحلة الانتشار (propagation phase): يضع الفيروس نسخة مماثلة من نفسه على البرامج الأخرى أو في مناطق معينة خاصة بنظام التشغيل على القرص. ويحتوي كل برنامج مصاب الآن على نسخة من الفيروس تدخل بدورها هي الأخرى مرحلة انتشار جديدة.
- مرحلة الإثارة (triggering phase): يُنشَّط الفيروس لأداء الوظيفة المطلوبة منه. وكما هو الحال مع مرحلة الخمول، يمكن أن تنشأ عملية الإثارة بسبب تشكيلة مختلفة من أحداث النظام، بما في ذلك عدد المرات التي قامت فيها تلك النسخة من الفيروس بنسخ نفسها.
- مرحلة التنفيذ (execution phase): وفيها تُؤدّى العملية المطلوبة. قد تكون تلك العملية غير مؤذية كعرض رسالة على الشاشة، أو قد تكون ضارة كتدمير البرامج وملفات البيانات.

تُؤدي معظم الفيروسات عملها بطريقة خاصة بنظام تشغيل معين، وفي بعض الحالات بطريقة خاصة أيضاً بعتاد أجهزة معين. ومن ثَمَّ فهي مُصمّمة لاستغلال نقاط الضعف الموجودة بأنظمة بعينها.

❖ تركيب الفيروس:

يمكن أن يُضمَّن الفيروس في بداية برنامج قابل للتنفيذ أو نهايته، كما يمكن أن يُضمَّن في أشكالٍ أخرى. يكمن المفتاح لعمل الفيروس في أنه عند استدعاء البرنامج المصاب يتم تنفيذ تعليمات الفيروس أولاً ثم بعد ذلك تنفيذ تعليمات البرنامج الأصلي.

يُبيّن الشكل 10-1 تصوراً عاماً جداً لتركيب الفيروس (مأخوذاً من [COHE94]). في هذه الحالة تم تضمين تعليمات الفيروس V في بداية برامج مصابة، وافترض أن نقطة الدخول للبرنامج عنداستدعائه هي السطر الأول من البرنامج.

```
program V:=
{    goto main;
     1234567;

     subroutine infect-executable :=
     {loop:
          file := get-random-executable-file;
          if(first-line-of-file = 1234567) then
               goto loop;
          else
               prepend V to file;
     }

     subroutine do-damage :=
     {whatever damage is to be done}

     subroutine trigger-pulled :=
     {return true if some condition holds}

     main:  main-program :=
     {
          infect-executable;
          if trigger-pulled then do-damage;
          goto  next;
     }

     next:
}
```

الشكل 10-1: فيروس بسيط.

يبدأ البرنامج المصاب بتعليمات الفيروس ويعمل على النحو الآتي: تسبب أول تعليمة القفز إلى برنامج الفيروس الأساسي. التعليمة الثانية هي علامة خاصة تُستخدَم من قِبَل الفيروس لتحديد ما إذا كان برنامج الضحية المحتمل مصاباً بهذا الفيروس بالفعل أم لا. وعند استدعاء البرنامج، ينتقل التحكم فوراً إلى برنامج الفيروس الأساسي. يفتش برنامج الفيروس أولاً عن ملفات قابلة للتنفيذ غير مصابة ويصيبها. بعد ذلك قد يقوم الفيروس بعملٍ ما، عادةً ما يكون ضاراً بالنظام. يمكن أن يُؤدَّي هذا العمل في كل مرة يُستدعى فيها البرنامج، أو يمكن أن يكون قنبلة منطقية تثار فقط في ظروف معينة. وأخيراً، يُحوِّل الفيروس التحكم إلى البرنامج الأصلي. وإذا كانت مرحلة العدوى سريعة إلى حدٍّ معقول، فمن غير المحتمل أن يلاحظ المستخدم أي اختلاف بين تنفيذ البرنامج المصاب وغير المصاب.

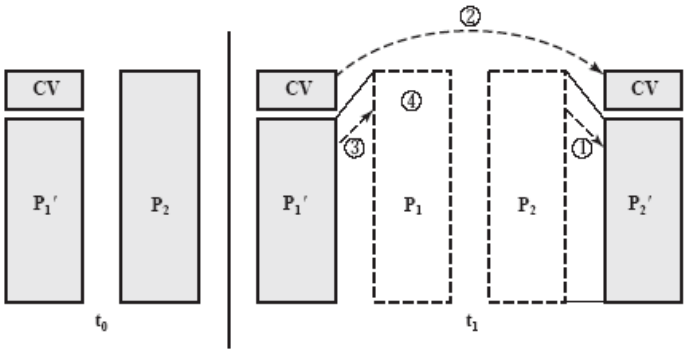
```
program CV:=
{    goto main;
     01234567;

     subroutine infect-executable :=
     {loop:
          file := get-random-executable-file;
          if(first-line-of-file = 01234567) then goto loop;
           (1)  compress file;
           (2)  prepend V to file;
     }

     main:  main-program :=
     {
          if ask-permission then infect-executable;
           (3)  uncompress rest-of-file;
           (4)  run uncompressed file;
     }
}
```

الشكل 10-2: منطق فيروس يقوم بضغط البرنامج الأصلي.

من السهل اكتشاف فيروس كالذي وصفناه للتو لأن حجم النسخة المصابة من البرنامج يكون أكبر من حجم النسخة المناظرة غير المصابة. ومن طرق إحباط مثل تلك الوسيلة البسيطة لاكتشاف الفيروس أن يقوم الفيروس بضغط الملف القابل للتنفيذ بحيث يتساوى حجم كلتا النسختين (المصابة وغير المصابة). يوضح الشكل 10-2 [COHE94] المنطق المطلوب بشكل عام؛ حيث تم ترقيم التعليمات المهمة في هذا الفيروس. يوضح الشكل 10-3 [COHE94] طريقة العمل.



الشكل 10-3: فيروس يقوم بضغط البرنامج الأصلي.

افترض أن البرنامج P_1 مصاب بالفيروس CV. عندما يُطلب تنفيذ ذلك البرنامج، ينتقل التحكم إلى الفيروس، والذي يقوم بالخطوات الآتية:

1. لكل ملف غير مصاب P_2 موجود، يضغط الفيروس أولاً ذلك الملف ليُنتج P'_2 بطول أقصر من البرنامج الأصلي بمقدار حجم الفيروس.
2. تُضاف نسخة من الفيروس في بداية البرنامج المضغوط.
3. يُزال ضغط النسخة المضغوطة P'_1 للبرنامج المصاب الأصلي.
4. يُنفَّذ البرنامج الأصلي غير المضغوط.

في هذا المثال، لا يعمل الفيروس شيئاً أكثر من الانتشار (العدوى). كما في المثال السابق، قد يتضمَّن الفيروس قنبلة منطقية.

❖ **العدوى الأولى:**

بمجرد أن يتمكن الفيروس من الدخول على النظام بإصابة برنامج واحد، يصبح قادراً على إصابة بعض الملفات الأخرى القابلة للتنفيذ أو كلها على ذلك النظام عند تنفيذ البرنامج المصاب. ومن ثَمَّ يمكن منع العدوى الفيروسية بالكامل بمنع الفيروس من الدخول في المقام الأول. ولسوء الحظ، عملية المنع تلك صعبة جداً لأن الفيروس يمكن أن يكون جزءاً من أي برنامج خارج النظام. ولذلك ما لم يكن الشخص مقتنعاً بأخذ قطعة خالية من الحديد ثم يكتب عليها كل ما يخصه من برامج النظام والتطبيقات، فسيكون في موقف ضعف.

10-1-3 أنواع الفيروسات

منذ ظهور الفيروسات والصراع مستمرٌ بين كُتَّاب برامج الفيروسات وكُتَّاب البرامج المضادة للفيروسات. وكلما طُوِّرت إجراءات مضادة فعَّالة لأنواع الفيروسات الموجودة، طُوِّرت أنواع جديدة أخرى من الفيروسات. واقترح [STEP93] الفئات الآتية لتمثيل أشد أنواع الفيروسات خطورة:

- الفيروس الطفيلي (parasitic virus): وهو النوع التقليدي الذي لا يزال يمثل الشكل الأكثر شيوعاً من الفيروسات. يربط الفيروس الطفيلي نفسه بالملفات القابلة للتنفيذ، ويتضاعف عندما يُنفَّذ البرنامج المصاب بالعثور على الملفات الأخرى القابلة للتنفيذ وإصابتها.
- الفيروس المقيم بالذاكرة (memory-resident virus): يسكن في الذاكرة الرئيسة كجزء من برنامج مقيم من برامج نظام التشغيل. ومن تلك اللحظة فصاعداً، يُصيب الفيروس كل البرامج التي تُنفَّذ.

- فيروس قطاع البدء (boot sector virus): يصيب سجل بدء رئيس (master boot record) أو سجل بدء (boot record) وينتشر عند بدأ تشغيل النظام من قرص يحتوي على الفيروس.
- الفيروس المُتخفي (stealth virus): شكل من أشكال الفيروسات مصمّم بطريقة تُمكّنه من إخفاء نفسه بحيث لا تتمكّن البرامج المضادة للفيروسات من اكتشافه.
- الفيروس متعدد الأشكال (polymorphic virus): فيروس يُغيّر شكله مع كل عدوى، مما يجعل عملية اكتشافه عن طريق "بصمته" مستحيلة.
- الفيروس المَسْخي (المتغير المظهر) (metamorphic virus): كما هو الحال مع الفيروس متعدد الأشكال، يتغيّر الفيروس المسخي في كل عدوى. الفرق بينهما هو أن الفيروس المسخي يعيد كتابة نفسه بالكامل في كل تكرار، مما يزيد من صعوبة اكتشافه. قد تُغيِّر الفيروسات المسخية من سلوكها بالإضافة إلى مظهرها.

تم مناقشة أحد الأمثلة للفيروسات المتخفية في وقت سابق: الفيروس الذي يستخدم الضغط (compression) لكي يكون حجم البرنامج المصاب هو نفسه بالضبط قبل الإصابة. هناك طرق ممكنة تُعدُّ أكثر تطوراً بكثير من ذلك. على سبيل المثال، يمكن أن يضع الفيروس "منطقاً اعتراضياً" (intercept logic) في برامج الدخل والخرج للأقراص (disk I/O routines)، بحيث إنه عند محاولة المستخدم قراءة الأجزاء المشكوك فيها بالقرص لاكتشاف الفيروس يقوم الفيروس بتقديم البرنامج الأصلي غير المصاب. وهكذا، فإن مصطلح "التخفي" ليس تعبيراً ينطبق على الفيروس في حد ذاته لكنه بالأحرى طريقة يستخدمها الفيروس في المراوغة وتجنب الاكتشاف.

أثناء عملية الانتشار، يُنشئ الفيروس متعدد الأشكال نسخاً متكافئة من حيث الوظيفة ولكن لها أنماط بتات مختلفة. كما هو الحال مع الفيروس المتخفي، الهدف من ذلك هو هزيمة البرامج التي تُستخدم لكشف الفيروسات. وفي هذه الحالة، يتغير "توقيع" الفيروس مع كل نسخة. ولتحقيق هذا الاختلاف، قد يُدخِل الفيروس تعليمات زائدة عن الحاجة بشكل عشوائي أو يبدل ترتيب التعليمات المستقلة بالبرنامج. من الطرق أخرى الأكثر فعالية استخدام التشفير. يُنشئ جزء من الفيروس، يطلق عليه بشكلٍ عام "مُحرِّك التغيير" (mutation engine)، مفتاح تشفير عشوائي لتشفير الجزء المتبقي من الفيروس. يُخزَّن المفتاح مع الفيروس، ويتم تعديل مُحرِّك التغيير نفسه. عندما يُطلب تنفيذ برنامج مصاب، يستخدم الفيروس المفتاح العشوائي المُخزَّن لإزالة تشفير الفيروس. وعندما يتكاثر الفيروس يختار مفتاحاً عشوائياً آخر.

من الأسلحة الأخرى في ترسانة كُتّاب الفيروسات استخدام حزمة برمجيات (toolkit) لإنشاء الفيروسات بشكل آلي. تُمكِّن مثل تلك البرمجيات الشخص المبتدئ نسبياً من إنشاء عدد من الفيروسات المختلفة بسرعة. ورغم أن الفيروسات التي تنتجها تلك البرمجيات تكون عادةً أقل تطوراً من الفيروسات الأخرى التي صُمِّمت من البداية بشكل مستقل، فإن العدد الهائل من الفيروسات الجديدة التي يمكن إنتاجها بهذه الطريقة يُمثّل مشكلة تواجه الأساليب المضادة للفيروسات.

10-1-4 فيروسات الماكرو (Macro Viruses)

في منتصف التسعينيات، أصبحت فيروسات الماكرو إلى حدٍّ كبير أكثر أنواع الفيروسات انتشاراً. تُشكل تلك الفيروسات على وجه الخصوص تهديداً أمنياً كبيراً لعدة أسباب:

- لا يعتمد فيروس الماكرو على منصَّة (platform) الأجهزة المستخدمة. فمثلاً كل فيروسات الماكرو تقريباً تصيب مستندات برنامج

مايكروسوفت لمعالجة الكلمات (Microsoft word). ومن ثَمَّ فإن أي عتاد أجهزة أو نظام تشغيل يدعم هذا البرنامج يمكن إصابته بتلك الفيروسات.

- تصيب فيروسات الماكرو المستندات، وليس الأجزاء القابلة للتنفيذ من البرامج. لاحظ أن أغلب المعلومات على نظام الحاسب توجد على شكل مستندات وليس على شكل برامج.
- تنتشر فيروسات الماكرو بسهولة. يمثل البريد الإلكتروني طريقة شائعة جداً لتحقيق ذلك.

تستغل فيروسات الماكرو خاصية موجودة ببرنامج معالجة الكلمات (word) وغيره من تطبيقات المكتب الأخرى كبرنامج مايكروسوفت إكسل للجداول الإلكترونية، ألا وهي الماكرو (macro). بشكلٍ أساسي، الماكرو هو برنامج قابل للتنفيذ مُضمَّن في مستند معالجة الكلمات أو نوع آخر من الملفات. عادةً ما يُستخدم الماكرو لأتمتة المهام المتكررة وذلك لتقليل حاجة مستخدمي التطبيق إلى الضغط على المفاتيح. وفي العادة تكون لغة البرمجة للماكرو نوعاً من لغة البيسك (BASIC). ويمكن أن يُعرِّف المستخدم سلسلة من ضربات المفاتيح في ماكرو ويُعدِّها بحيث يُستدعى الماكرو عند الضغط على أحد مفاتيح الوظائف (function keys) أو على مجموعة خاصة مختصرة من المفاتيح.

توفر الإصدارات المتعاقبة من برنامج معالجة الكلمات حمايةً متزايدةً ضد فيروسات الماكرو. فعلى سبيل المثال، توفر مايكروسوفت أداة اختيارية للحماية من فيروسات الماكرو تساعد على اكتشاف ملفات برنامج word المريبة، وتقوم بلفت نظر المستخدم إلى الخطر المحتمل عند فتح ملف يتضمن ماكرو. وطوَّرت أيضاً العديد من شركات البرامج المضادة للفيروسات منتجات للكشف عن فيروسات الماكرو وتصحيحها. كما هو الحال مع أنواع الفيروسات الأخرى، يستمر السباق في حقل فيروسات الماكرو، غير أنها لم تعد تمثل تهديد الفيروسات السائد.

5-1-10 فيروسات البريد الإلكتروني

يُعدُّ فيروس البريد الإلكتروني تطوراً أحدث في البرمجيات الخبيثة. استَخدَمت فيروسات البريد الإلكتروني الأولى التي انتشرت بسرعة، كفيروس ميليسا (Melissa)، ماكرو مُتضمِّناً في مستندات word الملحقة برسالة البريد. إذا فتح المُستلِم ملحق البريد الإلكتروني، فإنه يتم تنشيط الماكرو، ومن ثم:

1. يُرسل فيروس البريد الإلكتروني نفسه إلى كل شخص على قائمة العناوين في حزمة البريد الإلكتروني الخاصة بالمستخدم.
2. يتسبب الفيروس في أضرار محلية.

في نهاية عام 1999 ظهرت نوعيّة أكثر قوة من فيروس البريد الإلكتروني، حيث يمكن تنشيط تلك النوعية الأحدث بمجرد فتح رسالة البريد الإلكتروني التي تحتوي الفيروس دون الحاجة إلى فتح الملحق. يستخدم الفيروس لغة برمجة البيسك المرئية (Visual Basic) والمدعومة من قِبَل حزمة البريد الإلكتروني.

وهكذا نرى جيلاً جديداً من البرمجيات الخبيثة التي تصل عن طريق البريد الإلكتروني وتستخدم خصائص تطبيق البريد الإلكتروني للانتشار (مضاعفة نفسها) عبر الإنترنت. ينشر الفيروس نفسه بمجرد تنشيطه (بفتح البريد الإلكتروني نفسه أو بفتح أحد ملحقاته)، وينتقل لكل عناوين البريد الإلكتروني المعروفة لدى المضيف المصاب. ونتيجة لذلك أصبحت الفيروسات تنتشر الآن في غضون ساعات، بعد أن كان ذلك يستغرق شهوراً وسنين في السابق. لذا أصبح من الصعب جداً على البرامج المضادة للفيروسات الاستجابة لخطر تلك الفيروسات لتلافي حدوث ضرر كبير. وفي النهاية، ينبغي تزويد برامج الخدمات المرافقة للإنترنت وبرامج التطبيقات على الحاسب الشخصي بدرجة أكبر من الأمن لمواجهة ذلك التهديد المتزايد.

6-1-10 الديدان

الدودة هي برنامج يمكن أن يضاعف نفسه ويرسل النسخ من حاسب لآخر عبر توصيلات الشبكة. وعند وصولها، قد تنشط الدودة وتقوم بالتكاثر والانتشار مرة أخرى. بالإضافة إلى الانتشار، تؤدّي الدودة عادةً بعض الأعمال غير المرغوبة. لفيروس البريد الإلكتروني بعض خصائص الديدان، فهو ينتقل من نظام إلى آخر. ومع ذلك ما زلنا نصنّفه كفيروس لأنه يتطلب شخصاً لدفعه إلى الأمام. تنشط الدودة في البحث عن أجهزة أخرى لإصابتها، ويُستخدم كل جهازٍ مصابٍ بدوره كمنصَّة لشن هجمات على أجهزة أخرى.

تستخدم برامج ديدان الشبكة توصيلات الشبكة للانتشار من نظام لآخر. وبمجرد أن تنشط داخل نظامٍ ما، يمكن أن تتصرف دودة الشبكة كفيروس أو بكتيريا، كما يمكن أن تزرع أحصنة طروادة، أو أن تؤدي أي عددٍ من الأعمال المعرقلة أو التدميرية. ولمضاعفة نفسها تستخدم دودة الشبكة أحد وسائل الاتصال المتاحة عبر الشبكة؛ ومن أمثلة ذلك:

- البريد الإلكتروني: تُرسل الدودة نسخة من نفسها كرسالة بريد إلى الأنظمة الأخرى.
- إمكانية التنفيذ عن بُعد (remote execution): تُنفِّذ الدودة نسخة من نفسها على نظام آخر.
- إمكانية الدخول على الأنظمة عن بُعد (remote login): تُسجِّل الدودة للدخول على نظام بعيد كمُستخدِم ثم تصدر تعليمات لنسخ نفسها من نظامٍ إلى آخر.

تعمل النسخة الجديدة من برنامج الدودة على النظام البعيد حيث تقوم، بالإضافة إلى أي وظائف تؤديها على ذلك النظام، بمواصلة نشر نفسها بنفس الطريقة.

لدودة الشبكة نفس خصائص فيروس الحاسب: مرحلة الخمول، ومرحلة الانتشار، ومرحلة الإثارة، ومرحلة التنفيذ. بشكل عام، تتضمن مرحلة الانتشار الوظائف الآتية:

1. البحث عن أنظمة أخرى لإصابتها بفحص جداول المضيف (أو مستودعات مماثلة) التي تتضمن عناوين أنظمة بعيدة.
2. تنشئ اتصالاً بالنظام البعيد.
3. تنسخ نفسها للنظام البعيد وتتسبب في تشغيل تلك النسخة على ذلك النظام.

قد تحاول دودة الشبكة أيضاً تحديد ما إذا كان النظام قد سبق أن أصيب أم لا. في أنظمة البرمجة المتعدّدة، قد تقوم الدودة أيضاً بإخفاء تواجدها بتسمية نفسها كعملية نظام أو باسم آخر لا يمكن لمُشغِّل النظام ملاحظته. كما هو الحال مع الفيروسات، يصعب مواجهة ديدان الشبكة.

❖ دودة موريس:

حتى وقت ظهور الجيل الحالي من الديدان، كانت الدودة الأكثر شهرة هي تلك التي قام روبرت موريس بإطلاقها إلى الإنترنت في عام 1998. كانت دودة موريس مُصمَّمة للانتشار على أنظمة يونيكس واستخدمت عدداً من الطرق المختلفة للانتشار. وعندما تبدأ نسخة في التنفيذ، تكون مهمتها الأولى اكتشاف المضيفات الأخرى المعروفة لهذا المضيف التي تسمح بالدخول عليها من ذلك المضيف. تقوم الدودة بذلك عن طريق فحص تشكيلة من القوائم والجداول، بما في ذلك جداول النظام التي تحدد الأجهزة الأخرى التي تُعتبر محل ثقة لدى ذلك المضيف، وملفات تمرير البريد الإلكتروني، والجداول التي من خلالها يُعطَى المستخدمون صلاحيات الوصول إلى الحسابات عن بُعد، ومن ثم إنشاء برنامج يقوم بالإبلاغ عن حالة توصيلات الشبكة.

لكل مضيف يتم اكتشافه، تُجرِّب الدودة عدة طرقٍ مختلفةٍ لمحاولة الدخول:

1. تحاول التسجيل للدخول على المضيف البعيد كمستخدم شرعي. في هذه الطريقة تحاول الدودة في البداية اختراق ملف كلمة السر المحلي، وبعد ذلك تستخدم الأزواج المكتشفة لكلمات السر وهويات المستخدمين المناظرة. وتكمن الفرضية في أن عدداً كبيراً من المستخدمين يستخدمون نفس كلمة السر على نظم مختلفة. وللحصول على كلمات السر، تُشغِّل الدودة برنامجاً لكسر كلمات السر؛ يجرب البرنامج:
   a. كل اسم حساب للمستخدم والتبديلات البسيطة منه،
   b. قائمة من 432 كلمة سر مبيّتة (built-in) داخل دودة موريس اعتقد موريس أنها محتملة بدرجة كبيرة.
   c. كل الكلمات في دليل النظام المحلي.
2. تستغل خطأً برمجياً في بروتوكول finger للإبلاغ عن مكان المستخدم البعيد.
3. تستغل مدخلاً خلفياً في خيار التنقيح للعملية البعيدة التي تستلم وترسل البريد.

إذا نجحت أيٌّ من تلك الهجمات، تكون الدودة قد تمكنت من تحقيق اتصال بمُفسِّر الأوامر (command interpreter) لنظام التشغيل. ومن ثَمَّ تُرسل لهذا المُفسِّر برنامج بدء ذاتي (bootstrap) قصير، وتُصدر أمراً لتنفيذ ذلك البرنامج، وبعدها تُسجِّل خروجاً من النظام. بعد ذلك يستدعى برنامج البدء الذاتي البرنامج الأصلي ويُنزّل بقيّة الدودة، وبعدها تُنفَّذ الدودة الجديدة.

❖ هجمات الديدان المعاصرة:

بدأ العصر الحديث لتهديدات الدودة مع إطلاق دودة الكود الأحمر (Red Code) في يوليو/تموز من عام 2001. تستغل دودة الكود الأحمر ثغرةً أمنيةً في خادم معلومات الإنترنت ((Internet Information Server (IIS) من مايكروسوفت لاختراق النظام ومن ثم الانتشار من خلاله. تقوم الدودة أيضاً بتعطيل مدقّق ملفات النظام (system file checker) في نظام ويندوز. وتتلمّس الدودة كذلك عناوين IP عشوائية للانتشار إلى مضيفات أخرى. وخلال فترة زمنية معينة، تقتصر الدودة على الانتشار فقط. بعد ذلك تبدأ بشن هجوم حجب الخدمة ضد موقع ويب حكومي، وذلك بإرسال فيضان من الرزم إلى الموقع من عدة مضيفات. ثم تُعلِّق الدودة أنشطتها لتعاود نشاطها من جديد بعد ذلك بشكل دوري. في موجة الهجوم الثانية، أصابت دودة الكود الأحمر ما يقرب من 360 ألف خادم في غضون 14 ساعة. بالإضافة إلى الدمار الذي سببته في الخادمات المستهدفة، يمكن أن تستهلك الدودة حيزاً كبيراً من سعة الإرسال على الإنترنت مما يعرقل الخدمة.

كان الإصدار الثاني للدودة، الكود الأحمر 2 (Code Red II)، نسخة جديدة تستهدف خادمات مايكروسوفت للويب (IIS). إضافةً إلى ذلك، تُثبّت هذه الدودة الحديثة مدخلاً خلفياً يسمح لمهاجمي الحاسبات بتوجيه أنشطة الحاسبات الضحيّة. وفي أواخر عام 2001 ظهرت دودة أكثر مرونة، عُرِفت باسم نيمدا (Nimeda). انتشرت نيمدا بآليات متعدّدة:

- من زبون لآخر عن طريق البريد الإلكتروني.
- من زبون لآخر عن طريق موارد الشبكة المشتركة المفتوحة.
- من خادم ويب إلى زبون عن طريق تصفح مواقع الويب المصابة.

- من زبون إلى خادم ويب عن طريق المسح النشط واستغلال نقاط الضعف المختلفة المرتبطة باستعراض دليل خادم مايكروسوفت للويب الإصدار 4.0 و5.0.
- من زبون إلى خادم ويب عن طريق المسح بحثاً عن المداخل الخلفية التي تنشئها ديدان "الكود الأحمر 2".

تُعدِّل الدودة مستندات الويب (كملفات htm، html، asp) وملفات معينة قابلة للتنفيذ موجودة على الأنظمة التي تصيبها، وتُنشئ نسخاً عديدة من نفسها بأسماء ملفات مختلفة. وفي أوائل عام 2003 ظهرت دودة إس كيو إل سلامار (SQL Slammer). استغلت تلك الدودة نقطة الضعف المتعلقة بالفيضان (overflow) في خادم مايكروسوفت لقواعد بيانات SQL. وكانت سلامار مختصرة جداً وانتشرت بسرعة، مما تسبب في إصابة 90% من المضيفات الضعيفة خلال 10 دقائق. وشهدت أواخر عام 2003 ظهور دودة سوبيج.إف (Sobig.f) التي استغلت خادمات البروكسي المفتوحة لتحويل الأجهزة المصابة إلى محركات رسائل الدعاية (spam engines). وفي ذروة نشاطها كانت دودة سوبيج.إف (Sobig.f) مسؤولة عن توليد رسالة واحدة من بين كل 17 رسالة وأنتجت أكثر من مليون نسخة من نفسها خلال الساعات الأربع والعشرين الأولى من إطلاقها.

دودة مايدوم (Mydoom) هي دودة لرسائل البريد الإلكتروني تُرسَل لعدد كبير من المستخدمين، وظهرت في عام 2004. واتّبع مصممو تلك الدودة الاتجاه المتزايد لتثبيت مداخل خلفية في الحاسبات المصابة، مما يُمكِّن مهاجمي الحاسبات من الوصول لبيانات مهمة عن بُعد، ككلمات السر وأرقام بطاقات الائتمان. تكاثرت دودة "مايدوم" بمعدل ألف مرة في الدقيقة وأغرقت الإنترنت بحوالي 100 مليون رسالة مصابة في 36 ساعة.

7-1-10 الوضع الحالي لتقنية الديدان

يشمل الوضع الحالي لتقنية الديدان ما يأتي:

- تعدد المنصَّات (multiplatform): لا تقتصر الديدان الحديثة على إصابة أجهزة الحاسب التي تعمل بنظام ويندوز فقط، بل يمكن أن تهاجم منصَّات مختلفة، خاصةً تشكيلة أنظمة يونيكس شائعة الاستخدام.
- تعدد الاستغلال (multiexploit): تخترق الديدان الجديدة الأنظمة بتشكيلة متنوعة من الطرق باستخدام طرق مختلفة من الاستغلال (exploits) ضد نقاط الضعف في خادمات الويب، والمتصفحات، والبريد الإلكتروني، والملفات المشتركة، وتطبيقات الشبكة الأخرى.
- الانتشار السريع جداً (ultrafast spreading): واحد من الطرق المتّبعة لتسريع انتشار الدودة، حيث يتم إجراء مسح مُسبَق للإنترنت لتجميع عناوين الإنترنت للأجهزة الضعيفة لاستهدافها.
- تعدد الأشكال (polymorphic): لتجنب الاكتشاف والمرور عبر المرشحات وإحباط محاولات التحليل الفوري (الوقت الحقيقي) (real-time analysis)، تتبع الديدان أسلوب تعدد الأشكال الذي تستخدمه الفيروسات. حيث يكون لكل نسخة من الدودة تعليمات جديدة تنشئ في التوّ باستخدام تعليمات وأساليب تشفير متكافئة وظيفياً.
- الديدان المسخية (متعددة المظهر) (metamorphic): بالإضافة إلى تغيير مظهرها، للديدان المسخية عدة أنماط من السلوك التي يتم إطلاقها في المراحل المختلفة من عملية الانتشار.
- الديدان كوسائل نقل (transport vehicles): نظراً لأن الديدان يمكن أن تصيب عدداً كبيراً من الأنظمة بسرعة، فإنها تُعدُّ طريقة مثالية

لنشر أدوات أخرى للهجمات الموزَّعة، كهجوم زومبي الموزَّع لحجب الخدمة.

- استغلال ساعة الصفر (zero-day exploit): لتحقيق أكبر قدرٍ من المفاجأة وأقصى توزيع ممكن، يجب أن تستغل الدودة نقطة الضعف المجهولة التي اكتشفها العاملون في مجال الشبكات العامة فقط عند إطلاق الدودة.

10-2 الإجراءات المضادة للفيروسات

10-2-1 الأساليب المضادة للفيروسات

الحل المثالي لتهديدات الفيروسات هو الوقاية منها؛ أي عدم السماح للفيروس بالدخول للنظام في المقام الأول. تحقيق هذا الهدف عموماً مستحيل، رغم أن المنع يمكن أن يُخفّض عدد الهجمات الفيروسية الناجحة. الطريقة الأفضل بعد الوقاية أن تكون قادراً على عمل الآتي:

- الكشف (detection): عندما تحدث العدوى، يتضمن الكشف التأكد من أن العدوى حدثت وتحديد موضع الفيروس.
- تحديد الهوية (identification): بمجرد إنجاز الاكتشاف، يلزم التعرف على نوع الفيروس الذي أصاب البرنامج.
- الإزالة (removal): بعد تحديد الفيروس، يلزم إزالة كل آثاره من البرنامج المصاب وإعادة البرنامج لحالته الأصلية. ينبغي إزالة الفيروس من كل الأنظمة المصابة لإيقاف انتشار العدوى.

إذا نجح الكشف عن الفيروس لكن لم يتم التعرف عليه أو لم يمكن إزالته، فإن البديل هو استبعاد البرنامج المصاب وإعادة تحميل نسخة نظيفة من النسخة الاحتياطية.

تسير التطورات في تقنية الفيروسات وتقنية البرامج المضادة لها يداً بيد. كانت الفيروسات الأولى بسيطة نسبياً، ويمكن تمييزها والتخلص منها بحِزَم برامج بسيطة مضادة للفيروسات. ومع استمرار تطور سباق التسلح ضد الفيروسات، ازدادت الفيروسات، وبالطبع البرامج المضادة لها، تعقيداً وتطوراً. حدد [STEP93] أربعة أجيال للبرامج المضادة للفيروسات:

- الجيل الأول: برامج مسح (scanners) بسيطة.
- الجيل الثاني: برامج مسح اجتهادية (heuristic scanners).
- الجيل الثالث: مصائد أنشطة (activity traps).
- الجيل الرابع: الحماية الكاملة.

يتطلب برنامج المسح من الجيل الأول معرفة توقيع (بصمة) الفيروس ليتمكن من تمييزه. قد يتضمن الفيروس رموزاً عامة (wildcards) لكنه أساساً له نفس التركيب ونمط البتات في كل نسخة. وتصلح مثل تلك البرامج المرتبطة بتوقيعات معينة للكشف عن الفيروسات المعروفة فقط. ويحتفظ نوع آخر من برامج الجيل الأول بسجل لأطوال البرامج ويبحث عن التغييرات في الطول.

أما برامج الجيل الثاني فلا تعتمد على توقيعات معينة، وإنما تستخدم قواعد اجتهادية (heuristic rules) للبحث عن عدوى الفيروس المحتملة. ويبحث أحد أنواع ذلك الصنف من البرامج عن أجزاء الكود (code fragments) المرتبطة غالباً بالفيروسات. فعلى سبيل المثال قد يبحث البرنامج عن بداية حلقة تشفير (encryption loop) تُستخدم مع الفيروسات متعددة الأشكال ويكتشف مفتاح التشفير. عندما يكتشف المفتاح، يمكن أن يُزيل برنامج الماسح التشفير للتعرّف على الفيروس، ثمّ يزيل العدوى ويعيد البرنامج للعمل.

من الطرق الأخرى التي يستخدمها الجيل الثاني فحص السلامة (integrity checking). يمكن أن يُذيّل كل برنامج بمجموع تدقيقي (checksum). إذا

أصاب الفيروس البرنامج بدون تغيير قيمة المجموع التدقيقي الملحقة بالبرنامج، فسيتمكن فحص السلامة من اكتشاف الفيروس. لمواجهة فيروس متطوّر بما فيه الكفاية لتغيير القيمة المُخزَّنة للمجموع التدقيقي عندما يصيب البرنامج، يمكن استخدام كود التحوير المُشفَّرة (encrypted hash function). يُخزّن مفتاح التشفير بشكلٍ منفصل عن البرنامج حتى لا يتمكن الفيروس من إنشاء كود تحوير جديد ويقوم بتشفيره. باستخدام كود التحوير بدلاً من المجموع التدقيقي البسيط، سيُمنع الفيروس من تعديل البرنامج ليعطي نفس كود التحوير كما في السابق.

برامج الجيل الثالث برامج مقيمة بالذاكرة تتعرّف على الفيروس من سلوكه بدلاً من تركيبه في برنامج مصاب. تمتاز تلك البرامج بأنها لا تحتاج لتطوير توقيعات وقواعد اجتهادية لعدد كبير من الفيروسات، وإنما من الضروري فقط تمييز مجموعة صغيرة من الأفعال التي تصاحب محاولة الإصابة بالعدوى ومن ثم التدخل.

منتجات الجيل الرابع هي حزم برمجيات تشمل تشكيلة من الأساليب المضادة للفيروسات والتي تُستخدم معاً، بما في ذلك المسح ومصائد الأنشطة. بالإضافة إلى ذلك، تتضمّن الحزمة إمكانية التحكم في الوصول التي تُحِد من قدرة الفيروسات على اختراق النظام ومن ثَمَّ تُحِد من قدرتها على تعديل الملفات لنقل العدوى.

يستمر سباق التسلح ضد الفيروسات. وتستخدم حزم الجيل الرابع استراتيجيةً أكثر شمولية للدفاع، مما سيوسّع مجال الدفاع ليشمل إجراءات أكثر عمومية لأمن الحاسب.

10-2-2 الأساليب المتقدمة المضادة للفيروسات

يستمر ظهور أساليب ومنتجات أكثر تطوّراً لمكافحة للفيروسات. سنُلقي هنا الضوء على أسلوبين مهمين منها.

❖ **طريقة إزالة التشفير العامة:**

تُمكِّن طريقة إزالة التشفير العامة ((Generic decryption (GD) البرامج المضادة للفيروسات من اكتشاف الفيروسات بسهولة، حتى الفيروسات المتطورة متعددة الأشكال، مع الاحتفاظ بسرعة مسح عالية [NACH97]. تذكَّر أنه عند تنفيذ ملف يحتوي على فيروس متعدد الأشكال، يجب أن يُزيل الفيروس تشفير نفسه لينشط. لكي يُكتشَف مثل هذا التركيب، يجب تشغيل الملفات القابلة للتنفيذ من خلال ماسح GD يتضمن العناصر الآتية:

- مُحاكٍ لوحدة المعالجة المركزية: وهو برنامج يمثل حاسباً افتراضياً. تفسّر التعليمات الموجودة في ملف تنفيذي بالمُحاكي بدلاً من تنفيذها على المعالج التحتي. ويتضمّن المُحاكي نُسخاً برمجية من كل المسجِّلات (registers) والمكونات المادية الأخرى للمعالج حتى لا يتأثر المعالج التحتي بالبرامج التي تفسر بواسطة المُحاكي.
- ماسح لتوقيعات الفيروسات: وحدة لمسح الكود المستهدف بحثاً عن توقيعات الفيروسات المعروفة.
- وحدة تحكم في المحاكاة: وتتحكم في تنفيذ الكود المستهدف.

في بداية كل محاكاة، يبدأ المُحاكي بترجمة التعليمات في الكود المستهدف، الواحدة تلو الأخرى. ومن ثَمَّ إذا تضمَّن الكود إجراءً لإزالة التشفير وكشف الفيروس، فإن ذلك الكود سيُترجم. فعلياً، يقوم الفيروس بالعمل لحساب البرنامج المضاد للفيروسات وذلك بالإعلان عن الفيروس وكشفه.

بشكل دوري، تقوم وحدة التحكم بمقاطعة عملية التفسير لمسح الكود المستهدف بحثاً عن توقيعات الفيروسات.

لا يمكن أن تُلحِق عملية تفسير الكود المستهدف أي ضرر ببيئة الحاسب الشخصي الفعلية، حيث تتم الترجمة بالكامل في بيئة تحت السيطرة.

يُعدُّ تحديد مدة التشغيل لكل تفسير أصعب قضايا تصميم ماسح GD. عادةً ما تُنشَّط عناصر الفيروس بعد بدء تنفيذ البرنامج مباشرةً، ولكن ليس من الضروري أن يكون الأمر كذلك. كلما طالت الفترة التي يقوم فيها الماسح بمحاكاة برنامج معين، زاد احتمال اكتشاف أي فيروسات مخفية. ومع ذلك، يمكن أن يستغرق البرنامج المضاد للفيروسات مدة محدودة ويستهلك موارد محدودة قبل أن يشتكي المستخدمون.

❖ نظام المناعة الرقمي (Digital Immune System):

نظام المناعة الرقمي هو طريقة شاملة للحماية ضد الفيروسات قامت بتطويره شركة IBM [KEPH97a, KEPH97b]، وذلك لمواجهة خطر التهديد المتصاعد لانتشار الفيروسات من خلال الإنترنت. في البداية سنلقي بعض الضوء على هذا التهديد، ثم سنُلخِّص طريقة IBM.

بشكلٍ تقليدي، كانت تهديدات الفيروسات تتسم بالانتشار البطيء نسبياً للفيروسات الجديدة والتغيرات الجديدة. فعادةً ما كان يتم تحديث البرامج المضادة للفيروسات بشكل شهري، وكان هذا كافياً للسيطرة على المشكلة. أيضاً بشكل تقليدي، أدّت الإنترنت دوراً صغيراً نسبياً في انتشار الفيروسات. لكن كما بيَّن [CHES97] هناك اتجاهان رئيسان في تقنية الإنترنت كان لهما تأثيرٌ متزايدٌ على معدل انتشار الفيروسات في السنوات الأخيرة:

- أنظمة البريد المدمجة: جعلت أنظمة البريد مثل Notes من Lotus وOutlook من مايكروسوفت إرسال أي شيء لأي أحد، وكذلك معالجة الكائنات التي تُستلم، أمراً بسيطاً للغاية.
- أنظمة البرامج النقَّالة: سمحت التقنيات الحديثة مثل Java وActiveX للبرامج بالانتقال تلقائياً من نظام إلى آخر.

لمواجهة التهديدات الناجمة عن تلك الإمكانات المبنية على الإنترنت، طورت شركة IBM نموذجاً لنظام المناعة الرقمي. يمثل هذا النظام امتداداً لاستخدام المحاكاة وقد نوقش في الجزء الفرعي السابق، ويوفّر نظاماً متعدد الأغراض للمحاكاة وكشف الفيروسات. يهدف النظام إلى تسريع الاستجابة بحيث يتم إخماد الفيروسات بمجرد ظهورها تقريباً. وعندما يدخل فيروس جديد إلى مؤسسةٍ ما، يقوم نظام المناعة آلياً بأسره، وتحليّله، بالإضافة إلى كشفه والحماية منه، ثم إزالته، وتمرير المعلومات الخاصة به إلى الأنظمة التي تستخدم برامج IBM المضادة للفيروسات لاكتشافه قبل أن تتاح له فرصة إتلاف مكان آخر.

يوضح الشكل 10-4 الخطوات المعتادة لطريقة عمل نظام المناعة الرقمي:

1. يستخدم برنامج المراقبة على كل جهاز حاسب شخصي تشكيلة من القواعد الاجتهادية للكشف عن وجود الفيروس بناءً على سلوك النظام، أو التغييرات المريبة في البرامج، أو التوقيع المميِّز لسُلالة الفيروس. يُرسِل برنامج المراقبة نسخة من أي برنامج يعتقد أنه مصاب إلى جهاز حاسب إداري بالمؤسسة.
2. يُشفِّر جهاز الحاسب الإداري العيّنة ويرسلها إلى جهاز حاسب مركزي لتحليل الفيروسات.
3. ينشئ ذلك الجهاز بيئةً يمكن فيها تشغيل البرنامج المصاب للتحليل بأمان. تتضمّن الأساليب المستخدمة لهذا الغرض المحاكاة، أو إنشاء

بيئة محميّة لتنفيذ البرنامج المشكوك فيه ومراقبته. بعد ذلك يُنتِج جهاز تحليل الفيروسات وصفة للتعرف على الفيروس وإزالته.

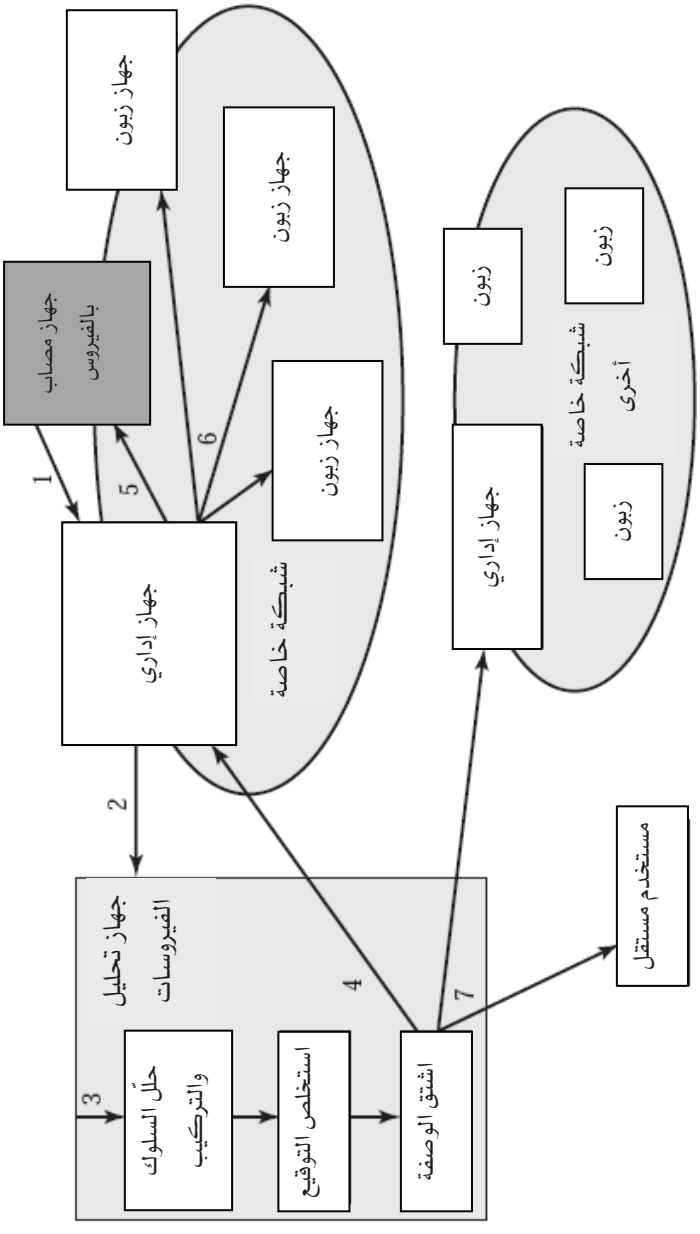
4. تُرسَل الوصفة الناتجة إلى جهاز الحاسب الإداري.
5. يُرسِل جهاز الحاسب الإداري الوصفة إلى الزبون المصاب.
6. تُرسَل الوصفة أيضاً إلى الزبائن الأخرى في المؤسسة.
7. يتلقّى المشتركون حول العالم بصفة منتظمة تحديثات للبرامج المضادة للفيروسات للحماية من الفيروسات الجديدة.

يعتمد نجاح نظام المناعة الرقمي على قدرة الجهاز الذي يقوم بتحليل الفيروسات على اكتشاف السلالات الجديدة والمبتكرة من الفيروسات. بتحليل ومراقبة الفيروسات التي تقبع في البرِّية، سيتسنى تحديث برامج المناعة الرقمية باستمرار لمواكبة التهديدات الجديدة.

10-2-3 برامج منع السلوك المشبوه (Behavior-Blocking Software)

على خلاف برامج المسح التي تعتمد على القواعد الاجتهادية أو على البصمة (توقيع الفيروس)، تُدمج برامج منع السلوك المشبوه مع نظام تشغيل الحاسب المضيف لتراقب سلوك البرامج بشكل آني لاكتشاف الأفعال المؤذية. عندئذ تقوم تلك البرامج بمنع الأفعال المؤذية المحتملة قبل أن تتمكّن من التأثير على النظام. يمكن أن يتضمّن السلوك المراقَب ما يأتي:

- محاولات فتح الملفات أو عرضها أو حذفها أو تعديلها.
- محاولات تهيئة (formatting) مشغل الأقراص وغير ذلك من عمليات الأقراص الأخرى غير القابلة للاسترداد.
- تعديلات لمنطق الملفات التنفيذية أو الملفات التي تتضمن ماكرو.
- تعديل إعدادات النظام الحرجة، كإعدادت بدء التشغيل.



الشكل 10-4: نظام مناعة رقمي.

- طلب إرسال محتوى قابل للتنفيذ بالبريد الإلكتروني أو عبر المراسلة الإلكترونية الفورية.
- إنشاء اتصال عبر الشبكة.

إذا اكتشف مانع السلوك المشبوه (behavior blocker) أن برنامجاً يشرع فيما يمكن أن يؤول إلى سلوكٍ مؤذٍ أثناء تشغيله، يمكنه منع هذا السلوك على الفور وإنهاء البرنامج المشبوه. ويُحقق ذلك ميزة أساسية على طرق الكشف المعروفة المضادة للفيروسات والمبنية على أساس البصمات أو القواعد الاجتهادية. فبينما يوجد حرفياً تريليونات من الطرق المختلفة لتشويش وإعادة ترتيب تعليمات الفيروس أو الدودة، كثيرٌ منها لن يتسنى اكتشافه من قِبَل ماسحات البصمات أو القواعد الاجتهادية، فإن أي كود مؤذٍ لابد في النهاية وأن يرسل طلباً واضح المعالم إلى نظام التشغيل. بافتراض أن مانع السلوك يستطيع اعترض كل مثل تلك الطلبات، فسيمكنه أن يكتشف ويمنع الأعمال المؤذية بغض النظر عن كيفية التشويه الذي تم إدخاله على منطق البرنامج للتمويه.

واضحٌ أنَّ القدرة على مراقبة البرامج أثناء تشغيلها في الوقت الحقيقي يحقق ميزةً كبيرةً لمانع السلوك المشبوه؛ ومع ذلك فإن لها مثالب أيضاً. نظراً لأنه ينبغي تشغيل الكود المؤذي على الجهاز المستهدف قبل أن يمكن تمييز كل جوانب سلوكه، فقد يتسبب في إلحاق كثير من الضرر بالنظام قبل اكتشافه ومنعه. على سبيل المثال، قد يخلط فيروس جديد عدداً من ملفات قد تبدو غير مهمة على القرص الصلب قبل إصابة أي ملف وقبل منعه. ورغم أن العدوى الفعلية قد أمكن منعها، فقد يعجز المستخدم عن تحديد مواقع ملفاته، مما يسبب فقداً في الإنتاجية أو يؤدي إلى عواقب أسوأ من ذلك.

3-10 هجمات حجب الخدمة الموزَّعة

تمثل هجمات حجب الخدمة الموزَّعة (DDoS) تهديداً أمنياً بالغ الخطورة للشركات، ويبدو أن خطر هذا التهديد في تزايد [VIJA02]. وفي إحدى الدراسات التي غطت فترة ثلاثة أسابيع في عام 2001، رصد الباحثون أكثر من 12,000 هجمة ضد أكثر من 5000 هدف محدّد. وتراوحت الأهداف من الشركات المشهورة للتجارة عبر الإنترنت كالأمازون والهوتميل إلى موفري خدمة الإنترنت (ISP) الأجانب الصغار ووصلات الاتصال عبر الهاتف [MOOR01]. وتجعل هجمات DDoS أنظمة الحاسب صعبة الوصول وذلك بغمر الخادمات، أو الشبكات، أو حتى الأنظمة الطرفية بحركة مرور عديمة الفائدة بحيث لا يستطيع المستخدمون الشرعيون الوصول إلى تلك الموارد. في هجوم DDoS نمطي يُطلب من عدد كبير من المضيفات المصابة إرسال رزمٍ عديمة الفائدة. غير أنه في السنوات الأخيرة أصبحت أساليب الهجوم وأدواته أكثر تطوراً وفعاليةً، وصار تتبّع أثر المهاجمين الحقيقيين أكثر صعوبةً، بينما بقت تقنيات الدفاع عاجزةً عن مقاومة الهجمات واسعة النطاق [CHAN02].

هجوم حجب الخدمة ((denial of service (DoS) هو محاولة لمنع مستخدمين شرعيين لخدمةٍ ما من استخدام تلك الخدمة. وعندما يَصدُر هذا الهجوم من مضيف واحد أو عقدة واحدة على الشبكة، يطلق عليه ببساطة هجوم DoS. ويُعدُّ هجوم حجب الخدمة الموزّع (DDoS) أكثر خطورة، حيث يكون المهاجم قادراً على تجنيد عددٍ من المضيفات في كافة أنحاء الإنترنت بشكلٍ آني ومنسّق للهجوم على الهدف. وسنركّز في هذا الجزء على هجمات DDoS. سننظر في البداية إلى طبيعة تلك الهجمات وأنواعها. ثم نتناول الوسائل التي تُمكِّن المهاجم من تجنيد مضيفات على الشبكة لإطلاق الهجوم. وأخيراً نناقش الإجراءات المضادة.

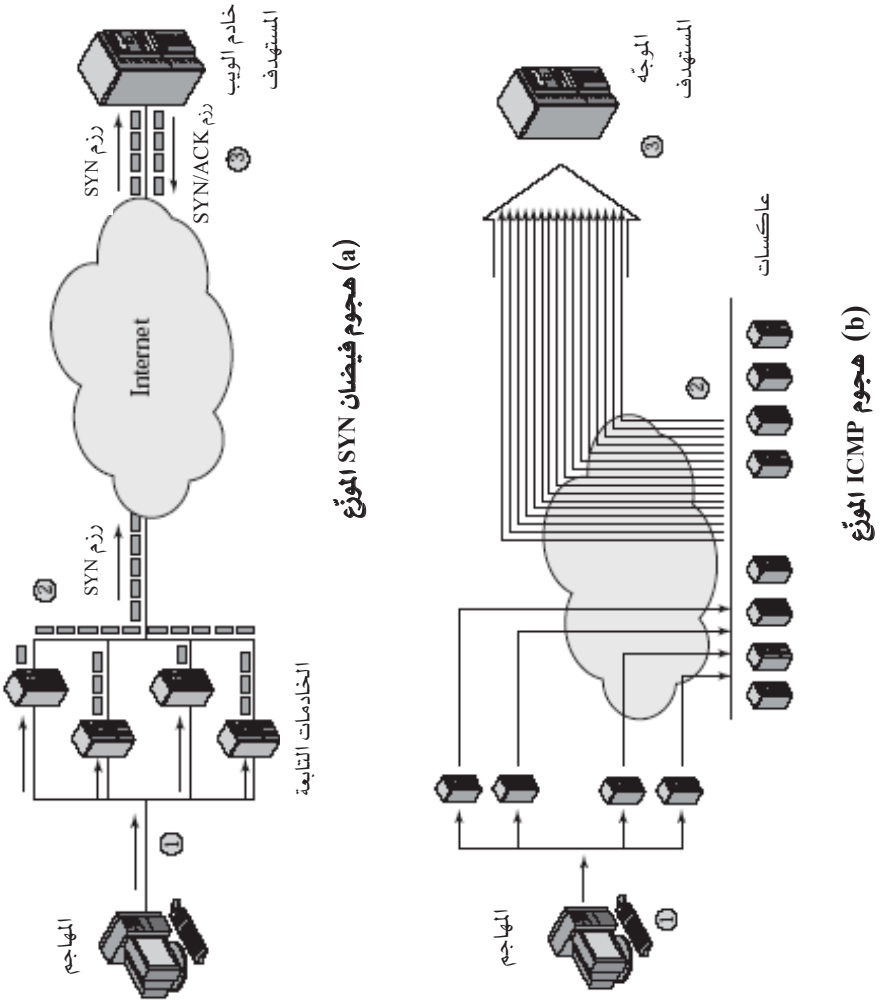
10-3-1 وصف هجوم DDoS

يحاول هجوم DDoS استهلاك موارد الهدف حتى لا يستطيع توفير الخدمة. وإحدى الطرق لتصنيف هجمات DDoS هي تصنيفها من حيث نوع المورد الذي يُستهلك. بشكل عام، يكون المورد الذي يُستهلك إما مورداً داخلياً على نظام المضيف المُستهدف، أو سعة نقل بيانات على الشبكة المحليّة الموصّل عليها ذلك المضيف. من الأمثلة البسيطة لهجوم مورد داخلي هجوم فيضان SYN.

يوضح الشكل 10-5 (a) الخطوات التي يتضمنها ذلك الهجوم:

1. أولاً يسيطر المهاجم على عدة مضيفات على الإنترنت، ويأمرها بالاتصال بخادم ويب الهدف.
2. تبدأ تلك المضيفات التابعة (slave hosts) بإرسال رزم SYN – للتزامن والتهيئة في بروتوكول TCP/IP – إلى الهدف بعنوان IP مزوّر للعودة.
3. تمثّل كل رزمة SYN طلباً لإنشاء توصيلة TCP. لكل رزمة من تلك الرزم، يردّ خادم الويب للمضيف المستهدف برزمة SYN/ACK (تزامن وإشعار استلام)، محاولاً تأسيس توصيلة TCP مع كيان TCP بعنوان IP المزوّر.

يحتفظ خادم الويب بهيكل بيانات لكل طلب SYN ينتظر الردّ، ومن ثم يصبح متورِّطاً أكثر كلما ازداد فيضان مرور بيانات. وتكون النتيجة عدم تلبية الطلبات لتوصيلات شرعية بينما يكون المضيف الضحيَّة ينتظر إكمال عددٍ كبيرٍ من التوصيلات المزيفة "نصف المفتوحة".



الشكل 10-5: أمثلة لهجمات DDoS البسيطة.

كثيراً ما يُستخدم هيكل بيانات حالة بروتوكول TCP كمورد داخلي مستهدف للهجوم، ولكنه ليس الوحيد. يعطي [CERT01] الأمثلة الآتية:

1. في كثير من الأنظمة يتوفر عددٌ محدود من هياكل البيانات لتخزين معلومات العملية (process) (كمُعرِّفات العملية، ومُدخلات جداول العملية، إلخ). قد يتمكن مهاجم من استهلاك هياكل البيانات تلك بكتابة برنامج بسيط لا يفعل شيئاً سوى إنشاء نسخة من نفسه مراراً وتكرارا.
2. قد يحاول المهاجم أيضاً استهلاك سعة القرص بطرق أخرى، مثل:
   - إنشاء عدد كبير جداً من رسائل البريد الإلكتروني.
   - تعمُد إنشاء أخطاء يقوم النظام بتسجيلها.
   - وضع ملفات في مناطق مجهولة بنظام نقل الملفات (ftp) أو مناطق مشتركة على الشبكة.

يوضح الشكل 10-5 (b) مثالاً لهجوم يستهلك موارد نقل البيانات، ويشمل الخطوات الآتية:

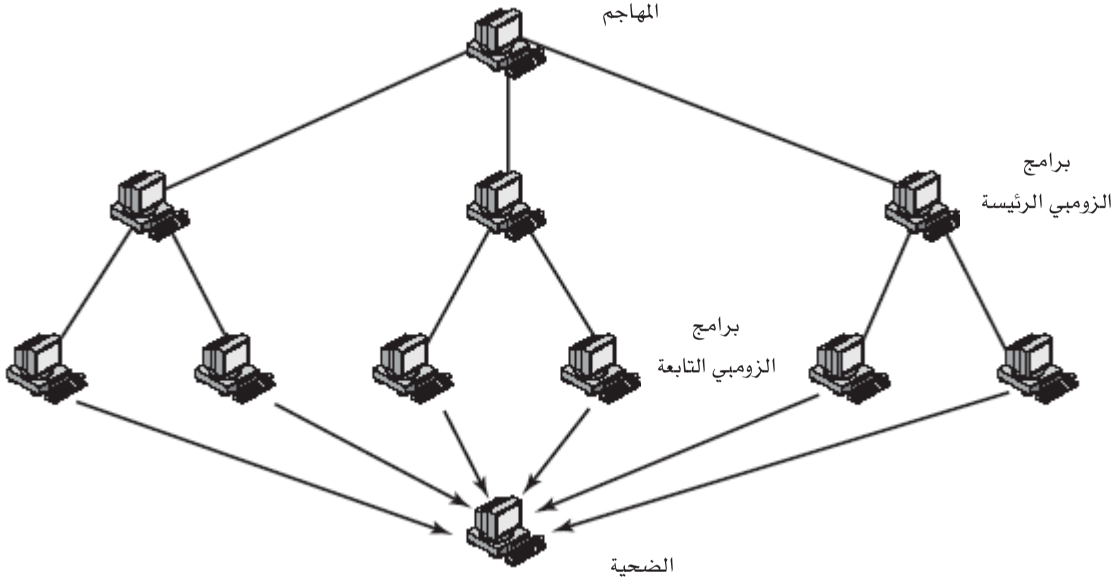
1. يسيطر المهاجم على عدة مضيفات على الإنترنت، ويأمرها بإرسال رزم صدى ICMP[1] بعنوان IP للهدف (كعنوان مزيف (spoofed address) للمصدر) إلى مجموعة من المضيفات التي تعمل بمثابة عاكسات (reflectors)، كما سنصف بعد ذلك.
2. تستلم العقد في موقع الإرجاع (bounce site) عدة طلبات مزيفة وترد بإرسال رزم رد للصدى إلى موقع الهدف.

[1] بروتوكول رسائل التحكم للإنترنت ((Internet Control Message Protocol (ICMP) هو بروتوكول في طبقة الشبكة (مستوى بروتوكول IP) لتبادل رزم التحكم بين موجّه ومضيف أو بين المضيفات. تتطلب رزمة الصدى (ECHO) من المستقبل أن يرد برزمة الاستجابة للصدى (echo reply) مما يفيد في فحص الاتصال بين كيانات مختلفة.

3. يصاب موجّه الهدف بفيضان الرزم من موقع الإرجاع، ومن ثم لا تبقى سعة كافية لنقل حركة البيانات الشرعية.

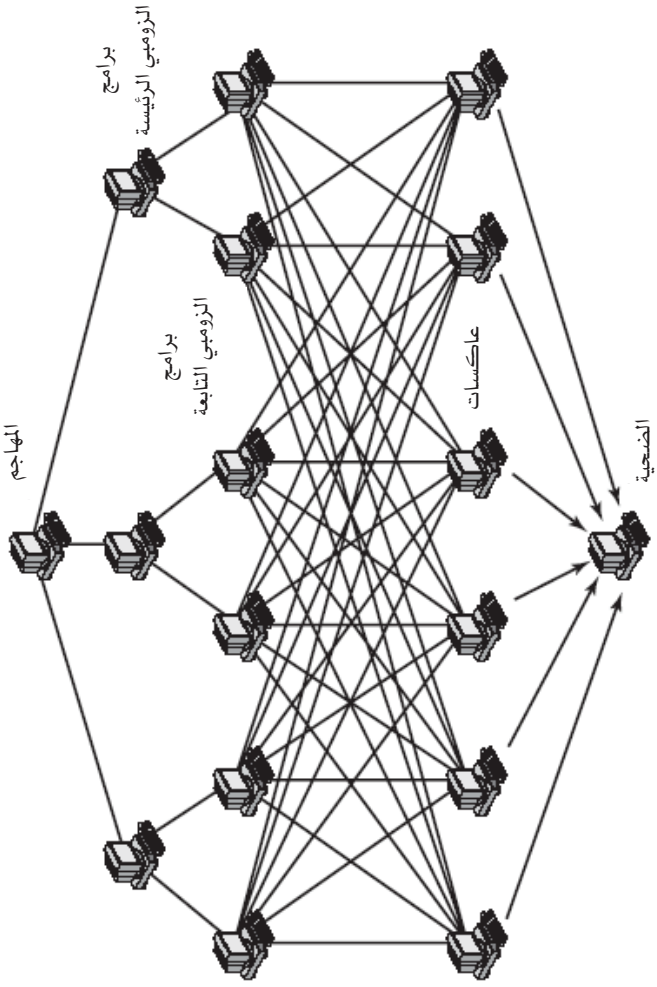
من طريق التصنيف الأخرى تصنيف هجمات DDoS كهجمات مباشرة (direct) أو هجمات انعكاسية (reflector). في هجوم DDoS المباشر (الشكل 10-6 (a))، يستطيع المهاجم زراعة برامج الزومبي على عدد من المواقع الموزَّعة في كافة أنحاء الإنترنت. في أغلب الأحيان، يتضمّن هجوم DDoS مستويين من مضيفات الزومبي: الزومبي الرئيس والزومبي التابع. وتُصاب المضيفات في كلا المستويين بالبرمجيات الخبيثة. ويوجِّه المهاجم مضيفات الزومبي الرئيسة وينسق فيما بينها، والتي تقوم بدورها بالإثارة والتنسيق فيما يتعلق بمضيفات الزومبي التابعة. إن استخدام مستويين من الزومبي يزيد من صعوبة تتبّع الهجوم للوصول إلى مصدره الأصلي، ويوفر بيئة مرنة لشبكة المهاجمين.

يضيف هجوم DDoS الانعكاسي طبقةً أخرى من المضيفات (الشكل 10-6 (b)). في هذا النوع من الهجوم، تنشئ مضيفات الزومبي التابعة رزماً تتطلب رداً يتضمن عنوان IP للهدف كعنوان IP للمصدر في ترويسة العنوان بالرزمة. وتُرسَل تلك الرزم إلى الأجهزة غير المصابة والمعروفة بالعاكسات. ترد الأجهزة غير المصابة برزم موجّهة نحو جهاز الهدف. يمكن أن يتضمّن هجوم DDoS الانعكاسي بسهولة مضيفات أكثر وحركة مرور بيانات أكبر من هجوم DDoS المباشر، ومن ثَمَّ يكون أكثر ضرراً. بل إن تتبّع الهجوم إلى مصدره الأصلي أو ترشيح رزم الهجوم يكون أكثر صعوبة؛ لأن الهجوم يأتي من أجهزة غير مصابة ومنتشرة على مناطق كبيرة.



(a) هجوم DDoS مباشر

الشكل 10-6 أنواع هجمات DDoS التي تعتمد على الفيضان.



(b) هجوم DDoS انعكاسي

تابع الشكل 10-6 أنواع هجمات DDoS التي تعتمد على الفيضان.

10-3-2 إنشاء شبكة الهجوم

في الخطوة الأولى من هجوم DDoS يتمكن المهاجم من إصابة عددٍ من الأجهزة ببرامج الزومبي التي ستُستخدم في النهاية لتنفيذ الهجوم. تضم المكونات الأساسية اللازمة للهجوم في هذه المرحلة ما يأتي:

1. البرامج التي يمكنها تنفيذ هجوم DDoS. يجب أن تصلح تلك البرامج لتشغيلها على عدد كبير من الأجهزة، وأن تكون قادرة على إخفاء وجودها، وأن تكون قادرة على الاتصال مع المهاجم أو يكون لها نوع من المثيرات الآلية الموقوتة، وأن تكون قادرة على شن الهجوم المقصود نحو الهدف.
2. نقطة ضعف في عددٍ كبيرٍ من الأنظمة. يجب أن يتعرف المهاجم على نقطة ضعفٍ أمنيةٍ فَشَل كثيرٌ من مديري الأنظمة والمستخدمين في ترقيعها ويستطيع المهاجم من خلالها تثبيت برامج الزومبي على تلك الأنظمة.
3. استراتيجية لتحديد أماكن الأجهزة التي تعاني من نقاط الضعف، وتُعرف هذه العملية بالمسح.

في عملية المسح، يبحث المهاجم أولاً عن عددٍ من الأنظمة التي تعاني من نقاط الضعف ويصيبها. وعادةً ما تُكرر برامج الزومبي التي تُثبَّت على الأنظمة المصابة بتكرار نفس عملية المسح إلى أن يتم تكوين شبكة كبيرة من الأنظمة الموزّعة المصابة. استعرَض [MIRK04] الأنواع الآتية من استراتيجيات المسح:

- المسح العشوائي: يتقصّى كل مضيف مصاب عناوين IP في فضاء العناوين بطريقة عشوائية باستخدام بذرة (seed) مختلفة. تُنتج هذه

الطريقة كميةً ضخمةً من حركة مرور بيانات الإنترنت، مما قد يسبب عرقلة عامة حتى قبل شن الهجوم الفعلي.

- المسح باستخدام قائمة الإصابة (hit-list): يجمع المهاجم أولاً قائمة طويلة من الأنظمة الضعيفة المحتملة. ويمكن أن تكون هذه عملية بطيئة وتتم على فترة طويلة لتفادي اكتشاف أن هجوماً يجري تنفيذه. وبمجرد تكوين القائمة، يبدأ المهاجم بإصابة الأنظمة الموجودة بالقائمة. ويُزوّد كل نظام مصاب بجزء من القائمة ليقوم بمسحها. وتؤدّي هذه الاستراتيجية إلى فترة مسح قصيرة جداً، مما قد يجعل من الصعب اكتشاف أن العدوى جارٍ انتشارها.
- المسح بشكل طوبولوجي: تستخدم هذه الطريقة معلومات موجودة على النظام الضحيّة لمعرفة المزيد من المضيفات لمسحها.
- مسح الشبكة الفرعية المحليّة: إذا أمكن إصابة مضيف خلف برنامج الحماية (الجدار الناري) (firewall)، يقوم ذلك المضيف بعد ذلك بالبحث عن أهداف ضمن شبكته المحليّة. يستخدم المضيف هيكل عناوين الشبكة الفرعية لإيجاد مضيفات أخرى محميَّة ببرنامج الجدار الناري (ضد الطرق الأخرى).

10-3-3 الإجراءات المضادة لهجوم DDoS

بشكلٍ عام هناك ثلاثة من خطوط الدفاع ضد هجمات DDoS [CHAN02]:

- منع الهجوم وإحباطه (قبل الهجوم): تُمكّن هذه الآليات النظام الضحيّة من تحمّل محاولات الهجوم بدون حجب الخدمة عن الزبائن الشرعيين. وتتضمّن الأساليب المستخدمة فرض سياسات لاستهلاك الموارد وتوفير

موارد احتياطية متوفرة عند الطّلب. بالإضافة لذلك، تُعدِّل تلك الآليات الأنظمة والبروتوكولات على الإنترنت لتقليل احتمالات هجمات DDoS.

- اكتشاف الهجوم والترشيح (أثناء الهجوم): تحاول هذه الآليات اكتشاف الهجوم بمجرد أن يبدأ وتستجيب لذلك فوراً، مما يقلّل من تأثير الهجوم على الهدف. ويتضمّن الكشف عن الهجوم البحث عن أنماط السلوك المريبة. ويتضمّن الردّ ترشيح رزم يُحتمل أن تكون جزءاً من الهجوم.
- تتبُّع مصدر الهجوم والتعرف عليه (أثناء وبعد الهجوم): تلك محاولة لمعرفة مصدر الهجوم كخطوة أولى لمنع الهجمات في المستقبل. ومع ذلك، لا تؤدي هذه الطريقة عادةً إلى نتائج سريعةً بما فيه الكفاية لتسكين هجوم مستمر، وقد لا تكون لها نتائج تُذكر.

يكمن التحدي في التعامل مع هجمات DDoS في الطرق الكثيرة التي يمكن أن تحدث بها. ولذا ينبغي أن تتطوَّر إجراءات المواجهة باستمرار لتواكب التقدم في أساليب الهجوم.

4-10 توصيات للمطالعة

[CASS01] Cass, S. "Anatomy of Malice." *IEEE Spectrum*, November 2001.

[CHAN02] Chang, R. "Defending Against Flooding-Based Distributed Denial-of-Service Attacks: A Tutorial." *IEEE Communications Magazine*, October 2002.

[FORR97] Forrest, S.; Hofmeyr, S.; and Somayaji, A. "Computer Immunology." *Communications of the ACM*, October 1997.

[HARL01] Harley, D.; Slade, R.; and Gattiker, U. *Viruses Revealed*. New York: Osborne/McGraw-Hill, 2001.

[KEPH97] Kephart, J.; Sorkin, G.; Chess, D.; and White, S. "Fighting Computer Viruses." *Scientific American*, November 1997.

[MEIN01] Meinel, C. "Code Red for the Web." *Scientific American*, October 2001.

[MIRK04] Mirkovic, J., and Relher, P. "A Taxonomy of DDoS Attack and DDoS Defense Mechanisms." *ACM SIGCOMM Computer Communications Review*, April 2004.

[NACH97] Nachenberg, C. "Computer Virus-Antivirus Coevolution." *Communications of the ACM*, January 1997.

[PATR04] Patrikakis, C.; Masikos, M.; and Zouraraki, O. "Distributed Denial of Service Attacks." *The Internet Protocol Journal*, December 2004.

[SZOR05] Szor, P., *The Art of Computer Virus Research and Defense*.Reading, MA: Addison-Wesley, 2005.

5-10 مصادر للمعلومات على الويب

- AntiVirus Online: موقع لشركة IBM يتضمن معلومات عن الفيروسات.
- Vmyths: موقع مخصص لعرض خدع الفيروسات (virus hoaxes) وتبديد المفاهيم الخاطئة عن الفيروسات الحقيقية.
- DDoS Attacks/Tools: قائمة واسعة من الوصلات التشعبية (عناوين الويب) والوثائق.

6-10 مصطلحات رئيسة

auto-rooter	برنامج للوصول التلقائي للجذر
backdoor	مدخل خلفي
digital immune system	نظام مناعة رقمي
direct DDoS attack	هجوم حجب الخدمة الموزَّع المباشر
distributed denial of service (DDoS)	حجب الخدمة الموزَّع
downloaders	برامج التنزيل
e-mail virus	فيروس البريد الإلكتروني
exploits	استغلالات (نقاط ضعف)
flooder	مُرسِل الفيضان
keyloggers	مسجلات ضربات المفاتيح
kit	علبة أدوات (لتوليد فيروسات)
logic bomb	قنبلة منطقية
macro virus	فيروس الماكرو
malicious software (malware)	برمجيات مؤذية
polymorphic virus	فيروس متعدد الأشكال
reflector DDoS attack	هجوم DDoS الانعكاسي
rootkit	الجذور الخفية (أو أدوات الجذر)
spammer program	برنامج إرسال رسائل الدعاية
stealth virus	فيروس مُتَخفٍّ
trapdoor	باب المصيدة (المدخل الخلفي)
Trojan horse	حصان طروادة
virus	فيروس
worm	دودة
zombie	برنامج زومبي

7-10 أسئلة للمراجعة ومسائل

1-7-10 أسئلة للمراجعة

1-10 ما دور الضغط (compression) في طريقة عمل الفيروس؟

2-10 ما دور التشفير في طريقة عمل الفيروس؟

3-10 ما المراحل المعتادة لعمل الفيروس والدودة؟

4-10 بشكلٍ عام كيف تنتشر الديدان؟

5-10 ما المقصود بنظام المناعة الرقمي؟

6-10 كيف تعمل برامج منع السلوك المشبوه؟

7-10 ماذا تعرف عن هجوم حجب الخدمة الموزَّع (DDoS)؟

2-7-10 مسائل

1-10 يوجد خلل في برنامج الفيروس الموجود بالشكل 10-1؛ ما هو؟

2-10 السؤال الذي يطرح نفسه هو ما إذا كان من الممكن تطوير برنامج يستطيع تحليل قطعة من البرمجيات ويحدد ما إذا كانت تمثل فيروساً أم لا؟ اعتبر أن لدينا برنامج D يُفترض أنه قادر على ذلك. بمعنى أنه لأي برنامج P، إذا شغّلنا D(P) فإن النتيجة تكون صح (true) في حالة ما إذا كان P فيروساً أو خطأ (false) إذا لم يكن P فيروساً. الآن خذ في الاعتبار البرنامج الآتي:

```
program CV :=
{ . . .
main-program :=
{
if D(CV) then
      goto next;
else
      infect-executable;
}
next:
}
```

في هذا البرنامج تُعدُّ infect-executable وحدة برمجية لمسح الذاكرة بحثاً عن برامج تنفيذية ونسخ نفسها في تلك البرامج. حدد ما إذا كان D يمكنه أن يقرر بدقة ما إذا كان CV عبارة عن فيروس أم لا.

11

الفصل الحادي عشر

الجدران النارية

محتويات الفصل:

11-1 مبادئ تصميم الجدران النارية

11-1-1 خصائص الجدران النارية

11-1-2 أنواع الجدران النارية

11-1-3 تهيئة الجدران النارية

11-2 الأنظمة الموثوقة

11-2-1 التحكم في الوصول للبيانات

11-2-2 مفهوم الأنظمة الموثوقة

11-2-3 الدفاع ضد حصان طروادة

11-3 المعايير المشتركة لتقييم أمن تقنية المعلومات

11-3-1 المتطلبات

11-3-2 الملامح والأهداف

11-4 توصيات للمطالعة

11-5 مصادر للمعلومات على الويب

11-6 مصطلحات رئيسة

11-7 أسئلة للمراجعة ومسائل

"تعتمد قوة الموقع على جعل قوي الحماية المحيطة به غير قابلة للاختراق بشكلٍ عمليّ" – من كتاب "عن الحرب"، لكارل فون كلاوزفيتز.

"في اليوم الذي تصبح فيه قائداً، سُدّ معابر الحدود، وحطّم السجلات الرسمية، وأوقف مرور جميع المبعوثين" – من كتاب "فن الحرب" لصن تزو.

النقاط الأساسية:

- يُشكل الجدار الناري (جدار الحماية) حاجزاً لحركة المرور بالشبكة في الاتجاهات المختلفة، حيث يقوم بالسماح أو عدم السماح بالمرور طبقاً للسياسة الأمنية لهذا الجدار.
- يمكن تصميم الجدار الناري للعمل كمرشح على مستوى رزم بروتوكول الإنترنت (IP packets)، أو للعمل على أيٍّ من الطبقات العليا للبروتوكولات.
- يتكون النظام الموثوق من جهاز حاسب ونظام تشغيل يمكن التحقق من تنفيذهما لسياسة أمنية معينة. وعادةً، يركِّز مثل هذا النظام على عملية التحكم في الوصول (access control)، وذلك تبعاً لسياسة تُحدِّد المستخدمين الذين يُسمح لهم بالوصول لكل خدمة من الخدمات المتاحة.
- المعايير المشتركة لأمن المعلومات هي مبادرة دولية لتحديد مجموعة مشتركة من متطلبات الأمن ووسائل محدَّدة ومنضبطة لتقييم المنتجات على أساس تلك المتطلبات.

يمكن للجدران النارية أن تكون وسائل فعَّالة لحماية نظام محلي أو شبكة نظم من التهديدات الأمنية المتعلقة بالشبكات، كما توفر في نفس الوقت إمكانية الاتصال بالعالم الخارجي عن طريق الشبكات واسعة النطاق وشبكة الإنترنت.

سوف نبدأ هذا الفصل بلمحة عامة عن وظائف الجدران النارية ومبادئ تصميمها، ثم نعالج بعدها قضية أمن الجدار الناري نفسه، وبصفة خاصة مفهوم الأنظمة الموثوقة، أو أنظمة التشغيل الآمنة.

1-11 مبادئ تصميم الجدران النارية

لقد شهدت نظم المعلومات في الشركات والوكالات الحكومية والمنظمات الأخرى تطوراً مستمراً تَمثَّل في المراحل الآتية:

- مرحلة النظام المركزي لمعالجة البيانات: يتكون من حاسب مركزي أساسي يدعم عدداً من المحطات الطرفية المتصلة به مباشرةً.
- مرحلة الشبكات المحلية (LANs): تربط أجهزة الحاسب الشخصية والمحطات الطرفية بعضها ببعض كما تربطها أيضاً بالحاسب المركزي.
- مرحلة شبكة المنشآت (Premises network): تتألَّف من عدد من الشبكات ذات النطاق المحلي، لربط الحاسبات الشخصية مع الأنظمة الخادمة وربما مع حاسب مركزي أو حاسبين مركزيين.
- مرحلة شبكة المؤسسة (Enterprise-wide network): تتألَّف من عدة شبكات منشآت موزعة جغرافيا ومرتبطة ببعضها عن طريق شبكة خاصة واسعة النطاق (WAN).
- مرحلة الارتباط بالإنترنت: يتم فيها ربط جميع شبكات المنشآت بشبكة الإنترنت. يُمكِن أيضاً أن تكون تلك الشبكات مرتبطة فيما بينها من خلال شبكة خاصة واسعة النطاق بالإضافة إلى ارتباطها عن طريق الإنترنت.

لم يعد الارتباط بالإنترنت أمراً اختيارياً بالنسبة للمؤسسات أو الشركات، حيث أصبحت المعلومات والخدمات المتاحة على الإنترنت ضرورية لأي مؤسسة. وعلاوةً على ذلك يرغب المستخدمون بتلك المؤسسات في الاتصال بالإنترنت، وإن لم يتوفر لهم ذلك عبر شبكاتهم المحلية، فسيقومون باستخدام الاتصال الهاتفي من حاسباتهم الشخصية إلى أي موفر لخدمة الإنترنت (ISP). وبالرغم من أن الارتباط بالإنترنت يوفر فوائد للمؤسسات، فإنه يُمكِّن العالم الخارجي من الوصول إلى مكوِّنات الشبكة المحلية والتفاعل معها مما يشكل تهديداً لتلك المؤسسات. في

حين أنه من الممكن تجهيز كل محطة عمل خادم على الشبكة الخاصة بخصائص أمنية قوية – كالحماية ضد الاختراق – فإن هذا لا يُعدُّ أمراً عملياً. وإذا اعتبرنا شبكة تتألَّف من المئات أو حتى الآلاف من الأنظمة، وتعمل وفقَ مزيجٍ من الإصدارات المختلفة لأنظمة تشغيل مختلفة مثل يونيكس (UNIX) أو ويندوز، فإن أي ثغرة أمنية يتم اكتشافها تستدعي القيام بتحديث كل أنظمة التشغيل التي يُحتمل تأثرها بتلك الثغرة حتى يتم علاجها. أما العلاج البديل لذلك – والذي يلاقي قبولاً متزايداً – فهو الجدار الناري، ويتم وضعه بين الشبكة الخاصة والإنترنت كوصلة يمكن التحكم فيها وللعمل بمثابة سياج أمني خارجي. الهدف من ذلك هو حماية الشبكة الخاصة من هجمات الإنترنت والعمل كنقطة تفتيش يمكنها فرض الأمور الأمنية ومراجعتها. قد يتكون هذا الجدار الناري من جهاز حاسب واحد أو عدد من الأجهزة (اثنين أو أكثر) التي تتعاون فيما بينها لأداء تلك الوظيفة.

في هذا الجزء من هذا الفصل، سنستعرض الخصائص العامة للجدران النارية، ثم على أنواع الجدران النارية الأكثر استخداماً في الوقت الحالي، وأخيراً سنقوم بفحص بعض إعدادات الجدار الناري الأكثر شيوعاً.

1-1-11 خصائص الجدار الناري

حدد المرجع [BELL94b] قائمة الأهداف الآتية للتصاميم المختلفة للجدار الناري:

1. المرور من داخل الشبكة إلى خارجها، والعكس بالعكس، لابد وأن يكون عبر الجدار الناري. ويتم ذلك عن طريق منع الوصول إلى الشبكة المحلية إلا من خلال الجدار الناري. وكما سيتضح لاحقاً يمكن إنجاز ذلك بأشكال متعددة.
2. تحدّد السياسة الأمنية المحلية ما يُسمح له وما لا يُسمح له بالعبور من خلال الجدار الناري، كما سيتم إيضاحه لاحقاً في هذا الجزء. هناك أنواع متعددة من الجدران النارية تنفذ أنماطاً مختلفة من السياسات الأمنية.

3. ينبغي أن يكون الجدار الناري ذاته محصَّناً ضد الاختراق، مما يعني ضرورة استخدام نظام حاسب موثوق مع نظام تشغيل آمن؛ وسنتناول هذا الموضوع في الجزء 11-2.

حدَّد المرجع [SMIT97] أربعة أساليب عامة تستخدمها الجدران النارية للتحكم في الوصول (النفاذ) ولفرض السياسة الأمنية للموقع. كان تركيز الجدران النارية بالدرجة الأولى على التحكم في الخدمات، لكنها تطورت منذ ذلك الحين للقيام بالمهام الأربعة الآتية:

- التحكم في الخدمات: يحدد أنواع خدمات الإنترنت – الداخلية منها والخارجية – التي يمكن الوصول إليها. وقد يقوم الجدار الناري بفلترة حركة المرور على أساس عنوان بروتوكول الإنترنت (IP address)، ورقم منفذ TCP (بروتوكول التحكم في الإرسال)؛ كما قد يقوم بدور الخادم المفوَّض (البروكسي) الذي يستقبل كل طلب خدمة ويفسره قبل تمريره؛ كما قد يقوم باستضافة برامج خادمات أخرى كخدمة الويب والبريد الإلكتروني.
- التحكم في الاتجاه: يحدد اتجاه كل طلب من طلبات الخدمة ليتم بدؤها والسماح بتدفقها عبر الجدار الناري.
- التحكم في المستخدم: يتحكم في الوصول إلى خدمةٍ ما حسب المستخدم الذي يحاول الوصول لتلك الخدمة. و عادة ما تطبق هذه الخاصية على المستخدمين داخل محيط الجدار الناري (المستخدمين المحليين). كما يمكن تطبيقها على طلبات الخدمة الواردة من المستخدمين الخارجيين؛ وهذا يتطلب إحدى تقنيات التوثيق الآمنة مثل التي يتم توفيرها في بروتوكول الإنترنت الآمن (IPSec) (انظر الفصل السادس).
- التحكم في السلوك: يتحكم في الكيفية التي يمكن بها استخدام خدمات معينة. فعلى سبيل المثال، يمكن للجدار الناري ترشيح (فلترة) البريد الإلكتروني للقضاء على رسائل الدعاية الإلكترونية (spam)، أو قد

يمكنه السماح للمستخدمين الخارجيين بالوصول إلى بعض أجزاء من المعلومات الموجودة على خادم الويب المحلي.

قبل الشروع في تفاصيل أنواع الجدار الناري وتكويناته، فإنه من المفيد تلخيص ما يمكن للمرء أن يتوقعه من الجدار الناري كما يأتي:

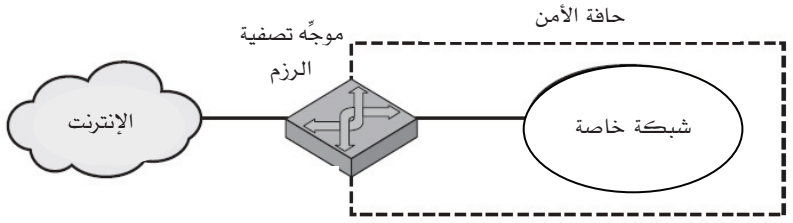
1. يُحدِّد الجدار الناري نقطة اختناق (choke point) تمنع المستخدمين غير المصرّح لهم من دخول الشبكة المحميَّة، وتحظر على الخدمات الحساسة الدخول أو الخروج من الشبكة، كما توفر الحماية ضد جميع أنواع التحايل على عناوين بروتوكول الإنترنت (IP spoofing) وهجمات التوجيه (routing attack) المختلفة. إن استخدام نقطة اختناق واحدة يُبَسِّط عملية الإدارة الأمنية لأن القدرات الأمنية مدموجة على حاسب واحد أو عدد محدود من الحاسبات.
2. يوفر الجدار الناري موقعاً لرصد العمليات المتعلقة بالأمن. كما يمكن تنفيذ عمليات التدقيق والمراجعة بالإضافة إلى الإنذار على نظام الجدار الناري.
3. يُمثِّل الجدار الناري منصَّة ملائمة للعديد من وظائف الإنترنت غير المتعلقة بالأمن، والتي تشمل مترجم عنوان الشبكة، والذي يترجم العناوين المحلية لعناوين الإنترنت المقابلة، وكذلك وظيفة إدارة الشبكة التي تدقق أو تسجل استخدامات الإنترنت.
4. يمكن للجدار الناري أن يكون بيئةً ملائمةً لبروتوكول الإنترنت الآمن (IPSec)، وذلك باستخدام نمط النفق (tunnel mode) الموصوف بالفصل السادس، كما يمكن أيضاً استخدام الجدار الناري لتنفيذ الشبكات الخاصة الافتراضية ((Virtual Private Networks (VPNs).

للجدران النارية حدودٌ لا تتعداها، فمثلاً:

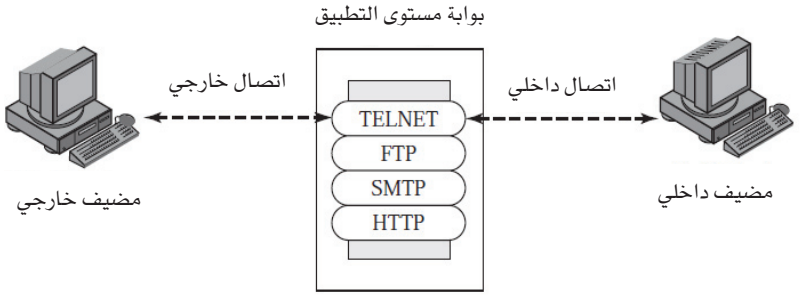
1. لا يمكن للجدار الناري توفير الحماية ضد الهجمات التي تتخطاه. فمثلاً قد يتوفر للنظم الداخلية إمكانية الاتصال الهاتفي بموفر خدمة الإنترنت (ISP)، وقد تدعم الشبكة المحلية الداخلية وحدة لأجهزة المودم التي توفر إمكانية الاتصال بالشبكة من الخارج للموظفين المسافرين.
2. لا يوفر الجدار الناري حمايةً ضد التهديدات الداخلية، مثل الموظف الساخط أو الموظف الذي يتعاون عن غير قصد مع أحد المهاجمين الخارجيين.
3. لا يمكن للجدار الناري الحماية ضد نقل برامج أو ملفات محمَّلة بالفيروسات. فنظراً لتنوع أنظمة التشغيل والتطبيقات البرمجية المعتمدة داخل محيط هذا الجدار، فإنه من غير العملي وربما من المستحيل على الجدار الناري فحص كافة الملفات الواردة ورسائل البريد الإلكتروني للتأكد من خلوها من الفيروسات.

11-1-2 أنواع الجدران النارية

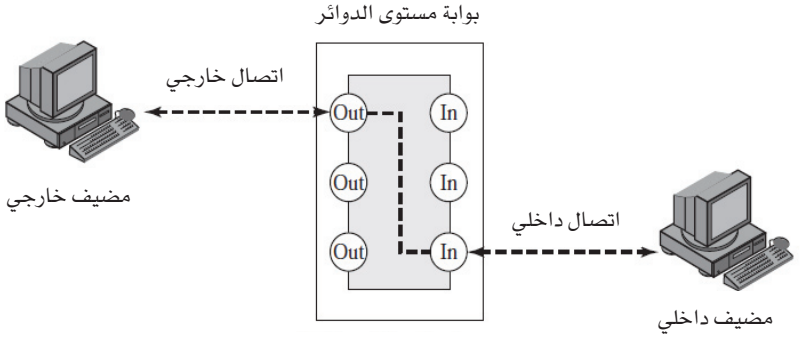
يوضِّح الشكل 11-1 ثلاثة أنواع شائعة من الجدران النارية: مرشحات الرزم (مصفيات أو فلاتر)، وبوابات مستوى التطبيق، وبوابات مستوى الدوائر. وسوف نتناول هذه الأنواع بالترتيب.



(a) موجِّه تصفية الرزم



(b) بوابة مستوى التطبيق



(c) بوابة مستوى الدوائر

الشكل 11-1: أنواع الجدران النارية.

❖ مرشحات الرزم (Packet Filters):

يطبِّق موجِّه ترشيح الرزم مجموعة من القواعد على كل رزمة من رزم بروتوكول الإنترنت قادمة أو مغادرة ومن ثَمَّ يقوم بتمريرها أو التخلص منها. ويتم عادةً تهيئة الموجِّه لترشيح الرزم في كلا الاتجاهين (من الشبكة الداخلية وإليها). تستند قواعد الترشيح على المعلومات المتضمَّنة في الرزمة:

- عنوان بروتوكول الإنترنت (IP address) للمصدر: عنوان بروتوكول الإنترنت للنظام الذي أنشأ تلك الرزمة (على سبيل المثال: 192.178.1.1).
- عنوان بروتوكول الإنترنت (IP address) للوجهة: عنوان بروتوكول الإنترنت للنظام الذي تحاول تلك الرزمة الوصول إليه (على سبيل المثال: 192.168.1.2).
- عنوان مستوى النقل (transport-level address) للمصدر والوجهة: رقم منفذ بروتوكول طبقة النقل (مثل: TCP أو UDP)، والذي يحدد نوعية تطبيقات الشبكة مثل SNMP أو TELNET.
- حقل بروتوكول الإنترنت: ويحدد بروتوكول النقل.
- الواجهة: بالنسبة لموجِّه ذي ثلاثة منافذ أو أكثر، هي واجهة الموجِّه التي تأتي منها الرزمة أو تتجه إليها.

وعادةً ما تتم تهيئة مرشح الرزم بتحديد قائمة من القواعد المبنيَّة على تطابقات مع حقول بترويسة بروتوكول الإنترنت أو ترويسة TCP. فإذا وُجد تطابق مع أحد تلك القواعد، فإنه وتبعاً لتلك القاعدة يتم تمرير الرزمة أو التخلص منها. وإذا لم يوجد تطابق مع أي من القواعد، فيتم اتخاذ إجراء مفترض. ويمكن اتباع إحدى سياستين مفترضتين:

- الإجراء المفترض = التخلص من الرزمة (اسقاطها): فالذي لا يُنص صراحةً على السماح له يُحظر.
- الإجراء المفترض = تمرير الرزمة: فالذي لا يُنص صراحةً على حظره يُسمح له.

الجدول 11-1: أمثلة لترشيح الرزم.

(A)

الإجراء	مضيفنا	المنفذ	مضيفهم	المنفذ	التعليق
ممنوع	*	*	SPIGOT	*	لا نثق بهؤلاء الناس
مسموح	OUR-GW	25	*	*	الاتصال بمنفذ SMTP الخاص بنا

(B)

الإجراء	مضيفنا	المنفذ	مضيفهم	المنفذ	التعليق
ممنوع	*	*	*	*	افتراضي

(C)

الإجراء	مضيفنا	المنفذ	مضيفهم	المنفذ	التعليق
مسموح	*	*	*	25	الاتصال بمنفذ SMTP الخاص بنا

(D)

الإجراء	المصدر	المنفذ	الوجهة	المنفذ	الأعلام	التعليق
مسموح	مضيفاتنا	*	*	25		رزمنا لمنفذ SMTP الخاص بهم
مسموح	*	25	*	*	ACK	ردودهم

(E)

الإجراء	المصدر	المنفذ	الوجهة	المنفذ	الأعلام	التعليق
مسموح	مضيفاتنا	*	*	*		اتصالاتنا الصادرة
مسموح	*	*	*	*	ACK	الردود على اتصالاتنا
مسموح	*	*	*	> 1024		حركة المرور لغير الخادمات

التخلص من الرزمة كإجراء مفترض هي السياسة الأكثر تحفظاً. فبدايةً يتم حظر كل شيء حيث تُضاف الخدمات الواحدة تلو الأخرى كلٌ حسب حالتها. وهذه السياسة هي الأكثر وضوحاً للمستخدمين الذين يرون الجدار الناري عائقاً. أما سياسة تمرير الرزمة كإجراء مفترض فهي أكثر سهولةً بالنسبة للمستخدمين ولكنها أقل أمناً؛ حيث يجب على مسؤول الأمن التعامل السريع مع كل تهديد جديد للأمن بمجرد اكتشافه.

يعطي الجدول 11-1 – المأخوذ من [BELL94b] – أمثلةً على بعض مجموعات قواعد ترشيح الرزم. يتم تطبيق القواعد في كل مجموعة بالترتيب من أعلاها إلى أسفلها. يستخدم الرمز "*" في حقل ما كرمز عام يتطابق مع كل شيء. وسنعتبر هنا أن الإجراء المفترض هو التخلص من الرزمة.

A. يُسمح للبريد الوارد بالنفاذ (منفذ 25 لواردات SMTP)، ولكن فقط لبوَّابة المضيف (gateway host). ولكن يمكن حظر الرزم الواردة من مضيف خارجي معين SPIGOT نظراً لوجود سوابق عديدة لهذا المضيف في إرسال ملفات ضخمة من خلال رسائل البريد الإلكتروني.

B. هذا بيان صريح للسياسة المفترضة. وتحوي جميع مجموعات القواعد هذه القاعدة بشكلٍ ضمني كقاعدتها الأخيرة.

C. تهدف مجموعة القواعد هذه إلى النص على أن أي مضيف داخلي يمكنه إرسال بريد إلى الخارج. فأي رزمة TCP تقصد منفذ 25 يتم توجيهها إلى خادم بروتوكول SMTP الخاص بحاسب الوجهة (destination machine). ولكن المشكلة في هذه القاعدة هو أن استخدام المنفذ 25 لاستلام SMTP يتم فقط كوضع مفترض؛ بمعنى أنه من الممكن تهيئة حاسب آخر خارجي لربط تطبيقات أخرى بالمنفذ 25. وبما أن هذه قاعدة مكتوبة، فإنه يمكن للمهاجمين النفاذ إلى الحاسبات الداخلية عن طريق إرسال رزم برقم 25 لمنفذ مصدر TCP.

D. تحقق مجموعة القواعد هذه النتيجة المرجوَّة التي لم تتحقق في (C). وتستفيد هذه القواعد من أحد خصائص وصلات الـ TCP. فبمجرد إنشاء

اتصالٍ ما، يتم ضبط مؤشر ACK لقطعة TCP في الوضعية 1 للإشعار باستلام القطع المرسلة من الطرف الآخر. ومن ثَمَّ فإن مجموعة القواعد هذه تقضي بالسماح لرزم بروتوكول الإنترنت التي يكون عنوان بروتوكول الإنترنت للمصدر فيها هو أحد المضيفات الداخلية ورقم منفذ الـ TCP للوجهة فيها هو 25. كما تقضي أيضاً بالسماح بمرور الرزم الواردة والتي يكون رقم منفذ الـ TCP للمصدر فيها هو 25 والتي يكون مؤشر ACK لقطعة TCP بها في الوضعية 1. لاحظ أننا عيَّنَّا نظامي المصدر والوجهة صراحةً لتحديد هذه القواعد بشكلٍ واضح.

E. مجموعة القواعد هذه هي نهج لمعالجة توصيلات FTP. ففي حالة توصيلات FTP، يتم استخدام توصيلتي TCP: توصيلة تحكم للتجهيز لنقل الملف، وتوصيلة بيانات للنقل الفعلي للملف. وتستخدم توصيلة البيانات رقم منفذ مختلفاً يتم تعيينه بطريقة ديناميكية لعملية النقل. ومعظم الأنظمة الخادمة، ومن ثَمَّ معظم أهداف الهجوم، تعمل على منافذ لها أرقام منخفضة؛ وفي المقابل تميل معظم المكالمات الصادرة إلى استخدام المنافذ ذات الأرقام العالية (عادةً أكبر من 1023). ولذا فإن مجموعة القواعد هذه تسمح بما يأتي:
   - الرزم التي تنشأ داخلياً.
   - رزم الرد على اتصال أنشئ عن طريق حاسب داخلي.
   - رزم متجهة لمنافذ عالية الرقم بأيٍّ من الحاسبات الداخلية.

   يتطلب هذا المخطط تهيئة النظم بحيث تسمح باستخدام المنافذ ذات الأرقام المناسبة فقط.

تشير مجموعة القواعد (E) إلى صعوبة التعامل مع التطبيقات التي تعمل على مستوى ترشيح الرزم. وهناك طريقة أخرى أيضاً للتعامل مع بروتوكول نقل الملفات (FTP) والتطبيقات المماثلة باستخدام بوَّابة مستوى التطبيق (application-level gateway) وسنتناولها لاحقاً في هذا الجزء.

أحد أهم مزايا موجِّه ترشيح الرزم (packet-filtering router) هو بساطته الشديدة. وعادةً ما تكون مرشحات الرزم شفافةً للمستخدمين وعالية السرعة. ويذكر المَرجع [WACK02] القائمة التالية لنقاط الضعف (الثغرات) الموجودة بالجدران النارية الخاصة بمرشحات الرزم:

- لا تقوم مرشحات الرزم بفحص بيانات الطبقات العليا، ولذا لا تستطيع منع الهجمات التي تستغل نقاط ضعف أو وظائف تطبيقات محدّدة. فعلى سبيل المثال، لا يمكن لجدار ناري يعمل مرشحاً للرزم عرقلة أوامر تطبيق محدَّد؛ لأنه إذا سمح ذلك الجدار الناري لتطبيق معين، فإن جميع الوظائف المتاحة في هذا التطبيق يكون مسموحاً بها أيضاً.
- نظراً لقلة المعلومات المتاحة للجدار الناري لترشيح الرزم ، فإن وظيفة التسجيل/التدوين (logging) الموجودة بهذه الجدران النارية محدودة للغاية، وعادةً تحتوي سجلات مرشح الرزم على نفس المعلومات التي تُستخدم عادةً في اتخاذ قرارات التحكم في الوصول (عنوان المصدر، وعنوان الوجهة، ونوع حركة المرور).
- معظم الجدران النارية لترشيح الرزم لا تدعم الأساليب المتقدمة لتوثيق المستخدمين. ومرةً أخرى، فإن هذا العيب في الغالب ناتجٌ عن عدم توفر وظائف الطبقات العليا بالجدار الناري.
- كما أن الجدران النارية عموماً عرضة للهجمات والإساءات التي تستغل المشكلات الموجودة بمواصفات ورصَّة بروتوكولات TCP/IP، مثل تزييف عنوان طبقة الشبكة. الكثير من الجدران النارية لترشيح الرزم لا يمكنها الكشف عن رزم الشبكات التي تم بها تغيير معلومات العنونة الخاصة بطبقة OSI الثالثة. وعموماً يتم استخدام هجمات الانتحال (التحايل وتزييف العناوين) من قبل الدخلاء من أجل تجاوز الضوابط الأمنية المنفَّذة بالجدار الناري.
- أخيراً، ونظراً لعدد المتغيرات القليلة المستخدمة في اتخاذ قرارات التحكم في الوصول، فإن الجدران النارية لترشيح الرزم عرضة للخروقات الأمنية

الناجمة عن تهيئتها بطريقة خاطئة. وبعبارة أخرى، من السهل تهيئة الجدران النارية لترشيح الرزم بصورة خاطئة غير متعمدة مما يسمح بمرور أنواع معينة من حركة المرور أو المصادر أو الوجهات التي ينبغي أن تُحظَر وفقاً لسياسة أمن المعلومات الخاصة بتلك المنظمة.

وفيما يلي بعض الهجمات التي يمكن أن تستهدف موجِّهات ترشيح الرزم والتدابير المضادة المناسبة:

- انتحال عنوان بروتوكول الإنترنت (IP address spoofing): وفيه يُرسِل الدخيل (المهاجم) رزماً من الخارج مع حقل عنوان IP للمصدر يحتوي على عنوان مضيف داخلي، ويأمل المهاجم في أن يسمح استخدام العنوان المزيف باختراق النظم التي تستخدم احتياطات أمنية بسيطة تعتمد على عنوان المصدر ويتم فيها قبول الرزم القادمة من المضيفات الداخلية الموثوقة. والتدبير المضاد في هذه الحالة هو التخلص من الرزم التي لها عنوان مصدر داخلي إذا وصلت هذه الرزم عن طريق واجهة خارجية.
- هجمات توجيه المصدر (source routing attacks): يقوم المصدر بتحديد المسار الذي ينبغي أن تتخذه الرزمة في عبورها خلال شبكة الإنترنت، أملاً في تجاوز الإجراءات الأمنية التي لا تقوم بتحليل معلومات توجيه المصدر. والتدبير المضاد في هذه الحالة هو التخلص من كافة الرزم التي تستخدم هذا الخيار.
- هجمات التجزئة الدقيقة (tiny fragment attacks): يستخدم الدخيل خيار تجزئة بروتوكول الإنترنت لإنشاء رزم جزئية صغيرة جداً، ويجعل معلومات ترويسة TCP في جزئية رزمة منفصلة. يصمم هذا الهجوم بطريقة تتحايل على قواعد الترشيح التي تعتمد على معلومات ترويسة TCP. فى العادة يقوم مرشح الرزم باتخاذ قرار الترشيح اعتماداً على الجزئية الأولى من الرزمة، ثم يتم ترشيح جميع الجزئيات اللاحقة من تلك الرزمة على أساس أنها تشكل جزءاً من الرزمة التي تم رفض جزئيتها الأولى. لكن في هذا الهجوم يأمل المهاجم أن يقوم موجِّه الترشيح بفحص الجزئية الأولى فقط وتمرير

الجزئيات المتبقية. يمكن التغلب على هجمات التجزئة الدقيقة بفرض قاعدة تنص على أنه يجب أن تحتوي الجزئية الأولى من الرزمة على قدرٍ محدّدٍ مسبقاً من ترويسة النقل كحدٍ أدنى. وإذا تم رفض الجزئية الأولى، فإنه يمكن للمرشح تذكُّر الرزمة ونبذ جميع جزئياتها اللاحقة.

❖ جدران تفتيش الحالة (Stateful Inspection Firewalls):

يتخذ مرشح الرزم التقليدي قرار الترشيح لكل رزمة على حدة بدون أن يأخذ في الاعتبار أي سياق للطبقات العليا. نحتاج للنظر في بعض خلفيات هذا الموضوع؛ لفهم ما هو المقصود من السياق، ولفهم سبب محدودية قدرة مرشح الرزم التقليدي فيما يتعلق بالسياق. تتبع معظم التطبيقات القياسية المبنية على بروتوكول TCP نموذج الزبون/الخادم. فعلى سبيل المثال، بالنسبة لبروتوكول نقل البريد البسيط (SMTP)، يتم نقل البريد الإلكتروني من نظام الزبون إلى نظام الخادم حيث يقوم نظام الزبون بإرسال رسائل بريد إلكتروني جديدة – الصادرة عادةً من المستخدمين – ويقوم نظام الخادم بدوره بنقل الرسائل ووضعها في صناديق بريد المُرسَل إليهم. يعمل بروتوكول نقل البريد الإلكتروني من خلال إقامة اتصال TCP بين الزبون والخادم، ويكون فيه رقم منفذ خادم TCP الذي يحدد تطبيق خادم بروتوكول SMTP هو 25. أما رقم منفذ بروتوكول TCP لزبون بروتوكول SMTP، فيقع بين 1024 و65535 ويتم إنشاؤه من قبل زبون بروتوكول SMTP.

بصفة عامة عندما يُنشئ تطبيق من التطبيقات التي تستخدم TCP جلسةً مع مضيف بعيد، فإنه يُنشئ اتصال TCP يكون فيه رقم منفذ TCP للتطبيق البعيد (الخادم) هو عدد أقل من 1024، ويكون رقم منفذ TCP للتطبيق المحلي (الزبون) هو عدد بين 1024 و65535. تخصَّص أرقام المنافذ الأقل من 1024 وهي الأرقام "المعروفة جيداً" (well-known) بشكلٍ دائم لتطبيقات محدّدة (كرقم 25 لخادم بروتوكول SMTP). أما الأرقام التي تقع بين 1024 و65535 فيتم توليدها ديناميكياً وتخصيصها بصفة مؤقتة فقط خلال فترة اتصال TCP.

يجب أن يسمح الجدار الناري البسيط لترشيح الرزم بمرور ما يَرِدُ إلى جميع المنافذ عالية الترقيم لضمان سلامة حركة المرور المبنية على TCP؛ مما يخلق ضعفاً يمكن استغلاله من قبل المستخدمين غير المخوَّلين.

أما جدار تفتيش الحالة فيقوم بالتشديد على قواعد حركة مرور TCP من خلال إنشاء دليل لاتصالات TCP الصادرة، كما هو مبين بالجدول رقم 11-2، ويخصص في هذا الدليل مُدخلاً لكل اتصال فعَّال، ثم يقوم هذا الجدار الناري بالسماح لرزم حركة المرور الواردة إلى المنافذ عالية الترقيم فقط التي تتوافق مع أحد مُدخلات هذا الدليل.

الجدول 11-2: مثال لدليل حالة الاتصال لجدار تفتيش الحالة.

عنوان المصدر	منفذ المصدر	عنوان الوجهة	منفذ الوجهة	حالة الاتصال
192.168.1.100	1030	210.9.88.29	80	مؤسَّس
192.168.1.102	1031	216.32.42.123	80	مؤسَّس
192.168.1.101	1033	173.66.32.122	25	مؤسَّس
192.168.1.106	1035	177.231.32.12	79	مؤسَّس
223.43.21.231	1990	192.168.1.6	80	مؤسَّس
219.22.123.32	2112	192.168.1.6	80	مؤسَّس
210.99.212.18	3321	192.168.1.6	80	مؤسَّس
24.102.32.23	1025	192.168.1.6	80	مؤسَّس
223.212.212	1046	192.168.1.6	80	مؤسَّس

❖ بوَّابة مستوى التطبيق (Application-Level Gateway):

تعمل بوَّابة مستوى التطبيق – وتسمَّى أيضاً بالخادم المفوّض (proxy server) – كمُرَحِّل لحركة المرور على مستوى التطبيقات (الشكل 11-1 (b)). يقوم المستخدم بالاتصال بالبوَّابة باستخدام أحد تطبيقات TCP/IP، مثل Telnet أو بروتوكول نقل الملفات (FTP). تطلب البوَّابة من المستخدم اسم المضيف البعيد المطلوب الوصول إليه.

عندما يستجيب المستخدم ويوفر هويَّة المستخدم الصالحة ومعلومات التوثيق المطلوبة، تتصل البوَّابة بالتطبيق الذي يكون على المضيف البعيد وتُرَحِّل قطع بيانات TCP التي تتضمن بيانات التطبيق بين الطرفين. إذا لم تنفِّذ البوَّابة كود المفوّض لتطبيق معين، فلن تتوفر هذه الخدمة ولن يُسمح بتمريرها عبر جدار الحماية. فضلاً عن ذلك، يمكن ضبط البوَّابة لدعم سمات محدّدة فقط لتطبيق ما يعتبرها مدير الشبكة مقبولة مع رفض جميع السمات الأخرى.

غالباً ما تكون بوابات مستوى التطبيق أكثر أمناً من مرشحات الرزم؛ فبدلاً من محاولة التعامل مع كثير من التوليفات الممكنة التي يتم السماح لها أو منعها على مستوى TCP أو IP، تقوم بوَّابة مستوى التطبيق بفحص القليل فقط من التطبيقات المسموح بها. هذا بالإضافة إلى أنه من السهل تسجيل ومراجعة كافة أنواع حركة المرور الواردة على مستوى التطبيق.

العيب الرئيس لهذا النوع من البوابات هو المعالجة الإضافية التي تُجرَى لكل اتصال. وفعلياً يمكن اعتبار أن هناك اتصالين منقسمين بين المستخدمين الطرفيين؛ تكون فيهما البوَّابة هي نقطة الانقسام، حيث يجب على البوَّابة أن تفحص وتمرر حركة المرور في كلا الاتجاهين.

❖ بوَّابة مستوى الدائرة (Circuit-Level Gateway)

هناك نوع ثالث من الجدران النارية هو بوَّابة مستوى الدائرة (الشكل 11-1 (c)). يمكن أن يكون هذا النوع نظاماً بذاته أو أن يكون وظيفة خاصة يتم تشغيلها لصالح تطبيقات معينة من قِبَل بوَّابة مستوى التطبيقات. لا تسمح بوَّابة مستوى الدائرة بإجراء اتصال TCP من طرف لطرف؛ بل تُنشئ اتصالي TCP: واحداً بينها وبين مستخدم TCP لمضيف داخلي، وآخر بينها وبين مستخدم TCP لمضيف خارجي. وبمجرد إنشاء الاتصالين، تقوم البوَّابة عادةً بنقل قطع بيانات TCP من وصلة إلى أخرى دون فحص محتوياتها. هنا تكون وظيفة الأمن عبارة عن تحديد الاتصالات التي يمكن السماح بها.

في الاستخدامات المعتادة لبوابات مستوى الدائرة يثق مدير النظام (system administrator) بالمستخدمين الداخليين. هنا يمكن تهيئة البوَّابة لدعم الاتصالات الواردة على مستوى التطبيقات أو خدمات المُفَوِّض، بالإضافة إلى دعم الوظائف على مستوى الدوائر للاتصالات الصادرة. في هذا الوضع، قد تتكبد البوَّابة عبئاً إضافياً للتفتيش عن أية عمليات محظورة قد توجد ببيانات التطبيقات الواردة، أما في حالة البيانات الصادرة فلا تتكبد البوَّابة شيئاً من هذا العبء.

أحد الأمثلة العملية على بوابات مستوى الدائرة هو حزمة بروتوكول SOCKS [KOBL92]؛ وقد تم توصيف الإصدار 5 من SOCKS في طلب التعليقات رقم 1928 الذي يعرِّف SOCKS كما يأتي:

> "البروتوكول الموصوف هنا مصممٌ لتوفير إطار عمل لتطبيقات الزبون/الخادم في مجالي TCP وUDP لاستخدام خدمات الجدران النارية بسهولة وبشكلٍ آمن، وهذا البروتوكول من حيث المفهوم هو "طبقة حشوة" بين طبقة التطبيق وطبقة النقل، وعلى هذا فإنه لا يوفر خدمات البوَّابة لطبقة الشبكة، كتمرير رسائل ICMP."

تتألَّف حزمة SOCKS من العناصر الآتية:

- خادم SOCKS: يعمل على جدار ناري بنظام تشغيل يونيكس.
- مكتبة زبون SOCKS: تعمل على المضيفات الداخلية المحميَّة من قِبَل الجدار الناري.
- إصدارات تدعم SOCKS لكثير من برامج الزبون القياسية مثل بروتوكول نقل الملفات (FTP) وTELNET. وعادةً ما تتضمن تنفيذات SOCKS إعادة ترجمة (recompilation) أو إعادة ربط (relinking) لتطبيقات الزبون المبنية على TCP لكي تستخدم إجراءات التغليف (encapsulation) المناسبة بمكتبة SOCKS.

عندما يرغب زبون TCP في تأسيس اتصال مع كائن معين يمكن الوصول إليه فقط عبر جدار الحماية ، لا بد من فتح اتصال TCP إلى منفذ SOCKS مناسب على نظام خادم SOCKS. تقع خدمة SOCKS على منفذ TCP رقم 1080. إذا نجح طلب الاتصال هذا، فإن الزبون يتفاوض على أسلوب التوثيق الذي سيتم استخدامه، والذي على أساسه تتم عملية التوثيق، ومن ثمَّ يُرسِل الزبون طلب الترحيل. ويقوم خادم SOCKS بتقييم الطلب وبناءً عليه يُنشئ وصلة مناسبة أو يُهمِل الطلب. ومن الجدير بالذكر أن التعامل مع تبادلات UDP يتم بطريقة مماثلة؛ حيث يتم فتح اتصال TCP لتوثيق المستخدم من أجل إرسال قطع بيانات UDP واستقبالها، ومن ثم يتم تمريرها طالما ما زال اتصال TCP مفتوحاً.

❖ مضيف المَعقِل (Bastion Host)

مضيف المَعقِل هو نظام محدّد من قبل مسئول الجدار الناري كنقطة حرجة بشدة في أمن الشبكة. وعادةً ما يعمل مضيف المَعقِل بمثابة منصَّة لبوَّابة على مستوى الدائرة، أو على مستوى التطبيق. ومن السمات الشائعة لمضيف المَعقِل ما يأتي:

- يقوم مضيف المَعقِل بتنفيذ نسخة آمنة من نظام التشغيل بواسطة مكوِّناته المادية (hardware)، مما يجعل هذا النظام نظاماً موثوقاً.
- يتم فقط تنصيب الخدمات التي يعتبرها مدير الشبكة أساسية على مضيف المَعقِل. وتشمل هذه الخدمات تطبيقات المفوَّض مثل TELNET، وDNS، وFTP، وSMTP، وتوثيق المستخدم.
- قد يطلب مضيف المَعقِل توثيقاً إضافياً قبل أن يسمح للمستخدم بالوصول إلى خدمات المفوَّض. بالإضافة إلى ذلك، قد تطلب كل خدمة من خدمات المفوَّض توثيقاً خاصاً بها قبل السماح للمستخدم بالنفاذ.
- يتم ضبط كل مفوَّض لدعم مجموعة محدودة من الأوامر القياسية للتطبيقات.

- يتم ضبط كل مفوَّض بحيث يسمح بالوصول فقط إلى مجموعة محدّدة من الأنظمة المضيفة؛ وهذا يعني أن المجموعة المحدودة للأوامر المسموح بها يمكن تطبيقها فقط على هذه المجموعة المحدّدة من الأنظمة على الشبكة المحميَّة.
- يحتفظ كل مفوَّض بمعلومات مراجعة تفصيلية عن طريق تسجيل كل حركة مرور وكل اتصال ومدته. إن تسجيل مثل تلك المعلومات ضروري لاكتشاف هجمات الدخلاء وإنهائها.
- كل وحدة مفوَّض هي عبارة عن حزمة صغيرة من البرامج المصمَّمة خصيصاً لأمن الشبكة. وبسبب بساطتها النسبية، فمن السهل التحقق من العيوب الأمنية لهذه الوحدات. فمثلاً يحتوي تطبيق بريد يونيكس الإلكتروني عادةً على أكثر من 20,000 سطر من التعليمات البرمجية، في حين أن مفوَّض البريد قد يحتوي على أقل من 1000 سطر.
- يكون كل مفوَّض مستقلاً عن المفوَّضين الآخرين على مضيف المَعقِل. فإذا حدثت مشكلةٌ ما في تشغيل أي مفوَّض، أو إذا تم اكتشاف ضعف أمني به، فمن الممكن إلغاء تثبيته دون أن يؤثر ذلك على تشغيل تطبيقات المفوَّضين الآخرين. وأيضاً إذا كانت مجموعة المستخدمين تتطلب دعماً لخدمة جديدة، فإنه يمكن لمسئول الشبكة تثبيت المفوَّض المطلوب بسهولة على مضيف المَعقِل.
- عموماً لا يحتاج المفوَّض في عمله للنفاذ إلى القرص الصلب إلا عند قراءة ملف التهيئة (الإعدادات) الأوَّليّ (initial configuration file)؛ وهذا مما يصعِّب على الدخلاء تثبيت برامج أحصنة طروادة لالتقاط الرزم أو غيرها من الملفات الخطرة على مضيف المَعقِل.
- كل مفوَّض يعمل كمستخدم بدون امتيازات (صلاحيات privileges) في دليل خاص ومؤمَّن على مضيف المَعقِل.

3-1-11 تهيئة الجدار الناري

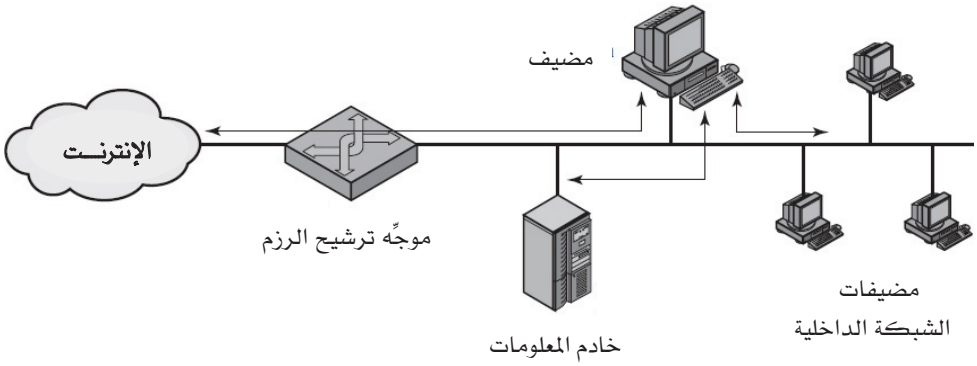
يمكن استخدام تهيئة بسيطة تتكون من نظام واحد، مثل موجِّه واحد لترشيح الرزم أو بوَّابة واحدة (كما بالشكل 11-1)، كما أنه من الممكن أيضاً استخدام تهيئات أكثر تعقيداً بل وأكثر شيوعاً. يوضِّح الشكل 11-2 ثلاثاً من تهيئات الجدار الناري الشائعة التي سنقوم بدراسة كلٍّ منها.

في جدار حماية المضيف المنتقى بتشكيل المَعقِل أحادي المقر (screened host firewall, single-homed bastion) (الشكل 11-2 (a))، يتألف الجدار الناري من نظامين: موجِّه ترشيح رزم، ومضيف المَعقِل. عادةً يتم تهيئة الموجِّه على النحو الآتي:

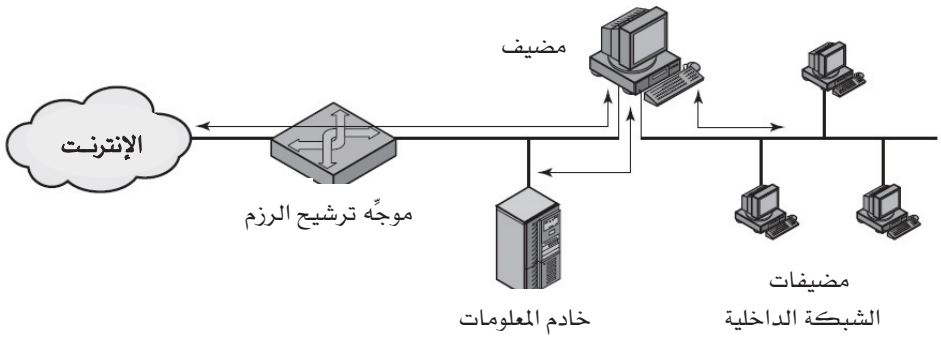
1. بالنسبة لحركة المرور من الإنترنت: يتم السماح لرزم بروتوكول الإنترنت المتجهة إلى مضيف المَعقِل فقط.
2. بالنسبة لحركة المرور من الشبكة الداخلية: يتم السماح لرزم بروتوكول الإنترنت القادمة من مضيف المَعقِل فقط.

يُنفِّذ مضيف المَعقِل مهام التوثيق ووظائف المفوَّض. تمتلك هذه التهيئة قدراً أكبر من الأمن مقارنةً مع مجرد استعمال موجِّه لترشيح الرزم أو مجرد بوَّابة على مستوى التطبيقات، وذلك لسببين:

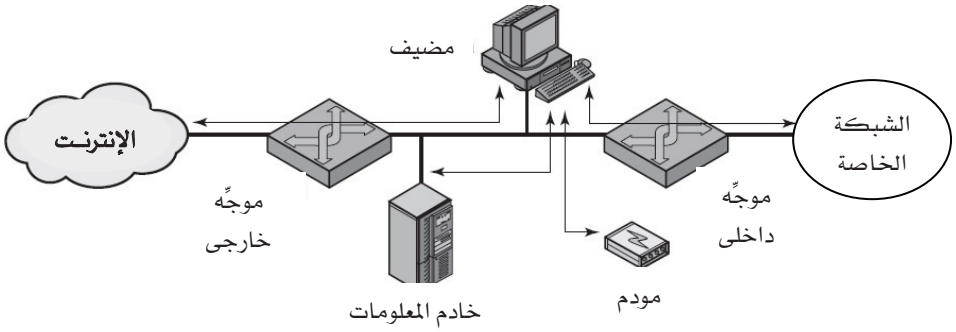
- قيام هذه التهيئة بعمل ترشيحين أحدهما على مستوى الرزمة والآخر على مستوى التطبيق، مما يسمح بقدر كبير من المرونة في تحديد السياسة الأمنية.
- أن على الدخلاء عموماً اختراق نظامين منفصلين قبل أن يتعرض أمن الشبكة الداخلية للخطر.



(a) جدار حماية المضيف المنتقى بتشكيل المعقل أحادي المقر.



(b) جدار حماية المضيف المنتقى بتشكيل المعقل ثنائي المقر



(c) تشكيل جدار حماية الشبكة الفرعية المنتقاة

الشكل 11-2: تهيئات الجدران النارية.

يمتاز هذا الشكل أيضاً بالمرونة في توفير الوصول المباشر إلى شبكة الإنترنت، فعلى سبيل المثال قد تشتمل الشبكة الداخلية على خادم معلومات عامة – مثل خادم الويب – والذي لا يتطلب قدراً عالياً من الأمن. وفي هذه الحالة، يمكن تهيئة جهاز التوجيه ليسمح بحركة مرور مباشرة بين خادم المعلومات والإنترنت.

في تشكيل المَعقِل أحادي المقر الموصوف للتوّ، إذا أصبح أمن موجِّه ترشيح الرزم مخترقاً أو منعدماً، عندها يمكن أن تتدفق حركة المرور مباشرةً خلال الموجِّه الواقع بين الإنترنت والأنظمة المضيفة الأخرى على الشبكة الخاصة. أما جدار حماية المضيف المنتقى بتشكيل المَعقِل ثنائي المقر (screened host firewall, dual-homed bastion) فيقوم بمنع مثل هذا الاختراق الأمني بشكلٍ مادي (الشكل 11-2 (b)). تتوفر في هذا التشكيل نفس مزايا الطبقات الأمنية المزدوجة المتوفرة في التشكيل السابق. ومرةً أخرى، فإنه يمكن السماح لخادم المعلومات أو الأنظمة المضيفة الأخرى بالاتصال المباشر مع الموجِّه إذا كان هذا متوافقاً مع السياسة الأمنية.

يُعد تشكيل جدار حماية الشبكة الفرعية المنتقاة (screened subnet firewall) كالمُبيَّن بالشكل 11-2 (c) الأكثر أماناً مقارنةً بجميع النظم السابق عرضها. ففي هذا التشكيل، يتم استخدام موجِّهين لترشيح الرزم، يقع أحدهما بين مضيف المَعقِل والإنترنت ويقع الآخر بين مضيف المَعقِل والشبكة الداخلية. يُنشئ هذا التشكيل شبكة فرعية معزولة قد تتكون ببساطة من مضيف المَعقِل فقط كما قد تضم أيضاً واحداً أو أكثر من خادمات المعلومات وأجهزة المودم من أجل توفير قدرة الاتصال الهاتفي (dial-in). عادةً ما يمكن لشبكة الإنترنت والشبكة الداخلية الوصول إلى الأنظمة المضيفة على الشبكة الفرعية المنتقاة، لكن يتم حظر حركة المرور عبر تلك الشبكة الفرعية. يوفر هذا التشكيل عدداً من المزايا:

- يوجد الآن ثلاثة مستويات من الدفاع تهدف إلى إحباط الدخلاء.

- يقوم الموجِّه الخارجي بالإعلان (على شبكة الإنترنت) فقط عن وجود شبكة فرعية منتقاة، ولذلك تصبح الشبكة الداخلية غير مرئية بالنسبة لشبكة الإنترنت.
- وبالمثل، فإن الموجِّه الداخلي يعلن (على الشبكة الداخلية) فقط عن وجود شبكة فرعية منتقاة، ولذلك لا يمكن للنظم على الشبكة الداخلية بناء مسارات مباشرة إلى شبكة الإنترنت.

11-2 الأنظمة الموثوقة

إحدى الطرق لتعزيز قدرة النظام على الدفاع ضد الدخلاء والبرمجيات الخبيثة هي تنفيذ تقنية النظم الموثوقة. ويعطي هذا الجزء لمحةً موجزةً عن هذا الموضوع. وسنبدأ بالنظر في بعض المفاهيم الأساسية للتحكم في الوصول إلى البيانات.

11-2-1 التحكم في الوصول إلى البيانات

بعد الوصول الناجح للشبكة، يحق للمستخدم النفاذ إلى أحد الأنظمة المضيفة والتطبيقات (أو إلى مجموعة منها). وعادةً لا يُعد هذا كافياً لنظام يتضمن بيانات حساسة في قاعدة بياناته. ويمكن أن يتم تعريف المستخدم للنظام من خلال إجراءات التحكم في الوصول للنظام وذلك بأن يحتفظ النظام بملف تعريف لكل مستخدم يحدد العمليات المسموح له بها والملفات التي يمكنه الوصول إليها، ويقوم نظام التشغيل بإعمال قواعد تستند إلى ذلك الملف. ومع ذلك يجب على نظام إدارة قواعد البيانات التحكم في الوصول إلى سجلات محدّدة أو حتى إلى أجزاء من بعض السجلات. فعلى سبيل المثال، قد يكون من المتاح لأي فرد في الإدارة الحصول على قائمة موظفي الشركة، لكن يُسمح فقط لأفراد معدودين بالوصول إلى معلومات الرواتب. وهذه المسألة هي أكثر من مجرد مستوى آخر من التفاصيل. ففي حين أن نظام التشغيل قد يمنح الإذن لمستخدمٍ ما بالوصول إلى ملف أو استخدام تطبيق معين بدون الحاجة لمزيد من التدقيق الأمني بعد ذلك، فإن نظام إدارة قواعد البيانات لا بُدّ وأن يقرر السماح من عدمه بشأن كل محاولة للحصول على بيانات جديدة.

وقرار كهذا لا يعتمد فقط على هويَّة المستخدم ولكن أيضاً على أجزاء البيانات التي يريد الحصول عليها، بل وحتى على المعلومات التي سبق أن حصل عليها.

وهناك نموذج عام للتحكم في الوصول كالنظام المستعمل من قبل نظم إدارة قواعد البيانات والملفات ألا وهو مصفوفة الوصول (access matrix) (انظر الشكل 3-11 (a))، والعناصر الأساسية لهذا النموذج هي كما يأتي:

- الفاعل (Subject): وهو كيانٌ قادرٌ على الوصول إلى الأشياء. وعموماً، فإن مفهوم الفاعل يكافئ مفهوم العملية (process). وفي الواقع أي مستخدم أو تطبيق لا يصل إلى كائنٍ ما إلا من خلال العملية التي تمثل ذلك المستخدم أو التطبيق.
- الكائن (Object): أي شيء يتم التحكم في الوصول أو النفاذ إليه؛ كالملفات أو أجزاء منها ، والبرامج، وقطاعات الذاكرة.
- حق الوصول (Access right): الطريقة التي يمكن للفاعل (subject) النفاذ بها لكائن ما (object)؛ كالقراءة (read)، والكتابة (write)، والتنفيذ (execute).

يمثل أحد محوري المصفوفة الفاعل الذي قد يحاول الوصول إلى البيانات. ويشمل ذلك عادةً، آحاد المستخدمين أو مجموعاتهم، بالرغم من إمكانية فرض عملية التحكم في الوصول على المحطات الطرفية والأنظمة المضيفة أو التطبيقات بدلاً من أو بالإضافة إلى المستخدمين. أما المحور الآخر فيعدِّد الكائنات التي يمكن الوصول إليها. وبتفصيل أكثر، قد تكون تلك الكائنات حقول بيانات فردية، أو قد تكون أيضاً مجموعات أكبر مثل السجلات (records) والملفات أو حتى قواعد بيانات بأكملها. يبين كل مُدخل في المصفوفة حقوق وصول فاعل معين لكائنٍ ما.

	البرنامج 1	...	القطعة A	القطعة B
العملية 1	قراءة وتنفيذ		قراءة وكتابة	
العملية 2				قراءة
. . .				

(a) مصفوفة الوصول

قائمة التحكم في الوصول للبرنامج 1:
العملية 1 (قراءة، تنفيذ)
قائمة التحكم في الوصول للقطعةA:
العملية 1 (قراءة، تنفيذ)
قائمة التحكم في الوصول للقطعة B:
العملية 2 (قراءة)

(b) قوائم التحكم في الوصول

قائمة الصلاحيات للعملية 1:
البرنامج 1 (قراءة، تنفيذ)
القطعة A (قراءة، كتابة)
قائمة الصلاحيات للعملية 2:
القطعة B (قراءة)

(c) قائمة الصلاحيات

الشكل 11-3: هياكل التحكم في الوصول.

عادةً ما تكون معظم مُدخلات مصفوفة الوصول خاوية (مصفوفة متناثرة sparse matrix)، ويتم تنفيذها عملياً بتقسيمها بأحد طريقتين. فيمكن تقسيم المصفوفة بحسب الأعمدة مما يولد عدداً من قوائم التحكم في الوصول (access control lists (ACLs)) (الشكل 11-3 (b)). ومن ثَمَّ، يكون لكل كائن قائمة خاصة به تحدد لكل مستخدم الحقوق المسموح له بها. وقد تحتوي قائمة التحكم هذه على مُدخل عام أو افتراضي، مما يسمح للمستخدمين غير المذكورين صراحةً بمجموعة حقوق افتراضية (default). ويمكن أن تشمل عناصر هذه القائمة آحاد المستخدمين أو مجموعاتهم.

كذلك يمكن تقسيم المصفوفة بحسب الصفوف مما يولد عدداً من تذاكر الإمكانات (capability tickets) (الشكل 11-3 (c)). وتحدد تذكرة الإمكانات الكائنات والعمليات المخولة لكل مستخدم والذي يكون لديه عدد محدّد من التذاكر، وقد يُخوَّل للمستخدم إقراض هذه التذاكر أو إعطائها لآخرين. ولأن التذاكر قد تكون متناثرة في جميع أنحاء النظام، فإنها تمثل مشكلة أمنية أكبر من مشاكل قوائم التحكم في الوصول؛ فبالتحديد يجب أن تكون التذكرة غير قابلة للتزوير. وإحدى الطرق لتحقيق ذلك أن يمسك نظام التشغيل بجميع التذاكر نيابةً عن المستخدمين وأن يتم تخزينها في منطقة من الذاكرة لا يحق للمستخدمين الوصول إليها.

11-2-2 مفهوم النظم الموثوقة

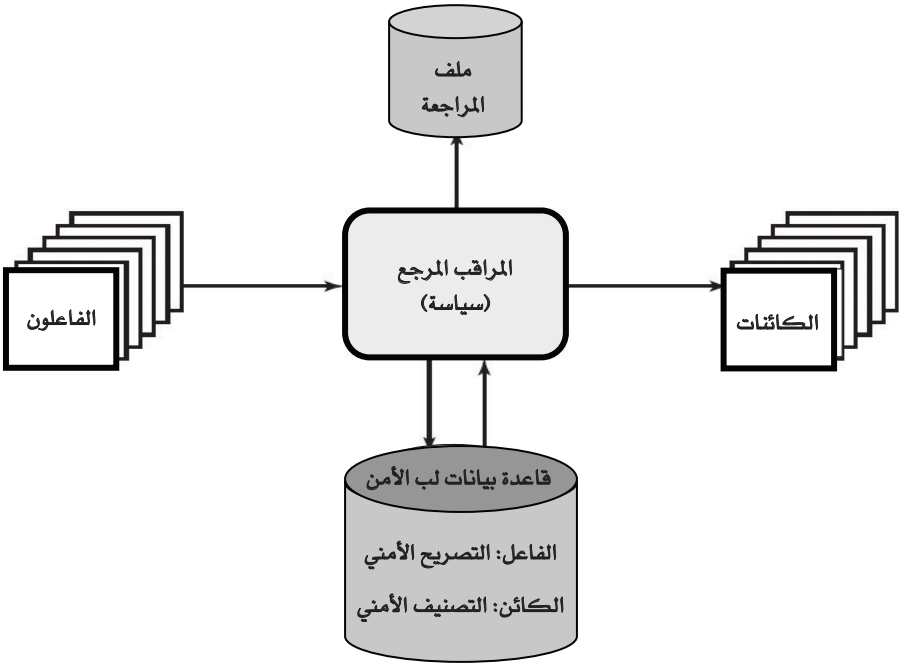
معظم ما ناقشناه حتى الآن معنيٌّ بحماية رسالة معينة أو عنصرٍ ما من هجمات سلبية (passive attacks) أو نشطة (active) من قِبَل مُستخدِم معين. وهناك متطلبٌ مختلفٌ نوعاً ما ولكنه مطلوبٌ على نطاق واسع ألا وهو حماية البيانات أو الموارد بدرجات أمنية متفاوتة. وهذا الأمر شائع في المجال العسكري، حيث يتم تصنيف المعلومات على أنها معلومات متاحة (U)، أو خاصة (C)، أو سرية (S)، أو سرية للغاية (TS)، أو ما هو أشد. وينطبق هذا المفهوم أيضاً في مجالات أخرى، والتي يمكن تنظيم المعلومات فيها إلى فئات إجمالية حيث يُمنح المستخدمون تصاريح

وصول إلى فئات معينة من تلك البيانات. فعلى سبيل المثال، يمكن تخصيص المستوى الأعلى من الأمن لوثائق وبيانات التخطيط الاستراتيجي للشركات، والتي يمكن الوصول إليها فقط من قِبَل مسئولي الشركة ومساعديهم؛ ويلي ذلك البيانات المالية وشئون الموظفين الحساسة، والتي يمكن الوصول إليها فقط من قبل إدارة شئون الموظفين ومسئولي الشركة، وهلم جرا.

يشار إلى المتطلبات التي يُراد فيها تصنيف البيانات إلى فئات أو مستويات متعددة بالأمن متعدد المستويات (multilevel security). والنص العام لمتطلب الأمن متعدد المستويات هو أن الفاعل الموجود بمستوى عالٍ لا يصح له نقل معلومات إلى كائن بمستوى أدنى أو مستوى غير مماثل إلا إذا كان تدفق المعلومات هذا يعكس بدقة إرادة مستخدم مخوّل له. أما لأغراض التنفيذ، فيمكن صياغة هذا المتطلب كشرطين ينبغي لنظام الأمن متعدد المستويات الالتزام بهما، وهما:

- ممنوع القراءة لأعلى (No read up): يمكن للفاعل فقط قراءة كائن ذي مستوى أمن مساوٍ أو أدنى، ويشار إلى ذلك بخاصية الأمن البسيط (Simple Security Property).
- ممنوع الكتابة لأسفل (No write down): يمكن للفاعل فقط الكتابة في كائن ذي مستوى أمن مساوٍ أو أعلى. ويشار إلى ذلك بخاصية * (خاصية النجمة)[1].

[1] النجمة '*' لا تعني شيئاً محدداً. حيث لم يأت أحد باسم مناسب لتلك الخاصية حتى كتابة أول تقرير لهذا النموذج. فاستُخدِمت النجمة في المسودة حتى يمكن تبديلها بسرعة بمجرد الاتفاق على اسمٍ للخاصية. وحيث أنه لم يتم الاتفاق على اسم ما فقد تم طبع التقرير كما هو بدون تبديل النجمة.



الشكل 11-4: المراقب المَرجع.

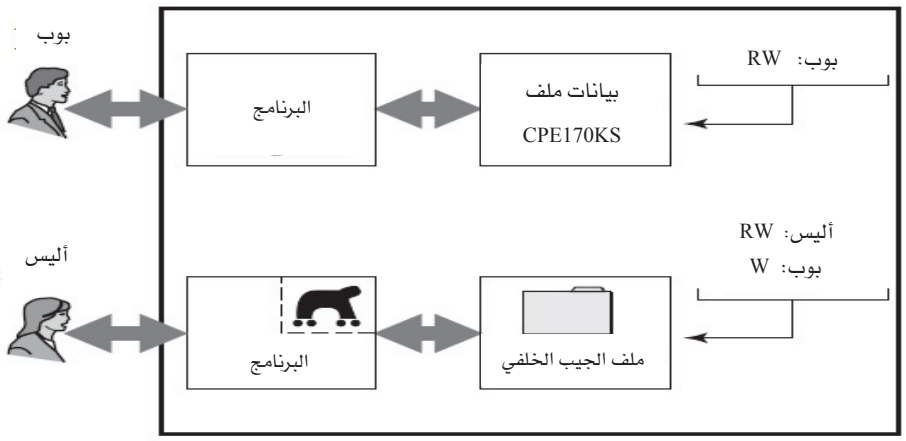
وهاتان القاعدتان إذا طبقتا على النحو الصحيح، فإنهما توفران الأمن متعدد المستويات. وبالنسبة لنظم معالجة البيانات، فإن النهج المتّبع الذي كان هدفاً لكثير من البحث والتطوير يستند إلى مفهوم المراقب المَرجع (reference monitor) كما هو موضح بالشكل 11-4. والمراقب المَرجع هو جزء بالمكوِّنات المادية ونظام التشغيل بجهاز الحاسب يقوم بالتحكم وتنظيم وصول فاعل ما إلى الكائنات على أساس المتغيرات الأمنية للفاعل والكائنات. ويمتلك المراقب المَرجع حق الوصول إلى الملف المعروف باسم لب قاعدة بيانات الأمن (security kernel database) والذي يسرد امتيازات الوصول (التصريح الأمني security clearance) لكل فاعل، وسمات الحماية (مستوى التصنيف classification level) لكل كائن. يفرض المراقب المَرجع

القواعد الأمنية (ممنوع القراءة لأعلى وممنوع الكتابة لأسفل)، كما يمتلك الخصائص الآتية:

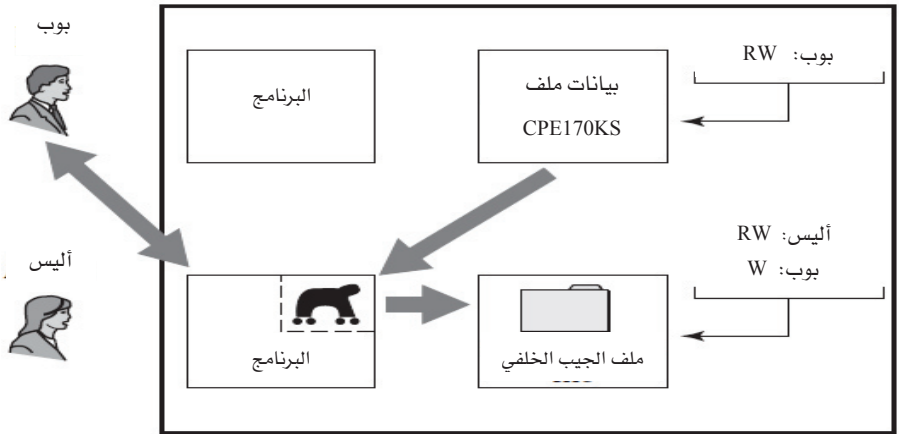
- الوساطة الكاملة: يتم تطبيق القواعد الأمنية على كل وصول (نفاذ)، وليس فقط – على سبيل المثال – عند فتح الملف.
- العزل: يتم حماية المراقب المَرجِع وقاعدة البيانات من أي تعديل غير مصرّح به.
- التحقق: يجب توفر إمكانية إثبات صحة المراقب المَرجِع؛ أي أنه يجب أن يكون من الممكن رياضياً إثبات أن المراقب المَرجِع يفرض القواعد الأمنية ويوفر الوساطة الكاملة والعزل.

وهذه شروط جامدة لا مرونة فيها. فشرط الوساطة الكاملة مثلاً يعني أن الوساطة لابد أن تتحقق في أي حالة وصول إلى البيانات سواءً كانت داخل الذاكرة الرئيسة أو على القرص الصلب أو غيره. وتنفيذ عمل مثل هذا كبرمجيات فقط يُشكِّل عبئاً كبيراً جداً في الأداء ليكون عملياً، ولهذا ينبغي تنفيذه على الأقل جزئياً كمكوِّن مادي (hardware). كما يعني شرط العزل أنه يجب أن لا يكون من الممكن للمهاجم، مهما كان ذكياً، أن يغير منطق المراقب المَرجِع أو محتويات لب قاعدة بيانات الأمن. وأخيراً، فشرط الإثبات الرياضي يُعدُّ تحدياً هائلاً لنظامٍ معقد مثل الحاسب متعدد الغرض (general-purpose computer). فالنظام الذي يمكنه أن يقدم مثل هذا التحقق يشار إليه بأنه نظام موثوق.

وثمة عنصر أخير مبين في الشكل 11-4 ألا وهو ملف المراجعة (audit file)؛ حيث يتم تخزين كل الأحداث الأمنية المهمة، وكل الخروقات الأمنية المكتشفة، وكل التغييرات في لب قاعدة بيانات الأمن المصرّح بها، في هذا الملف.

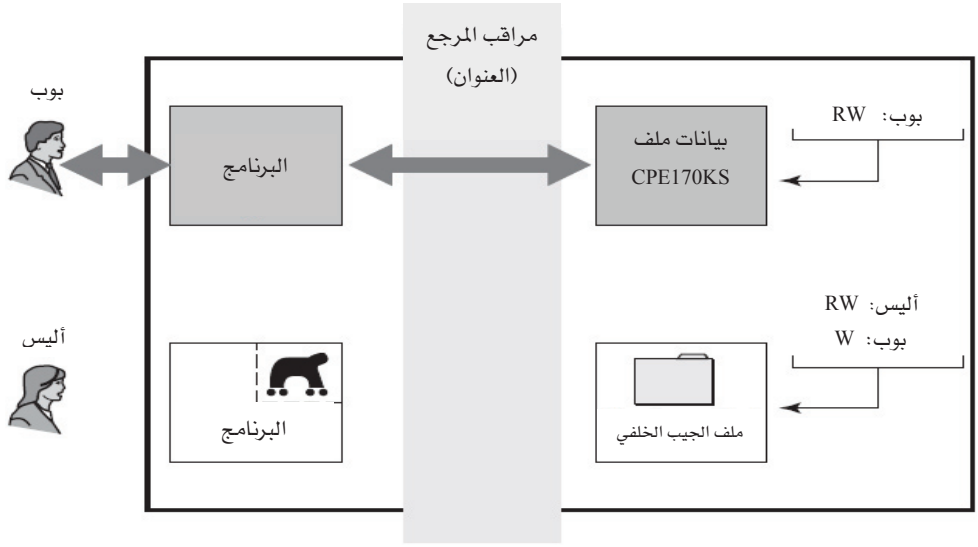


(a)

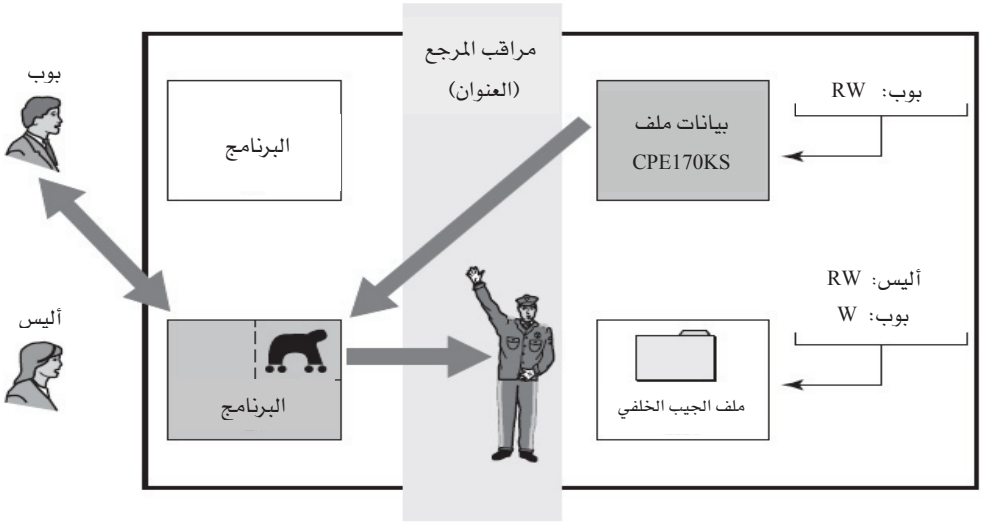


(b)

الشكل 11-5: حصان طروادة و نظام التشغيل الآمن.



(c)



(d)

تابع الشكل 11-5: حصان طروادة و نظام التشغيل الآمن.

أنشأت وزارة الدفاع الامريكية في عام 1981 – في محاولة لتلبية احتياجاتها الخاصة، وكخدمة عامة – مركزاً لأمن الحاسب داخل وكالة الأمن القومي (NSA) وذلك بهدف تشجيع توافر نظم الحاسب الموثوقة على نطاق واسع. وتحقق هذا الهدف عن طريق "برنامج تقييم المنتج التجاري" الخاص بالمركز. وبشكلٍ أساسي يحاول المركز تقييم المنتجات المتاحة تجارياً وتصنيفها من حيث استيفائها للمتطلبات الأمنية سالفة الذكر. حيث يُصنَّف المركز المنتجات المقيَّمة وفقاً للسمات الأمنية المتوفرة بها. وهذه التقييمات ضرورية لمشتريات وزارة الدفاع ولكن يتم نشرها وإتاحتها للجميع مجاناً. ولذلك فهي تعمل بمثابة إرشادات للعملاء الراغبين في شراء مثل تلك المنتجات المتاحة تجارياً.

11-2-3 الدفاع ضد حصان طروادة

هناك طريقة واحدة للتأمين ضد هجمات حصان طروادة ألا وهي استخدام نظام تشغيل آمن وموثوق. ويوضِّح الشكل 11-5 مثالاً على ذلك. في هذه الحالة، يُستخدم حصان طروادة للالتفاف حول آلية الأمن المعيارية والتي تستخدمها معظم أنظمة إدارة الملفات ونظم التشغيل: قائمة التحكم في الوصول. ففي هذا المثال، يتفاعل المستخدم بوب من خلال برنامجٍ ما مع ملف بيانات يحتوي على سلسلة الحروف الحساسة "CPE170KS". ثم يقوم المستخدم بوب بإنشاء ملف مع حقوق وصول "قراءة/كتابة" يسمح بها فقط للبرامج العاملة لحسابه: بمعنى أن العمليات (processes) التي يملكها بوب هي الوحيدة التي يمكنها الوصول إلى الملف.

يبدأ هجوم حصان طروادة عندما تستطيع مستخدمة مُعادية – تدعى أليس – الوصول بطريقة شرعية إلى النظام وتقوم بتثبيت برنامج حصان طروادة مع ملف خاص لاستخدامه في الهجوم كـ "جيب خلفي". تعطي أليس حق "قراءة/كتابة" لنفسها لهذا الملف، كما تعطي بوب حق "الكتابة" فقط (الشكل 11-5 (a)). وتدفع أليس بوب لاستحضار برنامج حصان طروادة – ربما عن طريق الإعلان عنه كأداة مفيدة – وعندما يكتشف البرنامج أنه يجري تنفيذه حالياً من قبل بوب، فإنه يقوم

بقراءة سلسلة الحروف الحساسة "CPE170KS" من ملف بوب، ونسخها إلى ملف جيب أليس الخلفي (الشكل 11-5 (b)). وتتوافق كلٌّ من عمليات القراءة والكتابة مع القيود التي تفرضها قوائم التحكم في الوصول. وحينئذٍ تستطيع أليس الوصول إلى ملف الجيب الخلفي في وقتٍ لاحقٍ لقراءة قيمة السلسلة.

إذا ما تم استخدام نظام تشغيل آمن في هذا السيناريو (الشكل 11-5 (c))، حيث تحدد مستويات الأمن عند الوصول إلى النظام على أساس بعض المعايير، مثل المحطة الطرفية التي تم الوصول منها إلى جهاز الحاسب، ومثل المستخدم كما تحدده كلمة المرور وهويَّة المستخدم. وفي هذا المثال هناك نوعان من مستويات الأمن: المستوى الحساس والمستوى العام، والمستوى الحساس هو الأعلى. فالعمليات (processes) التي يملكها بوب وملف بياناته خصِّص لها المستوى "الحساس" كمستوى للأمن. أما ملفات أليس وعملياتها فقد خصص لها المستوى "العام" كمستوى للأمن. عندما يستحضر بوب برنامج حصان طروادة (الشكل 11-5 (d))، فإن هذا البرنامج يحصل على نفس مستوى أمن بوب. ولذلك يصبح قادراً، تحت خاصية الأمن البسيط، على مراقبة سلسلة الحروف الحساسة. وعند محاولة البرنامج تخزين السلسلة في ملف عام (ملف الجيب الخلفي)، يُعدُّ ذلك انتهاكاً لخاصية *، ولذا يتم منع المحاولة من قِبَل المراقب المَرجِع، ويتم رفض محاولة الكتابة في ملف الجيب الخلفي على الرغم من أن قائمة التحكم في الوصول سمحت بها؛ لأن السياسة الأمنية لها الأسبقية على آلية قائمة التحكم بالوصول.

11-3 المعايير المشتركة لتقييم أمن تقنية المعلومات

العمل الذي قامت به وكالة الأمن القومي وغيرها من الوكالات الحكومية الأمريكية لتعريف صفات التقييم ومعاييره للنظم الموثوقة تم محاكاته بأعمال مماثلة في بلدان أخرى. والمعايير المشتركة (Common Criteria (CC)) لتقييم أمن تقنية المعلومات هي مبادرة دولية من قِبَل الهيئات المعنية بوضع المعايير في عدد من البلدان لتطوير معايير دولية، وتحديد المتطلبات الأمنية، ووضع معايير للتقييم.

11-3-1 المتطلبات

تُحدِّد المعايير المشتركة (CC) مجموعة عامة من المتطلبات الأمنية المتوقعة لاستخدامها في التقييم. ويشير مصطلح هدف التقييم (Target of Evaluation-TOE) إلى الجزء من المنتج أو النظام الخاضع للتقييم. وتصنَّف المتطلبات إلى نوعين:

- متطلبات وظيفية: تحدِّد السلوك الأمني المرجوّ. تعرِّف وثائق CC مجموعة من المكوِّنات الوظيفية الأمنية التي توفر وسيلة معيارية للتعبير عن المتطلبات الوظيفية للأمن لهدف التقييم (TOE).
- متطلبات التأمين: تشكِّل أساساً لاكتساب الثقة بأن المقاييس الأمنية المذكورة فعَّالة ومنفَّذة بشكلٍ صحيح. وتعرِّف وثائق CC مجموعة من مكوِّنات التأمين التي توفر طريقة موحدة للتعبير عن متطلبات التأمين لهدف التقييم (TOE).

الجدول 11-3: المتطلبات الوظيفية للمعايير المشتركة للأمن.

الصنف	الوصف
المراجعة	يتضمن إدراك المعلومات المتعلقة بالأنشطة الأمنية وتسجيلها وتخزينها وتحليلها. ويتم إنتاج سجلات المراجعة من قِبَل هذه الأنشطة، ويمكن دراستها لتحديد مدى أهميتها الأمنية
دعم التشفير	يُستخدَم عندما ينفَّذ هدف التقييم وظائف التشفير. وقد يُستخدَم، على سبيل المثال، لدعم الاتصالات ،وتحديد الهويَّة، والتوثيق ، أو فصل البيانات
الاتصالات	يوفر اثنين من العائلات المعنيّة بعدم التنصل من قِبَل المنشئ ومتلقي البيانات
حماية بيانات المستخدم	تحديد المتطلبات المتعلقة بحماية بيانات المستخدم ضمن هدف التقييم خلال الاستيراد، والتصدير، والتخزين، إضافة إلى السمات الأمنية المتصلة ببيانات المستخدم
التحقق والتوثيق	تضمن التحقق – دون أي لبس – من المستخدمين الموثقين والارتباط الصحيح للسمات الأمنية مع المستخدمين والموضوعات.
إدارة الأمن	تحدّد إدارة السمات الأمنية، والبيانات، والوظائف
الخصوصية	توفر للمستخدم الحماية ضد الانكشاف، وإساءة استخدام هويته من قِبَل المستخدمين الآخرين
حماية وظائف هدف التقييم الأمنية	تركّز على حماية بيانات TSF (وظائف هدف التقييم الأمنية TOE security functions)، وليس بيانات المستخدم. ويتعلق هذا الصنف بسلامة وإدارة آليات TSF والمعلومات.
استخدام المصادر	تدعم توافر المصادر اللازمة، مثل قابلية المعالجة وسعة التخزين. وتتضمن متطلبات تحمُّل الخطأ (fault tolerance)، والأولوية في تخصيص المصادر والخدمات.
وصول هدف التقييم	تحدّد المتطلبات الوظيفية، بالإضافة إلى تلك المحدّدة للتحقق والتوثيق، من أجل التحكم في إنشاء جلسة عمل مستخدم. وتعرض متطلبات توجيه وصول هدف التقييم، مثل أمور الحد من عدد ونطاق جلسات المستخدم، تاريخ الوصول وتعديل متغيرات الوصول.
المسارات/القنوات الموثوقة	تُعنى بمسارات الاتصالات الموثوقة بين المستخدمين والـ TSF وبين مجموعات من TSF.

يتم تقسيم كلٌّ من المتطلبات الوظيفية ومتطلبات التأمين إلى فئات. والفئة هي عبارة عن مجموعة من المتطلبات التي تشترك في رؤية مشتركة أو مقصد واحد. يُعرِّف الجدولان 11-3 و11-4 بصورة مختصرة الفئات الخاصة بالمتطلبات الوظيفية ومتطلبات التأمين. تحتوي كلٌّ من هذه الفئات على عدد من العائلات، وتضم كل عائلة متطلبات تشترك في الأهداف الأمنية ولكنها تختلف في شدتها أو صرامتها. فعلى سبيل المثال، تحتوي طبقة المراجعة على ست عائلات تتعامل مع الجوانب المختلفة لعملية المراجعة (مثل: توليد بيانات المراجعة، وتحليل بيانات المراجعة، وتخزين أحداث المراجعة). وتحتوي كل عائلة بدورها على واحد أو أكثر من المكوِّنات. يصف المكوّن مجموعة معينة من المتطلبات الأمنية ويمثل أصغر وحدة متطلبات أمنية يمكن اختيارها لإدراجها في الهياكل المعرفة في CC.

فعلى سبيل المثال، تشمل فئة دعم التشفير الخاصة بالمتطلبات الوظيفية عائلتين هما: عملية التشفير، وإدارة مفاتيح التشفير. وتحتوي عائلة إدارة مفاتيح التشفير على أربعة مكوِّنات تُستخدَم لتحديد ما يأتي: خوارزمية توليد المفاتيح مع طول المفتاح، وأسلوب توزيع المفاتيح، وأسلوب الوصول إلى المفاتيح، وأسلوب إتلاف المفاتيح.

يوجد لكل مكوّن معيار يُرجَع إليه لتحديد المتطلبات. فمثلاً عائلة عملية التشفير تتضمن مكوّن واحد يحدد الخوارزمية وطول المفتاح بناءً على المعيار المحدّد.

ويمكن جمع بعض المكوِّنات الوظيفية ومكوِّنات التأمين معاً في حزمة (package) واحدة يمكن إعادة استخدامها، وعادةً ما تكون تلك الحزم مفيدة في إنجاز الأهداف المرغوبة. ومن الأمثلة على تلك الحزم؛ المكوِّنات الوظيفية المطلوبة للأدوات التقديرية للتحكم في الوصول (Discretionary Access Controls).

الجدول 11-4: متطلبات ضمان المعايير المشتركة للأمن.

الفئة	الوصف
إدارة التهيئة	تتطلب المحافظة على سلامة هدف التقييم بدرجة كافية. وعلى وجه التحديد، توفر إدارة التهيئة الثقة بأن هدف التقييم والوثائق المستخدمة في عملية التقييم هي تلك التي أعدَّت للتوزيع.
التوصيل والتشغيل	يُعنى بالتدابير والإجراءات والمعايير الخاصة بتأمين عملية التسليم والتركيب والاستخدام العملي لهدف التقييم؛ وذلك لضمان أن الحماية الأمنية التي يوفرها هدف التقييم لا يتم تعرضها للخطأ خلال هذه الأحداث.
التطوير	يُعنى بضبط هدف التقييم من المواصفات المحدّدة في الهدف الأمني إلى التطبيق، وبترجمة المتطلبات الأمنية إلى مستويات التمثيل الدنيا.
وثائق الإرشاد	يُعنى بالاستخدام العملي الآمن لهدف التقييم من قِبَل المستخدمين والإداريين.
دعم دورة الحياة	يُعنى بدورة حياة هدف التقييم، ويشمل تعريف دورة الحياة، وأساليب وأمن تطور البيئة، ومعالجة نقاط الضعف المكتشفة من قِبَل مستخدمي هدف التقييم .
الإختبارات	يُعنى بالبرهنة على أن هدف التقييم يلتقي مع متطلباته الوظيفية. وتُعنى العائلات بمسائل الشمول، وعمق اختبار المُطَوِّر، ومتطلبات الاختبارات المستقلة.
تقييم نقاط الضعف	يُحدِّد متطلبات التعرف على نقاط الضعف القابلة للاستغلال، والتي قد تنتج عن عمليات الإنشاء أو التشغيل، أو الاستخدام غير الصحيح، أو التهيئة الخاطئة لهدف التقييم. والعائلات المحدّدة هنا هي المعنية بتحديد نقاط الضعف من خلال تحليل القنوات السرية، أو تحليل تهيئة هدف التقييم، أو اختبار قوة آليات الوظائف الأمنية، أو تحديد العيوب الناتجة خلال تطوير هدف التقييم. وتغطي العائلة الثانية التصنيف الأمني لعناصر هدف التقييم. وتشمل الثالثة والرابعة تحليل التغيرات للتأثيرات الأمنية، وتقديم أدلة على أن هذه الإجراءات يجري اتباعها. وهذه الفئة توفر لبنات البناء لتأسيس مخططات صيانة الضمان.
صيانة الضمان	يُحدِّد المتطلبات المراد تطبيقها بعد اعتماد هدف التقييم بناءً على المعايير المشتركة. تهدف هذه المتطلبات إلى ضمان أن هدف التقييم سيواصل الوفاء بأهدافه الأمنية مع دوام عمليات التغيير في هدف التقييم أو بيئته.

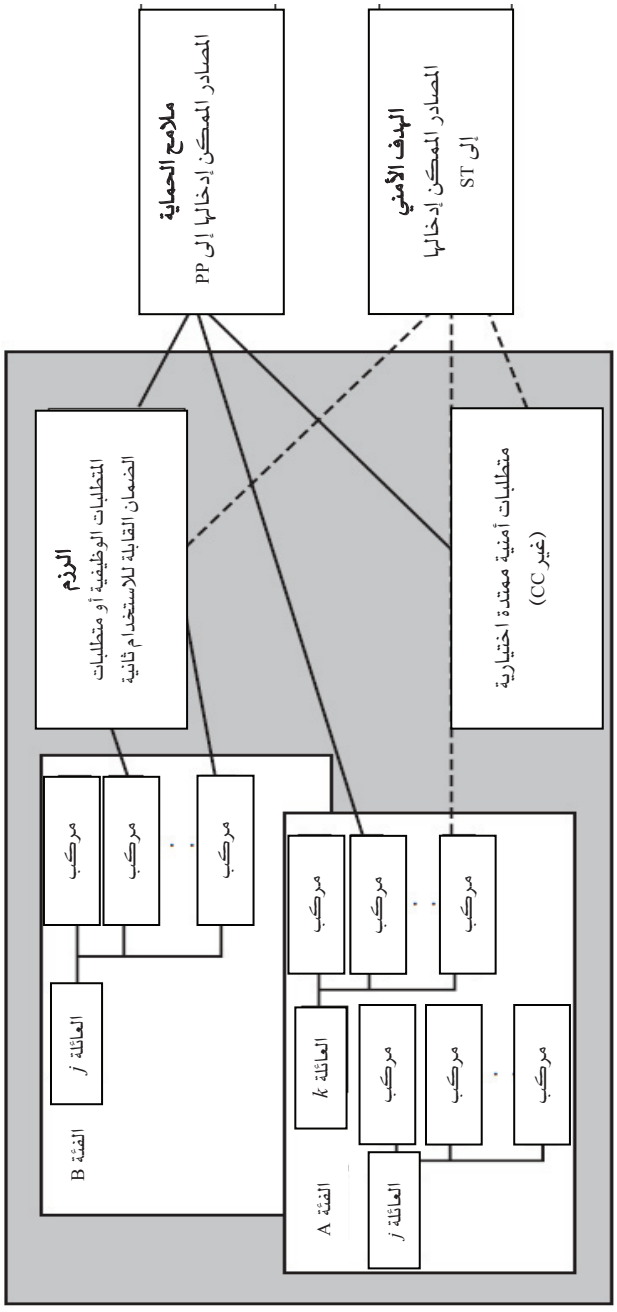
11-3-3 الملامح والأهداف

تحدِّد المعايير المشتركة (CC) أيضاً نوعين من الوثائق التي يمكن إنتاجها باستخدام المتطلبات المعرَّفة في CC.

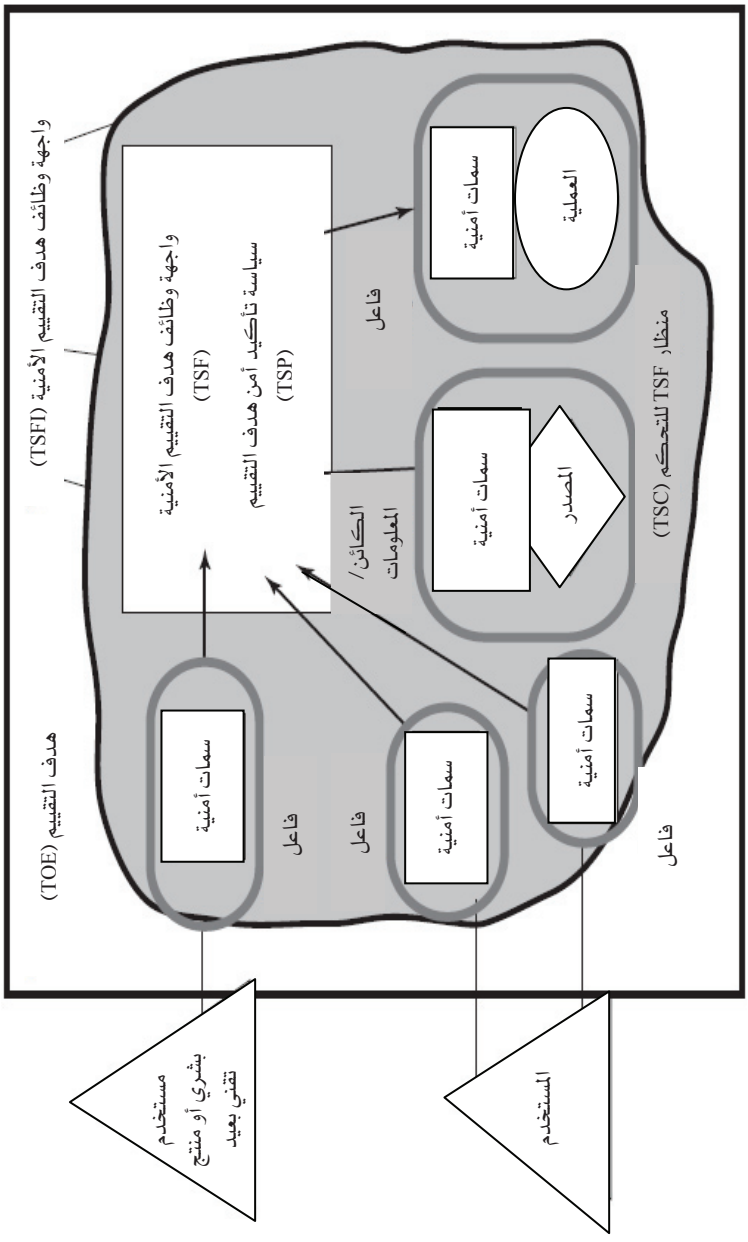
- ملامح الحماية ((Protection profiles (PPs): تحدِّد مجموعة من المتطلبات والأهداف الأمنية لا تعتمد على أسلوب التنفيذ لأحد أصناف المنتجات أو النظم التي تلبي احتياجات المستهلكين المتشابهة لأمن تقنية المعلومات. تستهدف ملامح الحماية أن تكون قابلة لإعادة الاستخدام، وأن تكون مفيدة وفعَّالة في تحقيق الأهداف التي تم تحديدها. وقد تم تطوير مفهوم ملامح الحماية لدعم تعريف المعايير الوظيفية، وكعامل مساعد في صياغة مواصفات الاقتناء. تعكس ملامح الحماية المتطلبات الأمنية للمستخدم.
- الأهداف الأمنية ((Security targets (STs): تتضمن الأهداف والمتطلبات الأمنية لتقنية المعلومات المحدّدة لهدف تقييم معين، كما تحدِّد المقاييس الوظيفية ومقاييس التأمين المقدَّمة من قِبَل هدف التقييم (TOE) المحدّد، لتلبية المتطلبات المنصوص عليها. وقد تتوافق الأهداف الأمنية مع واحد أو أكثر من الملامح الأمنية، كما تشكل أساساً للتقييم. ويتم دعم الأهداف الأمنية من قِبَل البائع أو المطور.

يوضِّح الشكل 11-6 العلاقة بين المتطلبات من جهة والملامح والغايات من جهة أخرى. وبالنسبة لملامح الحماية، يمكن للمستخدم اختيار عدد من العناصر لتحديد متطلبات المنتج المطلوب. كما يمكن للمستخدم أيضاً الاستعانة بالحزم المُعرَّفة مسبقاً التي تضم عدداً من المتطلبات التي تجتمع معاً عادةً في إطار وثيقة متطلبات المنتج. وبالمثل، يمكن للبائع أو المصمم أن يحدد عدداً من المكوِّنات والحزم لتحديد الأهداف الأمنية.

يبين الشكل 11-7 ما يشار إليه في وثائق CC كنموذج متطلبات الأمن الوظيفية. وجوهرياً، يستند هذا التوضيح إلى مفهوم المراقب المَرجع ولكن باستخدام المصطلحات وفلسفة التصميم الخاصة بالأهداف الأمنية.



الشكل 11-6: نظام و ترتيب متطلبات المعايير المشتركة.



الشكل 11-7: نموذج متطلبات الأمن الوظيفي.

4-11 توصيات للمطالعة

يقدم [CHAP00] شرحاً تقليدياً للجدران النارية. وهناك مرجع تقليدي آخر تم تحديثه مؤخراً هو [CHES03]. أما [LODI98]، و[OPPL97]، و[BELL94b] فهي مقالات جيدة تقدم استعراضاً عاماً للموضوع. أيضاً [WACK02] يقدم استعراضاً جيداً لتقنية الجدار الناري وسياساته. ويقدم [AUDI04] و[WILS05] مناقشات مفيدة للجدران النارية. كما يتضمّن [GASS88] دراسة شاملة لنظم الحاسب الموثوقة، ويوفر [PFLE03] و[GOLL99] أيضاً تغطية للموضوع. أما [FELT03] و[OPPL05] فقد قدما مناقشات مفيدة للحاسوبية الموثوقة.

[AUDI04] Audin, G. "Next-Gen Firewalls: What to Expect," *Business Communications Review*, June 2004.

[BELL94b] Bellovin, S., and Cheswick, W. "Network Firewalls," *IEEE Communications Magazine*, September 1994.

[CHAP00] Chapman, D., and Zwicky, E. *Building Internet Firewalls.* Sebastopol, CA: O'Reilly, 2000.

[CHES03] Cheswick, W., and Bellovin, S. *Firewalls and Internet Security: Repelling the Wily Hacker*. Reading, MA: Addison-Wesley, 2003.

[FELT03] Felten, E. "Understanding Trusted Computing: Will Its Benefits Outweigh Its Drawbacks?" *IEEE Security and Privacy*, May/June 2003.

[GASS88] Gasser, M. *Building a Secure Computer System*. New York: Van Nostrand Reinhold, 1988.

[GOLL99] Gollmann, D. *Computer Security.* New York: Wiley, 1999.

[LODI98] Lodin, S., and Schuba, C. "Firewalls Fend Off Invasions from the Net," *IEEE Spectrum*, February 1998.

[OPPL97] Oppliger, R. "Internet Security: Firewalls and Beyond," *Communications of the ACM*, May 1997.

[OPPL05] Oppliger, R., and Rytz, R. "Does Trusted Computing Remedy Computer Security Problems?" *IEEE Security and Privacy*, March/April 2005.

[PFLE03] Pfleeger, C. *Security in Computing*. Upper Saddle River, NJ: Prentice Hall, 2003.

[WACK02] Wack, J.; Cutler, K.; and Pole, J. *Guidelines on Firewalls and Firewall Policy*. NIST Special Publication SP 800-41, January 2002.

[WILS05] Wilson, J. "The Future of the Firewall."*Business Communications Review*, May 2005.

5-11 مصادر للمعلومات على الويب

- Firewall.com: يضم الكثير من الروابط لمراجع عن الجدار الناري وبعض الموارد البرمجية.
- Trusted Computing Group: مجموعة موردين لتطوير معايير الحاسب الموثوقة وتبنيّها. ويحتوي الموقع على أوراق بيضاء (white papers) ومواصفات وبعض روابط للموردين.
- Common Criteria Portal: الموقع الرسمي لمشروع المعايير المشتركة.

6-11 مصطلحات رئيسة

access control list (ACL)	قوائم التحكم في الوصول
access matrix	مصفوفة الوصول
access right	حق الوصول
application-level gateway	بوَّابة مستوى التطبيق
bastion host	مضيف المَعقِل
capability ticket	تذكرة الإمكانات
circuit-level gateway	بوَّابة مستوى الدوائر
common criteria (CC)	المعايير المشتركة
multilevel security	الأمن متعدد المستويات
object	كائن
packet-filtering router	موجِّه ترشيح الرزم
reference monitor	المراقب المَرجع
stateful inspection firewall	جدران تفتيش الحالة
subject	الفاعل
trusted system	النظام الموثوق

7-11 أسئلة للمراجعة ومسائل

1-7-11 أسئلة المراجعة

1-11 عَدِّد ثلاثة أهداف تصميم للجدار الناري.

2-11 عَدِّد أربع تقنيات تستخدمها الجدران النارية للتحكم في الوصول وتنفيذ السياسة الأمنية.

3-11 اذكر المعلومات التي يستخدمها موجِّه ترشيح الرزم.

4-11 اذكر بعض نقاط الضعف في موجِّه ترشيح الرزم.

5-11 ما الفرق بين موجِّه ترشيح الرزم والجدار الناري لتفتيش الحالة؟

6-11 ما المقصود ببوَّابة مستوى التطبيقات؟

7-11 ما المقصود ببوَّابة مستوى الدوائر؟

8-11 ما الفوارق بين التشكيلات الثلاثة بالشكل 11-2؟

9-11 في سياق التحكم في الوصول، ما الفرق بين الفاعل والكائن؟

10-11 ما الفرق بين قائمة التحكم في الوصول وتذكرة الإمكانات؟

11-11 اذكر القاعدتين اللتين يفرضهما المراقب المَرجِع؟

12-11 ما الخصائص اللازمة للمراقب المَرجِع؟

13-11 ما المقصود بالمعايير المشتركة (CC) للأمن؟

2-7-11 مسائل

1-11 كما ذكرنا في الجزء 11-1، إحدى الطرق للتغلب على هجمات التجزئة الدقيقة هي بفرض حد أدنى لطول ترويسة النقل التي يجب أن تَرِد في الجزئية الأولى من رزمة بروتوكول الإنترنت. وإذا تم رفض الجزئية الأولى، يمكن أن يتم رفض جميع الجزئيات اللاحقة. ومع ذلك، فإن طبيعة بروتوكول الانترنت هي أن هذه الجزئيات قد تصل بدون ترتيب. ولذا قد تمر جزئية متوسطة عبر المرشح قبل رفض الجزئية الأولى. كيف يمكن التعامل مع هذا الوضع؟

2-11 في رزمة IPv4، حجم الحمولة في الجزئية الأولى (بالبايتات) =

الطول الكلي - (4×IHL). وإذا كانت هذه القيمة أقل من الحد الأدنى المطلوب (8 بايتات لـ TCP)، فسيتم رفض هذه الجزئية وبقية الرزمة برمَّتها. اقترح طريقة بديلة لتحقيق نفس النتيجة باستخدام حقل إزاحة الجزئية فقط.

3-11 يصف طلب التعليقات RFC 791 الخاص ببروتوكول IPv4 خوارزمية إعادة التجميع التي تُنتِج جزئيات جديدة تحل محل الجزئيات السابقة. في حال وجود هذا البرنامج لإعادة التجميع، يمكن للمهاجم بناء سلسلة من الرزم بحيث تحتوي الجزئية الأولى فيها (ذات الإزاحة الصفرية أي العنوان النسبي = صفر) على بيانات غير ضارة (ومن ثَمَّ يتم تمريرها عن طريق مرشحات الرزمة الإدارية)، وتتداخل فيها بعض الرزم اللاحقة (إزاحتها غير صفرية) مع معلومات ترويسة TCP (منفذ الوجهة، على سبيل المثال)، وتتسبب في تعديلها. ومعظم التنفيذات المختلفة للمرشحات ستقوم بتمرير الرزمة الثانية لأنها ليست صفرية الإزاحة. اقترح طريقة يمكن استخدمها من قِبَل مرشح الرزم للتصدي لهذا الهجوم.

4-11 إن ضرورة قاعدة " ممنوع القراءة لأعلى" لنظام آمن متعدد المستويات واضحةٌ تماماً، فما هي أهمية قاعدة " ممنوع الكتابة لأسفل"؟

5-11 في الشكل 11-5 هناك وصلة واحدة من حصان طروادة "انسخ - ولاحظ - لاحقاً" مكسورة. هناك زاويتان أُخْرَيان يمكن لـ "دريك" (Drake) الهجوم منهما. بعد نفاذ دريك للنظام يحاول قراءة السلسلة مباشرةً، أو أن يعين مستوى أمن الحساسية لملف الجيب الخلفي. هل يمنع المراقب المَرجِع هذه الهجمات؟

الملاحـــق

الملحق A

بعض جوانب نظرية الأعداد

المحتويات:

A-1 الأعداد الأولية والأعداد الأولية نسبياً

- قواسم العدد
- الأعداد الأولية
- الأعداد الأولية نسبياً

A-2 الحساب المقاسي

قال الشيطان لدانيال وبستر: "كلفني بمهمة لا أستطيع القيام بها ، وسأعطيك أي شيء تطلبه في العالم" . فرد عليه دانيال وبستر: "حسناً . اثبت أن المعادلة $a^n + b^n = c^n$ ليس لها حل غير بدهي في الأعداد الصحيحة لقيم n الأكبر من 2" . اتفق الاثنان على فترة ثلاثة أيام من أجل هذا العمل، واختفى الشيطان. في نهاية فترة الثلاثة أيام، ظهر الشيطان منهكاً متوتراً يعض على شفتيه. فقال له دانيال وبستر: "حسنا ، ماذا فعلت في مهمتي؟ هل أثبتّ النظرية؟" . فرد عليه الشيطان: "ماذا؟ لا ... لا ، أنا لم أثبتها" . فقال له دانيال وبستر: "الآن إذن أستطيع أن أطلب ما بدا لي؟ المال؟ رئاسة الجمهورية؟" . فقال الشيطان: "ماذا؟ أوه ... ذلك... بالطبع - ولكن اسمع! ماذا لو تمكنّا من مجرد إثبات النظريتين الفرعيتين التاليتين ....."

– من كتاب الثرثرة الرياضية ، كليفتون فاديمان.

نقدم في هذا الملحق خلفية عن مفهومين أشرنا إليهما في هذا الكتاب: الأعداد الأولية والحساب المقاسي.

A-1 الأعداد الأولية والأعداد الأولية نسبياً

في هذا الجزء – ما لم يُذكر خلاف ذلك – سوف نتعامل فقط مع أعداد صحيحة غير سالبة. غير أن استخدام أعداد صحيحة سالبة لن يؤدي إلى أي اختلافات جوهرية.

قواسم العدد

نقول إن $b \neq 0$ يقسم العدد a، إذا كان $a = mb$ لقيمة من قيم m، حيث إن كلاً من a وb وc أعداد صحيحة. أي أن b يقسم a إذا لم يكن هناك باقٍ في عملية القسمة. يُستخدَم التعبير $b|a$ عادةً للدلالة على أن b يقسم a. أيضاً إذا كان $b|a$ فإننا نقول إن b قاسم لـ a. فمثلاً القواسم الموجبة للعدد 24 هي: 1، 2، 3، 4، 6، 8، 12، 24.

بناءً على ذلك، يمكن إثبات العلاقات الآتية:

- إذا كان $a|1$، فإن $a = \pm 1$
- إذا كان $a|b$ و$b|a$، فإن $a = \pm b$
- إذا كان $b \neq 0$ (لأي قيمة b) ، فإن $b|0$
- إذا كان $b|g$ و$b|h$، فإن $b|(mg+nh)$ لأي قيم للعددين الصحيحين m وn.

لتوضيح النقطة الأخيرة، لاحظ أنه: إذا كان $b|g$، فإن $g = b \times g_1$. وإذا كان $b|h$، فإن $h = b \times h_1$. ومن ثم:

$$mg + nh = mbg_1 + nbh_1 = b \times (mg_1 + nh_1)$$

ومن ثَمَّ فإن b تقسم $mg + nh$.

الأعداد الأولية

العدد الصحيح $p > 1$ هو عدد أولي إذا كان له قاسمان فقط هما $p \pm$ و$1 \pm$. تقوم الأعداد الأولية بدور مهم في نظرية الأعداد وفي الأساليب التي تم مناقشتها في الفصل الثالث.

يمكن التعبير عن أي عدد صحيح $a > 1$ كحاصل ضرب عدة معاملات بشكلٍ فريد على النحو الآتي:

$$a = p_1^{a_1} p_2^{a_2} \ldots p_t^{a_t}$$

حيث $p_1 < p_2 < \ldots < p_t$ أعداد أولية، وa_i أعداد صحيحة موجبة. فمثلاً:

$91 = 7 \times 13$ و$11011 = 7 \times 11^2 \times 13$.

من المفيد وضع هذه النتيجة بطريقة أخرى. فإذا كانت P هي المجموعة التي تضم كل الأعداد الأولية، فإن أي عدد صحيح موجب يمكن كتابته بطريقة فريدة كما يأتي:

$$a = \prod_{p \in P} p^{a_p}$$

حيث $a_p \geq 0$. الجانب الأيمن هو ناتج ضرب كل الأعداد الأولية الممكنة p، وتكون قيم معظم الأسس a_p أصفاراً لأي قيمة معينة لـ a.

يمكن تعيين قيمة أي عدد صحيح موجب ببساطة بتحديد قيم جميع الأسس غير الصفرية في المعادلة السابقة. وهكذا فإن العدد 12 يُمثّل بـ $\{a_2 = 2, a_3 = 1\}$، كما يُمثّل العدد 18 بـ $\{a_2 = 1, a_3 = 2\}$. إن ضرب عددين يناظر جمع الأسس المناظرة:

$$k = mn \rightarrow k_p = m_p + n_p$$

ماذا يعني كون $a|b$ بدلالة تلك العوامل الأولية؟ يمكن قسمة العدد الصحيح الذي له الصيغة p^k فقط على عدد صحيح يساوي نفس العدد الأولي p مرفوعاً لأس أصغر من أو يساوي ذلك الأس k، أي p^j، حيث $j \leq k$. ومن ثَمَّ يمكننا كتابة:

$$a|b \rightarrow a_p \leq b_p$$

وذلك لكل قيم p.

الأعداد الأولية نسبياً

سوف نستخدم الرمز $\gcd(a,b)$ للدلالة على القاسم المشترك الأكبر (greatest common divisor) للعددين a وb. يقال أن العدد الصحيح الموجب c هو القاسم المشترك الأكبر للعددين a وb إذا كان:

1. c هو قاسم لـ a وقاسم لـ b؛
2. أي قاسم لـ a وb يكون أيضاً قاسماً لـ c.

التعريف الآتي يكافئ التعريف أعلاه:

$$\gcd(a,b) = \max [k: \ k|a \text{ و } k|b \text{ بحيث إن}]$$

حيث تدل max على القيمة الكبرى.

نظراً لأننا نشترط أن يكون القاسم المشترك الأكبر موجباً، فإن:

$$\gcd(a,b) = \gcd(-a,b) = \gcd(a,-b) = \gcd(-a,-b)$$

وعموماً، فإن $\gcd(a,b) = \gcd(|a|,|b|)$؛ فمثلاً: $\gcd(60,24) = \gcd(60,-24) = 12$. أيضاً نظراً لأن كل الأعداد الصحيحة غير الصفرية تقسم الصفر، فإن $\gcd(a,0) = |a|$.

من السهل تعيين القاسم المشترك الأكبر لعددين إذا ما عبّرنا عن كل عدد كحاصل ضرب أعداد أولية. فمثلاً:

$$300 = 2^2 \text{ x } 3^1 \times 5^2$$

$$18 = 2^1 \times 3^2$$

$$\gcd(300,18) = 2^1 \times 3^1 \times 5^0 = 6$$

وعموماً فإنه لكل قيم p:

$$k = \gcd(a,b) \rightarrow k_p = \min(a_p, b_p)$$

حيث تدل min على القيمة الصغرى.

إن تحديد العوامل الأولية لعدد كبير ليس بالأمر الهين، ومن ثَمَّ فإن العلاقة أعلاه لا تقودنا بشكلٍ مباشر إلى طريقة لحساب القاسم المشترك الأكبر.

يُعدُّ العددان الصحيحان a وb عددين أوليين نسبياً إذا لم توجد عوامل أولية مشتركة بينهما، أي إذا كان العامل المشترك الوحيد بينهما هو 1. يكافئ هذا قولنا إن a وb عددان أوليان نسبياً إذا كان $\gcd(a,b) = 1$. فمثلاً 8 و15 عددان أوليان نسبياً، حيث إن قواسم 8 هي 1، 2، 4، 8، بينما قواسم 15 هي 1، 3، 5، 15، أي أن 1 هو العدد المشترك الوحيد على كلتا القائمتين.

A-2 الحساب المقاسي (Modular Arithmetic)

لأي عدد صحيح موجب n وأي عدد غير سالب a، إذا قسمنا a على n فإننا نحصل على خارج القسمة q وباقي القسمة r وكلّ منهما عدد صحيح، ويحققان العلاقة:

$$a = qn + r, \quad 0 \le r < n; \; q = \lfloor a/n \rfloor$$

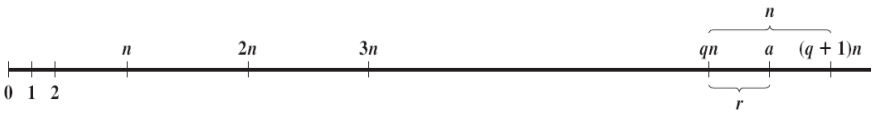
حيث $\lfloor x \rfloor$ تمثل أكبر عدد صحيح أقل من أو يساوي x.

يوضح الشكل A-1 أنه لقيمة معينة لـ a وقيمة موجبة لـ n، فسيكون من الممكن دائماً إيجاد قِيَم q وr اللتين تحققان العلاقة أعلاه. قم بتمثيل الأعداد الصحيحة على خط الأعداد؛ ستقع a في مكانٍ ما على ذلك الخط (الشكل يبين a موجبة، ولكن يمكن توضيح الحالة التي فيها a سالبة بطريقة مماثلة). ابدأ من الصفر، وتقدم إلى n، ثم $2n$، وهكذا حتى qn بحيث $qn \le a$ و $a < (q+1)n$. المسافة من qn إلى a هي r، ومن ثَمَّ نكون قد حصلنا على القيمتين الفريدتين q وr. عادةً ما يُطلق على الكمية r الباقي (residue).

إذا كانت a عدداً صحيحاً وn عدداً صحيحاً موجباً، فإننا نُعرِّف الكمية $a \bmod n$ على أنها الباقي الذي نحصل عليه عند قسمة a على n. ومن ثَمَّ فلكل عدد صحيح a يمكننا كتابة:

$$a = \lfloor a/n \rfloor \times n + (a \bmod n)$$

يُقال عن العددين الصحيحين a وb أنهما متطابقان من حيث باقي القسمة على n (congruent modulo n) إذا كان $(a \bmod n) = (b \bmod n)$، ويُكتب ذلك بالشكل: $a \equiv b \bmod n$. فعلى سبيل المثال: $73 \equiv 4 \bmod 23$، كما أن $21 \equiv -9 \bmod 10$. لاحظ أنه إذا كان $a \equiv 0 \bmod n$ فإن $n|a$.



الشكل A-1: العلاقة $a = qn + r$؛ حيث $0 \le r \le n$.

للعامل المقاسي mod n الخواص الآتية:

1. $a \equiv b \bmod n$ إذا كان $n|(a-b)$.
2. $a \equiv b \bmod n$ تعني ضمناً أن $(a \bmod n) = (b \bmod n)$.
3. $a \equiv b \bmod n$ تعني ضمناً أن $b \equiv a \bmod n$.
4. $a \equiv b \bmod n$ و $b \equiv c \bmod n$ تعني ضمناً أن $a \equiv c \bmod n$.

لتوضيح النقطة الأولى، إذا كانت $n|(a-b)$ فإن $(a-b) = kn$ لقيمة معينة لـ k. ومن ثَمَّ يمكننا أن نكتب: $a = b + kn$. ولذا، فإن: $(a \bmod n)$ يساوي الباقي من قسمة $(b + kn)$ على n يساوي الباقي من قسمة b على n يساوي $(b \bmod n)$. ويمكن إثبات النقاط الباقية بنفس السهولة.

يقوم العامل المقاسي (mod n) بعملية ترحيل (mapping) لجميع الأعداد الصحيحة إلى مجموعة الأعداد $\{0, 1, \ldots, (n-1)\}$، مما يُثير السؤال: هل بوسعنا القيام بعمليات رياضية في النطاق المحدود لمجموعة الأعداد تلك؟ الإجابة في الواقع أننا يمكننا ذلك، ويُطلق على هذا الأسلوب الحساب المقاسي.

للحساب المقاسي الخواص الآتية:

1. $[(a \bmod n) + (b \bmod n)] \bmod n = (a + b) \bmod n$
2. $[(a \bmod n) - (b \bmod n)] \bmod n = (a - b) \bmod n$
3. $[(a \bmod n) \times (b \bmod n)] \bmod n = (a \times b) \bmod n$

سنقوم بتوضيح الخاصية الأولى. لنرمز لـ $(a \bmod n)$ بالرمز r_a ولـ $(b \bmod n)$ بالرمز r_b. يمكننا عندئذٍ كتابة $a = r_a + jn$ لقيمة معينة من العدد الصحيح j، و$b = r_b + kn$ لقيمة معينة من العدد الصحيح k. ومن ثَمَّ:

$$\begin{aligned}(a+b) \bmod n &= (r_a + jn + r_b + kn) \bmod n \\ &= (r_a + r_b + (j+k)n) \bmod n \\ &= (r_a + r_b) \bmod n \\ &= [(a \bmod n) + (b \bmod n)] \bmod n\end{aligned}$$

ويمكن إثبات الخواص الباقية بنفس السهولة.

الملحق B

مشاريع لتعليم أمن الشبكات

المحتويات:

B-1 مشاريع بحثية

B-2 مشاريع برمجية

B-3 تمارين مختبرية

B-4 واجبات كتابية

B-5 واجبات حول القراءة وكتابة التقارير

إن التحليل والملاحظة، وأيضاً النظرية والخبرة، لا ينبغي أبداً لإحداهما أن تزدري الأخرى أو تستبعدها؛ بل على العكس من ذلك، فإنه في الواقع تدعم إحداهما الأخرى. — عن الحرب، كارل فون كلوزويتز

يعتقد كثير من القائمين على التعليم والتدريب أن مشاريع البحث أو التنفيذ حيوية لفهم موضوعات أمن الشبكات بوضوح. فبدون تلك المشاريع قد يكون من الصعب على الطلاب إدراك بعض المفاهيم الأساسية وكيفية حدوث التفاعل بين المكونات المختلفة للأنظمة. تقوم المشاريع بتعزيز المفاهيم الواردة في الكتاب، وزيادة استيعاب الطلاب لكيفية عمل خوارزميات أو بروتوكولات التشفير، ويمكن لها أيضا أن تحفزهم وتمنحهم الثقة في قدراتهم، ليس فقط على فهم، ولكن أيضاً على تنفيذ، تفاصيل مختلف الإمكانات الأمنية.

في هذا الكتاب، حاولت عرض مفاهيم أمن الشبكات بأوضح ما يمكن، كما وفّرت عدداً من مسائل الواجبات المنزلية لتعزيز تلك المفاهيم. ومع ذلك فقد يرغب كثير من القائمين بتدريس الكتاب في دعم مادته ببعض المشاريع. يقدم هذا الملحق بعض التوجيهات في هذا الصدد، كما يصف مواد الدعم الإضافي المتوفرة في دليل المعلّم. تضم مواد الدعم خمسة أنواع من المشاريع:

- مشاريع بحثية
- مشاريع برمجية
- تمارين مختبرية
- واجبات كتابية
- واجبات قراءة وتقارير

1-B مشاريع بحثية

يُعدُّ تكليف الطلاب بمشروع بحثي وسيلة فعالة لتعزيز المفاهيم الأساسية التي يتضمّنها المقرر ولتدريب الطلاب على المهارات البحثية. يمكن أن يتضمَّن مثل ذلك المشروع مسحاً لأدبيات موضوع معين (literature review) وكذلك

البحث على الإنترنت عن منتجات أمن الشبكات المعروضة للبيع، والأنشطة التي تقوم بها مختبرات البحوث الشهيرة حالياً، وجهود تطوير المعايير الأمنية. ويمكن إسناد المشاريع إلى فِرق من الطلاب، أو إلى الطلاب كلٍّ على حدة في حالة المشاريع الصغيرة. وعلى أية حال، يُفضّل أن يُطلب من الطلاب تقديم مُقترح بخصوص المشروع (project proposal) في وقت مبكر من الفصل الدراسي، وإعطاء المعلّم الوقت الكافي لتقييمه من حيث مناسبة الموضوع ومستوى الجهد المطلوب. وينبغي تزويد الطلاب بالمواد الآتية فيما يتعلق بالمشاريع البحثية:

- صيغة المُقترح.
- صيغة التقرير النهائي للمشروع.
- جدول زمني يوضح مواعيد التسليم للتقارير المرحلية والنهائية.
- قائمة ببعض موضوعات المشاريع المُحتملة.

يمكن للطلاب اختيار أحد الموضوعات المدرجة في دليل المعلم أو اختيار مشاريع أخرى من عندهم على نفس المستوى. يتضمَّن ملحق المعلّم صيغاً مقترحة لمُقترح المشروع والتقرير النهائي، فضلا عن قائمة تضم 15 موضوعاً للبحوث الممكنة.

2-B مشاريع برمجية

تعتبر مشاريع البرمجة أداة تعليمية مفيدة. وهناك كثير من المزايا لمشاريع البرمجة القائمة بذاتها والتي لا تشكل جزءاً من المرافق الأمنية المستخدمة:

1. يمكن للمعلّم الاختيار من بين تشكيلة واسعة من مفاهيم التشفير وأمن الشبكات كموضوعات للمشاريع التي تُطرح على الطلاب.
2. يمكن للطلاب برمجة المشاريع على أي جهاز حاسب متوفر وبأي لغة برمجة مناسبة، فتلك المشاريع لا تعتمد على نوعية الحاسب المستخدم ولا على لغة البرمجة.
3. لن يحتاج المعلّم إلى تنزيل أو تركيب أو إعداد أي بنية تحتية معيَّنة لدعم مشاريع البرمجة القائمة بذاتها.

هناك أيضاً قدرٌ من المرونة فيما يتعلق بحجم تلك المشاريع. فالمشاريع الكبيرة تُعطِي الطلاب شعوراً أكبر بالإنجاز، ولكن الطلاب الأقل قدرة في التنظيم أو في المهارات قد يشعرون أنهم تُرِكوا في مؤخرة الركب. وتحتاج المشاريع الكبيرة إلى مزيد من الجهد من أفضل الطلاب. ويمكن أن تتضمن المشاريع الصغيرة نسبة أعلى من المفاهيم مقارنةً بكود البرمجة المطلوب، ونظراً لأنه يمكن طرح عددٍ كبيرٍ من تلك المشاريع الصغيرة، فإن ذلك يتيح الفرصة لمعالجة مجموعة متنوعة من الموضوعات المختلفة.

مرةً أخرى، كما هو الحال في المشاريع البحثية، يجب أن يقدِّم الطلاب أولاً مقترحاً بالمشروع. وينبغي تزويد الطلاب بنفس المواد المذكورة في الجزء 1-B فيما يتعلق بالمشاريع البحثية. ويتضمَّن دليل المعلّم مجموعة من 12 مشروعاً للبرمجة يمكن طرحها.

قام الأشخاص التالية أسماؤهم مشكورين بتزويد المؤلف بالمشاريع البحثية ومشاريع البرمجة المقترحة في دليل المعلّم: Henning Schulzrinne من جامعة كولومبيا، وCetin Kaya Koc من جامعة ولاية أوريجان، وDavid M. Balenson من جامعة جورج واشنطن.

3-B تمارين مختبرية

قام Sanjay Rao وRuben Torres من جامعة بوردو بإعداد مجموعة من التجارب المختبرية والتي تشكل جزءاً من دليل المعلّم. تلك مشاريع تنفيذية يُفترض برمجتها على نظام التشغيل لينكس ولكن يمكن أيضاً تعديلها لتلائم أي بيئة أخرى من بيئات يونيكس. توفر تلك التمارين المختبرية تجربة واقعية في تنفيذ المهام والتطبيقات الأمنية.

B-4 واجبات كتابية

يمكن أن يكون للواجبات الكتابية أثراً مُضاعفاً في عملية التعلم بالمجالات التقنية كمجال التشفير وأمن الشبكات. توصّل أنصار حركة "الكتابة عبر المناهج الدراسية" ((Writing Across the Curriculum (WAC) (قم بزيارة الموقع http://wac.colostate.edu) إلى وجود فوائد كبيرة للكتابة في تسهيل مهام التعلم، حيث تؤدي الواجبات الكتابية إلى تنمية التفكير التفصيلي والمتكامل في موضوع معين.

وبالإضافة لذلك، تساعد تلك الواجبات الطلاب في التغلب على ميلهم لمتابعة الموضوع بحدٍّ أدنى من المشاركة الشخصية، مما يحدو بهم إلى الاقتصار على تعلم الحقائق وأساليب حل المسائل فقط دون السعي لتحصيل فهم أعمق للموضوع.

يتضمَّن دليل المعلّم عدداً من واجبات الكتابة مرتبةً حسب فصول الكتاب. وفي نهاية المطاف قد يجد المعلمون أن هذا هو أهم جزء في منهج تدريس المادة. وسأكون ممتناً لتلقي أية تعليقات في هذا المجال وكذلك أية اقتراحات لواجبات كتابية إضافية.

B-5 واجبات حول القراءة وكتابة التقارير

من الوسائل الأخرى الممتازة لتعزيز المفاهيم المستفادة من المقرر وتزويد الطلاب بتجربة في البحث، تعيين أوراق بحثية من أدبيات الموضوع لقراءتها وتحليلها. ويتضمَّن دليل المعلّم قائمة بأوراق بحثية مقترحة، واحدة أو اثنتين لكل فصل. وكل تلك الأوراق متوفرة بسهولة عن طريق الإنترنت أو في مكتبة أي كلية تقنية ذات مستوى جيد. ويتضمَّن دليل المعلّم أيضا صيغة مقترحة لمثل تلك الواجبات الكتابية.

الملحق C

وقفة مع ترجمة وتعريب المصطلحات

نظرا للجدل المثار حول قضية تعريب العلوم وغرابة المصطلحات واختلافها من بيئة لأخرى، آثرنا أن نوضح للقارئ بعض الأمور الهامة في هذا الملحق. فمن الجدير بالذكر أنه رغم هذا الجدل إلا أن هناك العديد من الجوانب الإيجابية للترجمة والتعريب والتي تتجاوز قضية المصطلحات والخلاف الدائر حول كيفية تعريبها (والذي يعيق في كثير من الأحيان مسيرة هذا العمل الجليل)؛ كسلاسة الأسلوب ووضوح العرض وبساطة التراكيب اللغوية لتوصيل المادة العلمية، والتوازن بين عدم التقيد الجامد باللفظ الأصلي ومراعاة الأمانة في النقل. وقد قمنا بتعريب المصطلحات في هذا الكتاب اعتماداً على التذوق اللغوي وعلى خبرتنا في المجال التقني للكتاب وبعد الرجوع لعدد من القواميس والمعاجم اللغوية الفنية المعتمدة كقاموس المورد، ومعجم المصطلحات العلمية (مجمع اللغة العربية بالقاهرة)، ومعجم الكيلاني لمصطلحات الكمبيوتر والإنترنت (للدكتور تيسير الكيلاني)، ودليل شعاع لمصطلحات الحاسب، ومعجم مصطلحات الكمبيوتر (الدار العربية للعلوم)، وبنك المصطلحات الموحدة (مكتب تنسيق التعريب بالرباط)، والبنك الآلي السعودي للمصطلحات العلمية (باسم)؛ وبالاطلاع على بعض ما نشر من الكتب العربية والموسوعات العلمية المتخصصة الأخرى؛ وبالاطلاع على عدد من مواقع الإنترنت كموقع الجمعية الدولية للمترجمين واللغويين العرب (واتا)، وموقع شبكة صوت العربية، وموقع الجمعية العلمية السورية للمعلوماتية، وموقع مجمع اللغة العربية الأردني (تعريب المصطلحات العلمية)، وموقع المركز العربي للتعريب والترجمة والنشر بدمشق (مجلة التعريب)، وموقع الموسوعة العربية، وموقع مجلة عالم الكمبيوتر والإنترنت، وغيرها من مواقع الإنترنت الأخرى؛ وأحياناً بالتشاور المتأني وبالاستعانة بمدقق لغوي. ومن الضوابط التي

راعيناها – قدر المستطاع – في انتقاء المصطلح العربي: مكافأته للمصطلح الأصلي وعدم التباسه مع مصطلح آخر وشيوعه وقصره وخفة لفظه ومطابقته لقواعد اللغة.

بشكلٍ عام هناك خمسة مذاهب رئيسة متعارف عليها في ترجمة المصطلحات. فبعض المترجمين يكتب المصطلح كما يُنطق (وهو ما يعرف بـ transliteration). فمثلاً يستخدمون كلمة "ماوس" لترجمة "mouse"، ويستخدمون "بت" لترجمة "bit"، ويستخدمون "إنترنت" لترجمة "Internet"، ويستخدمون "فيديو" لتعني "video"، وهكذا. وآخرون يستخدمون المرادف باللغة العربية للمصطلح فمثلاً يستخدمون "فأرة" لترجمة "mouse" ويستخدمون "رقم ثنائي" لترجمة "bit"، وهكذا. والمذهب الثالث يحاول صياغة ألفاظ ومصطلحات باللغة العربية كمقابل للمصطلح الأجنبي (وهو ما يعرف بالتحوير)؛ فمثلاً تُستخدم "الأشعة السينية" لترجمة "x-ray"، وتُستخدم "الشابكة" لترجمة "Internet"، وهكذا. وهناك من يفضل أن يكتب المصطلح كما هو بلغته الأصلية بدون تعريب. وهناك من يكتب المعنى الوصفي ("التراكيب الوصفية للمعنى الوظيفي") للمصطلح كأن تستخدم "مشيرة" لترجمة "mouse" وتقول "شبكة عالمية للربط بين الشبكات التي تستخدم مجموعة البروتوكولات TCP/IP للاتصال فيما بينها....." لترجمة "Internet". يلاحظ أن هذه المصطلحات تتفاوت فيما بينها لاختيار الطريقة الأنسب لترجمتها. وسبب ذلك الخلاف أن المصطلح يرتبط في الذهن بتصور معين والذي قد يعتمد على بيئة القارئ وسياق الحديث وبيئة المنشأ للمصطلح ومدى انتشاره. وهذا أمر خلافي حتى في المصطلحات غير التقنية؛ مثلاً ماذا يعني "لمس النساء"؟ وماذا تعني "حَبْحَب"؟ وغير ذلك الكثير.

فمثلاً ماذا نترجم الكلمات "Java"، "Google"، "Aloha"، "Cookies"، الخ. إن مثل هذه الكلمات تعتبر كأسماء الأعلام ارتبطت بتصور معين ويجب أن تترك كما هي. خاصةً وقد أصبحت تلك المصطلحات في كثير من الأحيان دارجة ومعروفة حتى لرجل الشارع، وأنه يتكرر ذكرها عشرات بل مئات المرات في الكتاب!! فهل يستساغ أن نترجم "Java" على أنها "نوع من البن"، ونترجم "Cookies" على أنها "كعك"، ونترجم "Internet" في كل مرة ترد في الكتاب على أنها "شبكة عالمية تربط بين الشبكات التي تستخدم مجموعة البروتوكولات TCP/IP للاتصال فيما بينها وتتألف من مجموعة من الأجهزة وخطوط النقل....."؟! أو أن نقصد مباشرة لما ارتبط بالذهن بقول "Java" أو "جافا"، وقول "كوكيز" أو "Cookies"، وقول "Internet" أو "إنترنت"!

الخلاصة: إنّنا لا ندعي أننا سنحل هذا الخلاف، لكننا راعينا المزج بين المذاهب المختلفة حفاظاً على سلاسة الأسلوب وسلامة التراكيب اللغوية ومراعاةً لمدى شيوع اللفظ في البيئة العربية حتى نتجنب استخدام ألفاظ مستغربة (قدر المستطاع) والتي قد يعيق أو يصعب من فهم المادة العلمية. وقد تم ذلك بالرجوع لعدد من مجامع اللغة العربية ومعاجم المصطلحات كما بينا سابقاً وبالاطلاع على بعض الكتب العربية في مجال الكتاب، وأحياناً بالاجتهاد والتشاور المتأني في بعض الألفاظ المستحدثة. فالهدف من الكتاب التقني هو توصيل المعلومة بشكل سريع وسلس للقارئ مع مراعاة قواعد اللغة ما أمكن ذلك (دون افتعال لغوي). وجدير بالذكر أن عدد المصطلحات التي يوجد خلاف في ترجمتها ليس كبيراً.

ولنأخذ عينة لترجمة بعض المصطلحات كما هو مبين في الجدول التالي:

□

المصدر	Bit	Byte	Internet	Web	Real time	Computer
مجمع اللغة العربية بالقاهرة	بيتة	بايت، ثمانية أرقام ثنائية، جزء من كلمة الحاسب	الإنترنت، الشبكة الدولية	الويب، الشبكة العنكبوتية العالمية	الوقت الحقيقي، الوقت الفعلي	الحاسب
بنك المصطلحات الموحدة، مكتب تنسيق التعريب، المنظمة العربية للتربية والثقافة والعلوم، المغرب	ثنائية	ثُمانية	- -	- -	الزمن الحقيقي، الآني	حاسوب، حاسب

البنك الآلي السعودي للمصطلحات (باسم)، السعودية	بت، بتة، رقم ثنائي، شارة، خانة	بايت، شارة	الإنترنت، الشبكة العنكبوتية العالمية	الشبكة العنكبوتية العالمية.	الوقت الحقيقي، الوقت الفعلي، الوقت الآني، فوري	الحاسب الآلي، آلة حاسبة، كمبيوتر
معجم الكيلاني لمصطلحات الكمبيوتر والإنترنت (للدكتور تيسير الكيلاني وهو فلسطيني قام بالتدريس في عدد من البلدان العربية)، مكتبة بيروت 2004	بت، رقم ثنائي، نبضة، شارة	بايت، مجموعة أرقام ثنائية، ثماني بتات	الإنترنت	الويب، نصوص الشبكة العالمية للاتصالات	الزمن الحقيقي	كمبيوتر، جهاز حاسب، الحاسب الآلي، الحاسب الإلكتروني
معجم مصطلحات الكمبيوتر، الدار العربية للعلوم، 2001، لبنان	بت، خانة ثنائية	بايت، ثماني بتات	الإنترنت	Web	الزمن الحقيقي	حاسوب، كمبيوتر
دليل شعاع لمصطلحات الحاسب، طبعة 2004، سوريا	بت، خلية ثنائية	بايت	الإنترنت	الويب	الزمن الحقيقي	حاسب
مترجم جوجل	مقدار ضئيل	بايت	الإنترنت	الإنترنت	الوقت الحقيقي	كمبيوتر
مترجم الوافي الذهبي	قطعة	بايت	الإنترنت	الويب	الوقت الحقيقي	حاسوب

الترجمة التي رجحناها لتلك المصطلحات

Bit	Byte	Internet	Web	Real time	Computer
بت	بايت	الإنترنت	الويب	الزمن الحقيقي	الحاسب

ونود أن نشير إلى أنه رغم تبني أحد المصطلحات إلا أننا في بعض الأحيان أشرنا لبعض البدائل الأخرى (وخصوصاً عندما يذكر المصطلح لأول مرة وعندما نجد لذلك ضرورة لتوصيل المعنى بشكل أفضل).

نذكر مثالاً آخر لمصطلحات قريبة في المعنى إلا أنها تعني أشياء مختلفة عندما تستخدم المصطلح الأصلي؛ مثال ذلك المصطلحات: label، token، flag، semaphore، signal. وقد استخدمنا ألفاظاً مختلفة للتفريق بينها كما يلي:

Label	token	flag	semaphore	signal
وسمة	علامة، توكن	علم	سيمافور	إشارة

في النهاية ينبغي التأكيد على أننا لسنا في هذا الكتاب بصدد إعداد معجم موحد للمصطلحات وإنما آثارنا التبسيط ووضوح وسلاسة العرض وتسلسل الأفكار لتوصيل المعنى العلمي المراد عند الوقوف على قضية المصطلح.

ثبت المراجع

ALVA90 Alvare, A. "How Crackers Crack Passwords or What Passwords to Avoid." *Proceedings, UNIX Security Workshop II*, August 1990.

ANDE80 Anderson, J. *Computer Security Threat Monitoring and Surveillance*. Fort Washington, PA: James P. Anderson Co., April 1980.

AUDI04 Audin, G. "Next-Gen Firewalls: What to Expect," *Business Communications Review*, June 2004.

AXEL00 Axelsson, S. "The Base-Rate Fallacy and the Difficulty of Intrusion Detection." *ACM Transactions and Information and System Security*, August 2000.

BACE00 Bace, R. *Intrusion Detection.* Indianapolis, IN: Macmillan Technical Publishing, 2000.

BACE01 Bace, R., and Mell, P. *Intrusion Detection Systems*. NIST Special Publication SP 800-31, November 2000.

BARR03 Barreto, P., and Rijmen, V. "The Whirlpool Hashing Function." Submitted to NESSIE, September 2000, revised May 2003.

BAUE88 Bauer, D., and Koblentz, M."NIDX—An Expert System for Real-Time Network Intrusion Detection." *Proceedings, Computer Networking Symposium*, April 1988.

BELL90 Bellovin, S., and Merritt, M. "Limitations of the Kerberos Authentication System." *Computer Communications Review*, October 1990.

BELL92 Bellovin, S. "There Be Dragons." *Proceedings, UNIX Security Symposium III*, September 1992.

BELL93 Bellovin, S. "Packets Found on an Internet." *Computer Communications Review*, July 1993.

BELL94 Bellovin, S., and Cheswick, W. “Network Firewalls.” *IEEE Communications Magazine*, September 1994.

BELL96a Bellare, M.; Canetti, R.; and Krawczyk, H. “Keying Hash Functions for Message Authentication.” *Proceedings, CRYPTO ’96*, August 1996; published by Springer-Verlag. An expanded version is available at http://www-cse.ucsd.edu/users/mihir.

BELL96b Bellare, M.; Canetti, R.; and Krawczyk, H. “The HMAC Construction.” *CryptoBytes*, Spring 1996.

BERS92 Berson, T. “Differential Cryptanalysis Mod 232 with Applications to MD5.” *Proceedings, EUROCRYPT ’92*, May 1992; published by Springer-Verlag.

BISH03 Bishop, M. *Computer Security: Art and Science*. Boston: Addison-Wesley, 2003.

BISH05 Bishop, M. *Introduction to Computer Security*. Boston: Addison-Wesley, 2005.

BLOO70 Bloom, B. “Space/time Trade-offs in Hash Coding with Allowable Errors.” *Communications of the ACM*, July 1970.

BLUM97a Blumenthal, U.; Hien, N.; and Wijnen, B. “Key Derivation for Network Management Applications.” IEEE Network, May/June, 1997.

BLUM97b Blumenthal, U, and Wijnen, B. “Security Features for SNMPv3.” *The Simple Times*, December 1997.

BOER93 Boer, B., and Bosselaers, A. “Collisions for the Compression Function of MD5.” *Proceedings, EUROCRYPT ’93*, 1993; published by Springer-Verlag.

BRYA88 Bryant, W. *Designing an Authentication System: A Dialogue in Four Scenes*. Project Athena document, February 1988. Available: http://web.mit.edu/kerberos/www/dialogue.html.

CASS01 Cass, S. “Anatomy of Malice.” *IEEE Spectrum*, November 2001.

CERT01 CERT Coordination Center. “Denial of Service Attacks.” June 2001. http://www.cert.org/tech_tips/denial_of_service.html

CERT02 CERT Coordination Center. "Multiple Vulnerabilities in Many Implementations of the Simple Network Management Protocol." CERT Advisory CA-2002-03, 25 June 2002. www.cert.org/advisories/CA-2002-03.html

CHAN02 Chang, R. "Defending Against Flooding-Based Distributed Denial-of-Service Attacks: A Tutorial." *IEEE Communications Magazine*, October 2002.

CHAP00 Chapman, D., and Zwicky, E. *Building Internet Firewalls*. Sebastopol, CA: O'Reilly, 2000.

CHEN98 Cheng, P., et al. "A Security Architecture for the Internet Protocol." *IBM Systems Journal*, Number 1, 1998.

CHES97 Chess, D. "The Future of Viruses on the Internet." *Proceedings, Virus Bulletin International Conference*, Oct. 1997.

CHES03 Cheswick, W., and Bellovin, S. *Firewalls and Internet Security: Repelling the Wily Hacker*. Reading, MA: Addison-Wesley, 2003.

COHE94 Cohen, F. A Short Course on Computer Viruses. New York: Wiley, 1994.

CORM01 Cormen, T.; Leiserson, C.; Rivest, R.; and Stein, C. *Introduction to Algorithms*. Cambridge, MA: MIT Press, 2001.

DAVI89 Davies, D., and Price, W. Security for Computer Networks. New York: Wiley, 1989.

DAMG89 Damgard, I. "A Design Principle for Hash Functions." *Proceedings, CRYPTO '89*, 1989; published by Springer-Verlag.

DAVI93 Davies, C., and Ganesan, R. "BApasswd: A New Proactive Password Checker." *Proceedings, 16th National Computer Security Conference*, September 1993.

DAWS96 Dawson, E., and Nielsen, L. "Automated Cryptoanalysis of XOR Plaintext Strings." *Cryptologia*, April 1996.

DENN87 Denning, D. “An Intrusion-Detection Model.” *IEEE Transactions on Software Engineering*, February 1987.

DIFF76 Diffie, W., and Hellman, M. “Multiuser Cryptographic Techniques.” *IEEE Transactions on Information Theory*, November 1976.

DIFF88 Diffie, W. “The First Ten Years of Public-Key Cryptography.” *Proceedings of the IEEE*, May 1988. Reprinted in [SIMM92].

DOBB96 Dobbertin, H. “The Status of MD5 After a Recent Attack.” *CryptoBytes*, Summer 1996.

DORA03 Doraswamy, N., and Harkins, D. *IPSec*. Upper Saddle River, NJ: Prentice Hall, 2003.

DREW99 Drew, G. *Using SET for Secure Electronic Commerce*. Upper Saddle River, NJ: Prentice Hall, 1999.

EFF98 Electronic Frontier Foundation. *Cracking DES: Secrets of Encryption Research, Wiretap Politics, and Chip Design*. Sebastopol, CA: O’Reilly, 1998

ENGE80 Enger, N., and Howerton, P. *Computer Security*. New York: Amacom, 1980.

FEIS73 Feistel, H. “Cryptography and Computer Privacy.” *Scientific American*, May 1973.

FELT03 Felten, E. “Understanding Trusted Computing:Will Its Benefits Outweigh Its Drawbacks?” *IEEE Security and Privacy*, May/June 2003.

FLUH00 Fluhrer, S., and McGrew, D. “Statistical Analysis of the Alleged RC4 Key Stream Generator.” *Proceedings, Fast Software Encryption*, 2000.

FLUH01 Fluhrer, S.; Mantin, I.; and Shamir, A. “Weakness in the Key Scheduling Algorithm of RC4.” *Proceedings, Workshop in Selected Areas of Cryptography*, 2001.

FORD95 Ford, W. “Advances in Public-Key Certificate Standards.” ACM SIGSAC Review, July 1995.

FORR97 Forrest, S.; Hofmeyr, S.; and Somayaji, A. “Computer Immunology.” *Communications of the ACM*, October 1997.

FRAN01 Frankel, S. *Demystifying the IPSec Puzzle.* Boston: Artech House, 2001.

GARD77 Gardner, M. "A New Kind of Cipher That Would Take Millions of Years to Break." *Scientific American*, August 1977.

GARF97 Garfinkel, S., and Spafford, G. *Web Security & Commerce*. Cambridge, MA: O'Reilly and Associates, 1997.

GASS88 Gasser, M. *Building a Secure Computer System*. New York: Van Nostrand Reinhold, 1988.

GAUD00 Gaudin, S. "The Omega Files." *Network World*, June 26, 2000.

GOLL99 Gollmann, D. *Computer Security.* New York: Wiley, 1999.

GUTM02 Gutmann, P. "PKI: It's Not Dead, Just Resting." *Computer*, August 2002.

HARL01 Harley, D.; Slade, R.; and Gattiker, U. *Viruses Revealed*. New York: Osborne/McGraw-Hill, 2001.

HEBE92 Heberlein, L.; Mukherjee,B.; and Levitt, K."Internetwork Security Monitor:An Intrusion-Detection System for Large-Scale Networks." *Proceedings, 15th National Computer Security Conference*, October 1992.

HELD96 Held, G. *Data and Image Compression: Tools and Techniques*. New York: Wiley, 1996.

HONE01 The Honeynet Project. *Know Your Enemy: Revealing the Security Tools, Tactics, and Motives of the Blackhat Community.* Reading, MA: Addison-Wesley, 2001.

HUIT98 Huitema, C. *IPv6: The New Internet Protocol*. Upper Saddle River, NJ: Prentice Hall, 1998.

IANS90 I'Anson, C., and Mitchell, C. "Security Defects in CCITT Recommendation X.509 – The Directory Authentication Framework." *Computer Communications Review*, April 1990.

ILGU93 Ilgun, K. “USTAT: A Real-Time Intrusion Detection System for UNIX.” *Proceedings, 1993 IEEE Computer Society Symposium on Research in Security and Privacy*, May 1993.

JAVI91 Javitz, H., and Valdes, A. “The SRI IDES Statistical Anomaly Detector.” *Proceedings, 1991 IEEE Computer Society Symposium on Research in Security and Privacy*, May 1991.

JIAN02 Jiang, G. “Multiple Vulnerabilities in SNMP.” *Security and Privacy Supplement to Computer Magazine*, 2002.

JUEN85 Jueneman, R.; Matyas, S.; and Meyer, C. “Message Authentication.” *IEEE Communications Magazine*, September 1988.

KENT00 Kent, S. “On the Trail of Intrusions into Information Systems,” *IEEE Spectrum*, December 2000.

KEPH97a Kephart, J.; Sorkin, G.; Chess, D.; and White, S. “Fighting Computer Viruses.” *Scientific American*, November 1997.

KEPH97b Kephart, J.; Sorkin, G.; Swimmer, B.; and White, S. “Blueprint for a Computer Immune System.” *Proceedings, Virus Bulletin International Conference*, October 1997.

KLEI90 Klein, D. “Foiling the Cracker: A Survey of, and Improvements to, Password Security.” *Proceedings, UNIX Security Workshop II*, August 1990.

KNUD98 Knudsen, L., et al. “Analysis Method for Alleged RC4.” *Proceedings, ASIACRYPT ’98*, 1998.

KOBL92 Koblas, D., and Koblas, M. “SOCKS.” *Proceedings, UNIX Security Symposium III*, September 1992.

KOHL89 Kohl, J. “The Use of Encryption in Kerberos for Network Authentication.” *Proceedings, Crypto ’89*, 1989; published by Springer-Verlag.

KOHL94 Kohl, J.; Neuman, B.; and Ts’o, T. “The Evolution of the Kerberos Authentication Service.” in Brazier, F., and Johansen, D. *Distributed Open Systems*. Los Alamitos, CA: IEEE Computer Society Press, 1994. Available at http://web.mit.edu/kerberos/www/papers.html.

KUMA97 Kumar, I. *Cryptology*. Laguna Hills, CA: Aegean Park Press, 1997.

LEUT94 Leutwyler, K. "Superhack." *Scientific American*, July 1994.

LODI98 Lodin, S., and Schuba, C. "Firewalls Fend Off Invasions from the Net," *IEEE Spectrum*, February 1998.

LUNT88 Lunt,T., and Jagannathan, R. "A Prototype Real-Time Intrusion-Detection Expert System." *Proceedings, 1988 IEEE Computer Society Symposium on Research in Security and Privacy*, April 1988.

MACG97 Macgregor, R.; Ezvan, C.; Liguori, L.; and Han, J. *Secure Electronic Transactions: Credit Card Payment on the Web in Theory and Practice.* IBM RedBook SG24-4978-00, 1997. Available: www.redbooks.ibm.com.

MADS93 Madsen, J. "World Record in Password Checking." Usenet, comp.security.misc newsgroup, August 18, 1993.

MANT01 Mantin, I., Shamir,A. "A Practical Attack on Broadcast RC4." *Proceedings, Fast Software Encryption*, 2001.

MARK97 Markham, T. "Internet Security Protocol." *Dr. Dobb's Journal*, June 1997.

MCHU00 McHugh, J.; Christie, A.; and Allen, J. "The Role of Intrusion Detection Systems." *IEEE Software*, September/October 2000.

MEIN01 Meinel, C. "Code Red for the Web." *Scientific American*, October 2001.

MENE97 Menezes, A.; van Oorschot, P.; and Vanstone, S. *Handbook of Applied Cryptography*. Boca Raton, FL: CRC Press, 1997.

MERK79 Merkle, R. Secrecy, *Authentication, and Public Key Systems*. PhD Thesis, Stanford University, June 1979.

MERK89 Merkle, R. "One Way Hash Functions and DES." Proceedings, *CRYPTO '89*, 1989; published by Springer-Verlag.

MEYE82 Meyer, C., and Matyas, S. *Cryptography: A New Dimension in Computer Data Security*. New York: Wiley, 1982.

MILL88 Miller, S.; Neuman, B.; Schiller, J.; and Saltzer, J. "Kerberos Authentication and Authorization System." Section E.2.1, Project Athena Technical Plan, MIT Project Athena, Cambridge, MA. 27 October 1988.

MIRK04 Mirkovic, J., and Relher, P. "A Taxonomy of DDoS Attack and DDoS Defense Mechanisms." *ACM SIGCOMM Computer Communications Review*, April 2004.

MIST98 Mister, S., and Tavares, S. "Cryptanalysis of RC4-Like Ciphers." *Proceedings, Workshop in Selected Areas of Cryptography*, SAC' 98. 1998.

MITC90 Mitchell, C.; Walker, M.; and Rush, D. "CCITT/ISO Standards for Secure Message Handling." *IEEE Journal on Selected Areas in Communications*, May 1989.

MOOR01 Moore, M. "Inferring Internet Denial-of-Service Activity." *Proceedings of the 10th USENIX Security Symposium*, 2001.

NACH97 Nachenberg, C."Computer Virus-Antivirus Coevolution." *Communications of the ACM*, January 1997.

NEED78 Needham, R., and Schroeder, M. "Using Encryption for Authentication in Large Networks of Computers." *Communications of the ACM*, December 1978.

NING04 Ning, P., et al. "Techniques and Tools for Analyzing Intrusion Alerts." *ACM Transactions on Information and System Security*, May 2004.

OPPL97 Oppliger, R. "Internet Security: Firewalls and Beyond," *Communications of the ACM*, May 1997.

OPPL05 Oppliger, R., and Rytz, R. "Does Trusted Computing Remedy Computer Security Problems?" *IEEE Security and Privacy*, March/April 2005.

PATR04 Patrikakis, C.; Masikos, M.; and Zouraraki, O. "Distributed Denial of Service Attacks." *The Internet Protocol Journal*, December 2004.

PAUL03 Paul, S., and Preneel, B. "Analysis of Non-fortuitous Predictive States of the RC4 Keystream Generator." *Proceedings, INDOCRYPT '03*, 2003.

PAUL04 Paul, S., and Preneel, B. "A New Weakness in the RC4 Keystream Generator and an Approach to Improve the Security of the Cipher." *Proceedings, Fast Software Encryption*, 2004.

PERL99 Perlman, R. "An Overview of PKI Trust Models." *IEEE Network*, November/December 1999.

PFLE03 Pfleeger, C. *Security in Computing*. Upper Saddle River, NJ: Prentice Hall, 2003.

PIAT91 Piattelli-Palmarini, M. "Probability: Neither Rational nor Capricious." Bostonia, March 1991.

PIEP03 Pieprzyk, J.; Hardjono, T.; and Seberry, J. *Fundamentals of Computer Security*. New York: Springer-Verlag, 2003.

PORR92 Porras, P. STAT: A State Transition Analysis Tool for Intrusion Detection. Master Thesis, University of California at Santa Barbara, July 1992.

PREN02 Preenel, B. "New European Schemes for Signature, Integrity and Encryption (NESSIE): A Status Report." *Proceedings of the 5th International Workshop on Practice and Theory in Public Key Cryptosystems: Public Key Cryptography*. 2002.

PROC01 Proctor, P., *The Practical Intrusion Detection Handbook.* Upper Saddle River, NJ: Prentice Hall, 2001.

PUDO02 Pudovkina, M. "Statistical Weaknesses in the Alleged RC4 Keystream Generator." *Proceedings, 4th International Workshop on Computer Science and Information Technologies*, 2002.

RESC01 Rescorla, E. *SSL and TLS: Designing and Building Secure Systems.* Reading, MA: Addison-Wesley, 2001.

RIVE78 Rivest, R.; Shamir, A.; and Adleman, L. "A Method for Obtaining Digital Signatures and Public Key Cryptosystems." *Communications of the ACM*, Feb. 1978.

ROBS95a Robshaw, M. Stream Ciphers. RSA Laboratories Technical Report TR-701, July 1995. http://www.rsasecurity.com/rsalabs

ROBS95b Robshaw, M. Block Ciphers. RSA Laboratories Technical Report TR-601, 1995. http://www.rsasecurity.com/rsalabs

RODR02 Rodriguez, A., et al. *TCP/IP Tutorial and Technical Overview*. Upper Saddle River: NJ: Prentice Hall, 2002.

SAFF93 Safford, D.; Schales, D.; and Hess, D. "The TAMU Security Package: An Ongoing Response to Internet Intruders in an Academic Environment." *Proceedings, UNIX Security Symposium IV*, October 1993.

SCHN00 Schnier, B. *Secrets and Lies: Digital Security in Networked World*. New York: Wiley, 2000.

SCHN96 Schneier, B. *Applied Cryptography*. New York: Wiley, 1996.

SIMM92 Simmons, G., ed. *Contemporary Cryptology: The Science of Information Integrity*. Piscataway, NJ: IEEE Press, 1992.

SING99 Singh, S. *The Code Book: The Science of Secrecy from Ancient Egypt to Quantum Cryptography*. New York: Anchor Books, 1999.

SMIT97 Smith, R. *Internet Cryptography*. Reading, MA: Addison-Wesley, 1997.

SNAP91 Snapp, S., et al. "A System for Distributed Intrusion Detection." *Proceedings, COMPCON*, Spring '91, 1991.

SPAF92a Spafford, E. "Observing Reusable Password Choices." *Proceedings, UNIX Security Symposium III*, September 1992.

SPAF92b Spafford, E. "OPUS: Preventing Weak Password Choices." *Computers and Security*, No. 3, 1992.

STAL99 Stallings, W. *SNMP, SNMPv2, SNMPv3, and RMON 1 and 2*. Reading, MA: Addison-Wesley, 1999.

STAL04 Stallings, W. *Computer Networking with Internet Protocols and Technology*. Upper Saddle River, NJ: Prentice Hall, 2004.

STAL06a Stallings, W. *Cryptography and Network Security: Principles and Practice*, Fourth Edition. Upper Saddle River, NJ: Prentice Hall, 2006.

STAL06b Stallings, W. "The Whirlpool Secure Hash Function." *Cryptologia*, January 2006.

STEI88 Steiner, J.; Neuman, C.; and Schiller, J. "Kerberos: An Authentication Service for Open Networked Systems." *Proceedings of the Winter 1988 USENIX Conference*, February 1988.

STEP93 Stephenson, P. "Preventive Medicine." *LAN Magazine*, November 1993.

STER92 Sterling, B. *The Hacker Crackdown: Law and Disorder on the Electronic Frontier*. New York: Bantam, 1992.

STIN06 Stinson, D. *Cryptography: Theory and Practice*. Boca Raton, FL: CRC Press, 2006.

STOL88 Stoll, C. "Stalking the Wily Hacker." *Communications of the ACM*, May 1988.

STOL89 Stoll, C. *The Cuckoo's Egg*. New York: Doubleday, 1989.

SZOR05 Szor, P., *The Art of Computer Virus Research and Defense*. Reading, MA: Addison-Wesley, 2005.

THOM84 Thompson, K. "Reflections on Trusting Trust (Deliberate Software Bugs)." *Communications of the ACM*, August 1984.

TIME90 Time, Inc. Computer Security, Understanding Computers Series. Alexandria, VA: Time-Life Books, 1990.

TSUD92 Tsudik, G. "Message Authentication with One-Way Hash Functions." *Proceedings, INFOCOM '92*, May 1992.

TUNG99 Tung, B. *Kerberos: A Network Authentication System*. Reading, MA: Addison-Wesley, 1999.

VACC89 Vaccaro, H., and Liepins, G. "Detection of Anomalous Computer Session Activity." *Proceedings of the IEEE Symposium on Research in Security and Privacy*, May 1989.

VIJA02 Vijayan, J. "Denial-of-Service Attacks Still a Threat." *ComputerWorld*, April 8, 2002.

WACK02 Wack, J.; Cutler, K.; and Pole, J. *Guidelines on Firewalls and Firewall Policy*. NIST Special Publication SP 800-41, January 2002.

WAGN00 Wagner, D., and Goldberg, I. “Proofs of Security for the UNIX Password Hashing Algorithm.” *Proceedings, ASIACRYPT ’00*, 2000.

WANG05 Wang, X.; Yin, Y.; and Yu, H. “Finding Collisions in the Full SHA-1.” *Proceedings, Crypto’05, 2005*; published by Springer-Verlag.

WILS05 Wilson, J. “The Future of the Firewall.” *Business Communications Review*, May 2005.

YUVA79 Yuval, G. “How to Swindle Rabin.” *Cryptologia*, July 1979.

ZIV77 Ziv, J., and Lempel, A. “A Universal Algorithm for Sequential Data Compression.” *IEEE Transactions on Information Theory*, May 1977.

المصطلحات الرئيسة

access control	التحكم في النفاذ
Access Control List (ACL)	قائمة التحكم في الوصول
access matrix	مصفوفة الوصول
access policy	سياسة الوصول، سياسة النفاذ
access right	حق الوصول
acquirer	مُحصِّل
active threat	تهديد أمني نشط أو ظاهر
Advanced Encryption Standard (AES)	معيار التشفير المتقدم (AES)
agent	وكيل
anti-replay service	خدمة مضادة لإعادة الإرسال
application-level gateway	بوابة مستوى التطبيق
audit record	سجل مراجعة
authentication	توثيق، التحقق من الهوية، استيثاق
Authentication Header (AH)	ترويسة التوثيق
authentication server	خادم التوثيق
authenticity	الموثوقية
auto-rooter	برنامج للوصول التلقائي للجذر
availability	الإتاحية، توافر الخدمة
backdoor	مدخل خلفي
base-rate fallacy	مغالطة المعدل الأساسي
bastion host	مضيف المعقل
Bayes' theorem	نظرية بايز
block cipher	تشفير الكتلة

brute-force attack	الهجوم الاستقصائي
capability ticket	تذكرة الإمكانات
cardholder	حامل البطاقة
Certification authority (CA)	سلطة التوثيق (التصديق)
Cipher Block Chaining (CBC) mode	نمط سَلسَلة كتل الشفرة (CBC)
Cipher FeedBack (CFB) mode	نمط التغذية المرتجعة للشفرة (CFB)
ciphertext	رسالة مشفَّرة
circuit-level gateway	بوابة مستوى الدوائر
Common Criteria (CC)	المعايير المشتركة
community	التَجمُّع
community name	اسم التَجمُّع
cryptanalysis	تحليل الشفرة
data confidentiality	سرية البيانات
Data Encryption Standard (DES)	معيار تشفير البيانات (DES)
data integrity	سلامة البيانات
decryption	إزالة التشفير
denial of service	الحرمان من الخدمة، حجب الخدمة
detached signature	توقيع منفصل
Diffie-Hellman key exchange	تبادل المفاتيح بطريقة ديفي - هيلمان
digital immune system	نظام مناعة رقمي
digital signature	توقيع رقمي
Digital Signature Standard (DSS)	معيار التوقيع الرقمي
direct DDoS attack	هجوم حجب الخدمة الموزَّع المباشر
Distributed Denial of Service (DDoS)	حجب الخدمة الموزَّع
downloaders	برامج التنزيل

dual signature	توقيع مزدوج
Electronic CodeBook (ECB) mode	نمط دفتر الشفرة الإلكتروني (ECB)
electronic mail	البريد الإلكتروني
Elliptic-Curve Cryptography (ECC)	تشفير بالمنحنى الإهليلجي
e-mail virus	فيروس البريد الإلكتروني
Encapsulating Security Payload (ESP)	التَّغليف الأمني للحمولة
encryption	التشفير
end-to-end encryption	التشفير من طرف إلى طرف
exploits	استغلالات (نقاط ضعف)
Feistel cipher	هيكل فيستل للتشفير
flooder	مُرسِل الفيضان
HMAC	خوارزمية HMAC لتوثيق الرسائل
honeypot	مصيدة أمنية
integrity	السلامة
Internet Security Association and Key Management Protocol (ISAKMP)	جمعية أمن الإنترنت وبروتوكول إدارة المفاتيح
intruder	المتسلل، المهاجم
intrusion detection	اكتشاف التسلل، اكتشاف الاختراق الامني
intrusion detection exchange format	صيغة تبادل البيانات لاكتشاف التسلل
IP Security (IPSec)	بروتوكول أمن الإنترنت
issuer	مُصدِر
Kerberos realm	عالم كيربيروس
Kerberos system	نظام كيربيروس

key distribution	توزيع المفاتيح
key exchange	تبادل المفاتيح
key localization	توطين المفاتيح
keyloggers	مسجلات ضربات المفاتيح
kit	علبة أدوات (لتوليد فيروسات)
lifetime	فترة الصلاحية
link encryption	تشفير الوصلة
logic bomb	قنبلة منطقية
macro virus	فيروس الماكرو
malicious software (malware)	برمجيات مؤذية
Management Information Base (MIB)	قاعدة معلومات الإدارة
management station	محطة الإدارة
masquerade	هجوم التنكر
MD5	خوارزمية MD5 لخلاصة الرسالة
merchant	تاجر
message authentication	توثيق الرسالة
Message Authentication Code (MAC)	كود توثيق الرسالة
message digest	خلاصة الرسالة
message processing model	نموذج معالجة الرسالة
multilevel security	الأمن متعدد المستويات
Multipurpose Internet Mail Extensions (MIME)	امتدادات بريد الإنترنت متعددة الأغراض
network management proxy	مُفوَّض إدارة الشبكة
nonce	عدد يُستخدَم لمرة واحدة
non-repudiation	عدم التنصل (عدم الإنكار)

Oakley key determination protocol	بروتوكول أوكلي لتعيين المفاتيح
object	كائن
one-way hash function	دالة تحوير أحادية الاتجاه
OSI security architecture	البنية الأمنية للربط بين الأنظمة المفتوحة
packet-filtering router	موجه ترشيح الرزم
passive attack	الهجوم السلبي، التهديد المستتر
password	كلمة السر، كلمة المرور
payment gateway	بوابة الدفع
plaintext	الرسالة الأصلية
polymorphic virus	فيروس متعدد الأشكال
Pretty Good Privacy (PGP)	خصوصية جيدة جداً
private key	مفتاح خاص
Propagating Cipher Block Chaining (PCBC) mode	نمط سَلسَلة الكتل للشفرة المنتشرة
public key	مفتاح عام
public-key certificate	شهادة المفتاح العام
public-key encryption	تشفير بالمفاتيح العامة
radix-64	الأساس 64
realm	عالم
reference monitor	المراقب المرجع
reflector DDoS attack	هجوم DDoS الانعكاسي
replay	إعادة التشغيل، إعادة الإرسال
replay attack	هجوم إعادة الإرسال
rootkit	الجذور الخفية (أو أدوات الجذر)
RSA	خوارزمية RSA
rule-based intrusion detection	اكتشاف التسلل بناءً على قواعد

salt	الملح
secret key	مفتاح سرّي
Secure Electronic Transaction (SET)	المُعامَلات الإلكترونية الآمنة (SET)
secure hash function	دالة تحوير آمنة
Secure Multipurpose Internet Mail Extensions (S/MIME)	امتدادات بريد الإنترنت متعددة الأغراض الآمنة
Secure Socket Layer (SSL)	طبقة المقابس الآمنة (SSL)
Security Association (SA)	الارتباط الأمني
security attacks	الهجمات الأمنية
security mechanisms	الآليات الأمنية
security services	الخدمات الأمنية
sequence number	رقم تسلسل
session key	مفتاح الجلسة
SHA-1	خوارزمية SHA-1
Simple Network Management Protocol (SNMP)	بروتوكول إدارة الشبكة البسيط
spammer program	برنامج إرسال رسائل الدعاية
stateful inspection firewall	جدران تفتيش الحالة
statistical anomaly detection	اكتشاف الشذوذ الإحصائي
stealth virus	فيروس مُتَخفٍّ
stream cipher	تشفير التدفق
strong collision resistance	مقاومة قوية للتصادمات
subject	الفاعل
subkey	مفتاح فرعي
symmetric encryption	التشفير المتماثل
ticket	تذكرة
Ticket-Granting Server (TGS)	خادم منح التذاكر

traffic analysis	تحليل حركة المرور
Transport Layer Security (TLS)	أمن طبقة النقل (TLS)
transport mode	نمط النقل
trapdoor	باب المصيدة (المدخل الخلفي)
Triple DES (3DES)	خوارزمية DES الثلاثية (3DES)
Trojan horse	حصان طروادة
trust	ثقة
trusted system	نظام موثوق فيه
tunnel mode	نمط النفق
user security model	نموذج أمن المستخدم
virus	فيروس
weak collision resistance	مقاومة ضعيفة للتصادمات
worm	دودة
X.509 certificate	شهادة X.509
ZIP	خوارزمية ضغط البيانات ZIP
zombie	برنامج زومبي

اختصارات

3DES	Triple Data Encryption Standard
AES	Advanced Encryption Standard
AH	Authentication Header
ANSI	American National Standards Institute
CBC	Cipher Block Chaining
CC	Common Criteria
CESG	Communications-Electronics Security Group
CFB	Cipher Feedback
CMAC	Cipher-Based Message Authentication Code
CRT	Chinese Remainder Theorem
DDoS	Distributed Denial of Service
DES	Data Encryption Standard
DoS	Denial of Service
DSA	Digital Signature Algorithm
DSS	Digital Signature Standard
ECB	Electronic Codebook
ESP	Encapsulating Security Payload
FIPS	Federal Information Processing Standard
IAB	Internet Architecture Board
IETF	Internet Engineering Task Force
IP	Internet Protocol
IPSec	IP Security
ISO	International Organization for Standardization
ITU	International Telecommunication Union
ITU-T	ITU Telecommunication Standardization Sector
IV	Initialization Vector
KDC	Key Distribution Center

LAN	Local Area Network
MAC	Message Authentication Code
MD5	Message Digest,Version 5
MIC	Message Integrity Code
MIME	Multipurpose Internet Mail Extension
MTU	Maximum Transmission Unit
NIST	National Institute of Standards and Technology
NSA	National Security Agency
OFB	Output Feedback
PCBC	Propagating Cipher Block Chaining
PGP	Pretty Good Privacy
PKI	Public Key Infrastructure
PRNG	Pseudorandom Number Generator
RFC	Request for Comments
RNG	Random Number Generator
RSA	Rivest-Shamir-Adelman
SET	Secure Electronic Transaction
SHA	Secure Hash Algorithm
SHS	Secure Hash Standard
S/MIME	Secure MIME
SNMP	Simple Network Management Protocol
SNMPv3	Simple Network Management Protocol Version 3
SSL	Secure Sockets Layer
TCP	Transmission Control Protocol
TLS	Transport Layer Security
UDP	User Datagram Protocol
WAN	Wide Area Network