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

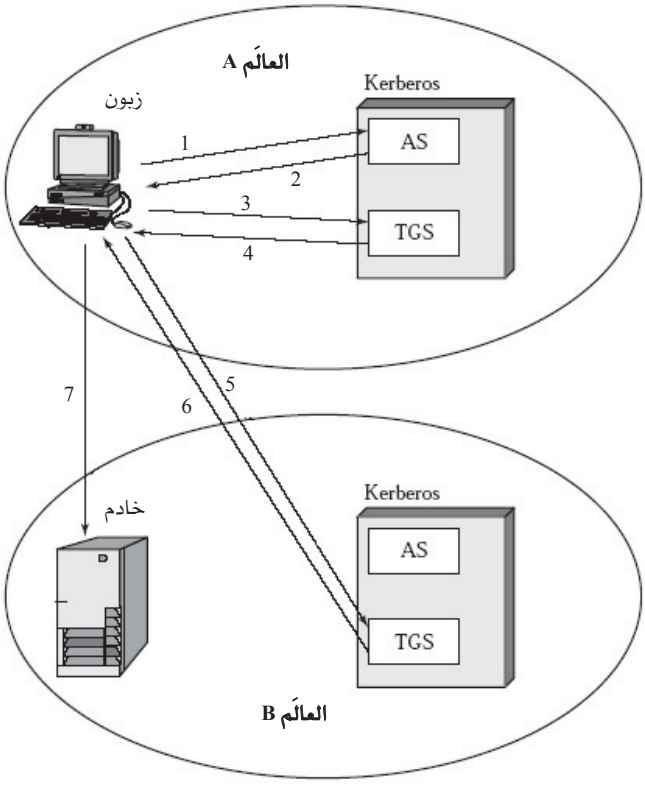
(5) $C \rightarrow TGS_{rem}$: $ID_{vrem} \| Ticket_{tgsrem} \| Authenticator_c$

(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

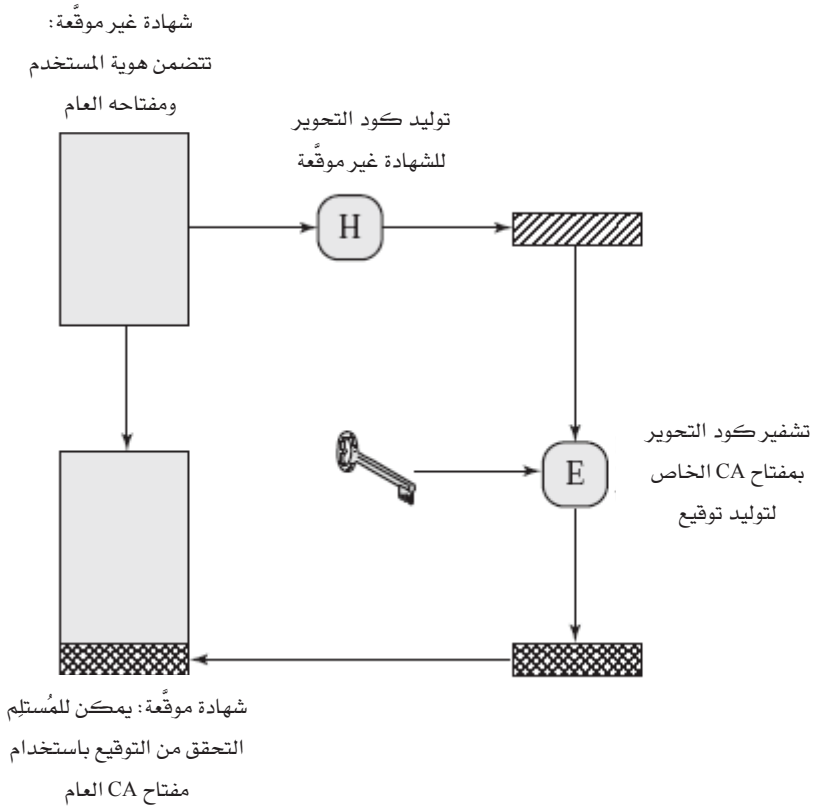
تحويل اسم المستخدم إلى عنوان شبكة، بالإضافة إلى خواص ومعلومات أخرى عن المستخدمين.

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.

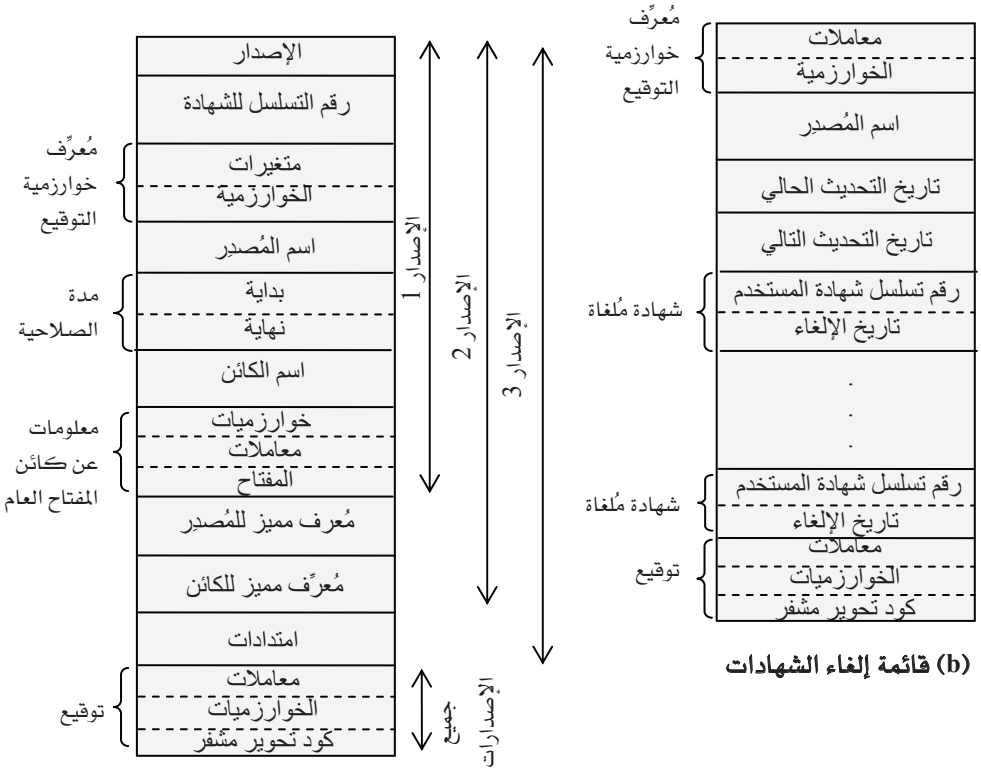
4-2-1 الشهادات

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:

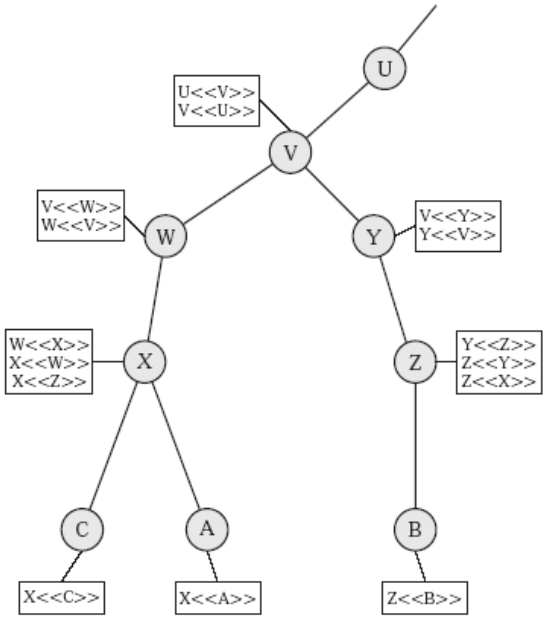
1. افتراض أن مفتاح المستخدم الخاص قد تعرض لخطر (مثلاً أصبح معروفاً للمهاجم).
2. لم يعد المستخدم مُوَثَّقاً من قِبَل تلك الهيئة CA.
3. افتراض أن شهادة CA قد تعرضت لخطر.

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.

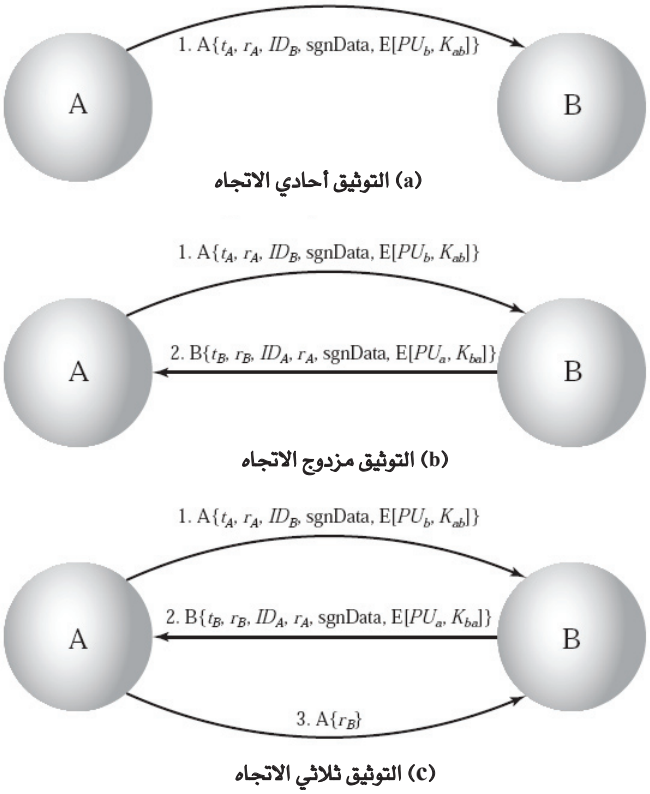
يشمل كل مُدخَل بالقائمة رقم التسلسل للشهادة المُلغاة وتاريخ إلغائها. ونظراً لأن أرقام التسلسل فريدة ضمن هيئة الشهادات، يكون رقم التسلسل كافياً لتمييز الشهادة. عندما يستلم مُستخدِمٌ شهادة في رسالة، يجب أن يقرّر ما إذا كانت تلك الشهادة مُلغاة أو لا. يمكن أن يفحص المستخدم الدليل كل مرة يستلم شهادة. لتفادى 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:

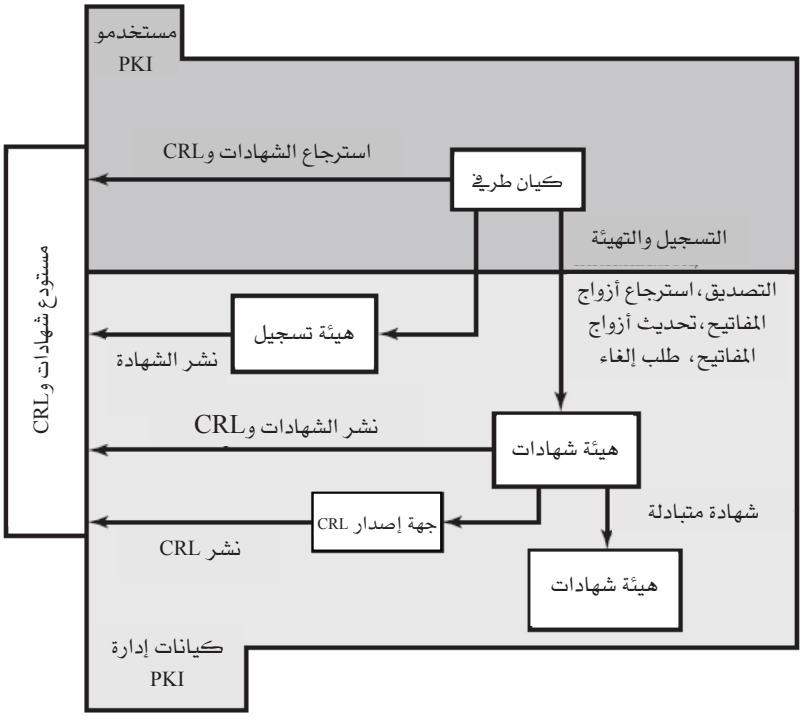
- الاسم البديل للكائن: يتضمن اسماً بديلاً أو أكثر بعدة صيغ مختلفة. هذا الحقل مهم لدعم بعض التطبيقات التي قد تستخدم صيغاً خاصة بها للأسماء (كالبريد الإلكتروني، وتبادل البيانات الإلكتروني (EDI)، وبروتوكول IPSec).
- الاسم البديل للمُصدِر: يتضمن اسماً بديلاً أو أكثر بعدة صيغ مختلفة.
- سمات دليل الكائن: يتضمن 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:

$$b[0] = \text{bit } 0 \text{ of } s[0]$$

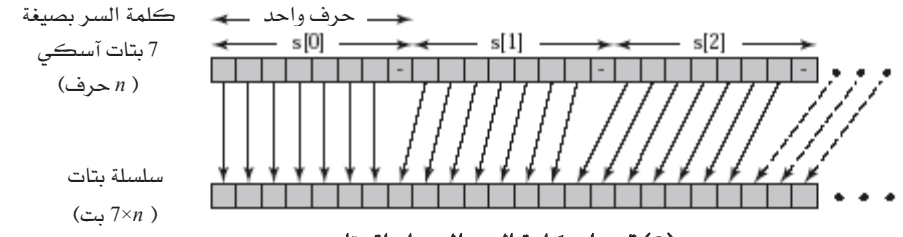
$$\dots$$

$$b[6] = \text{bit } 6 \text{ of } s[0]$$

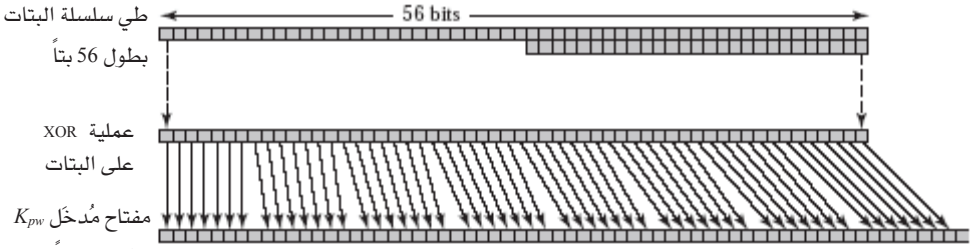
$$b[7] = \text{bit } 0 \text{ of } s[1]$$

$$\dots$$

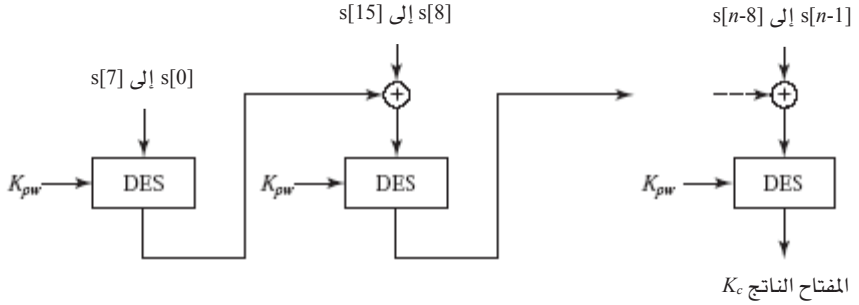
$$b[7i+m] = \text{bit } m \text{ of } s[i],\ 0 \le m \le 6$$



(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)

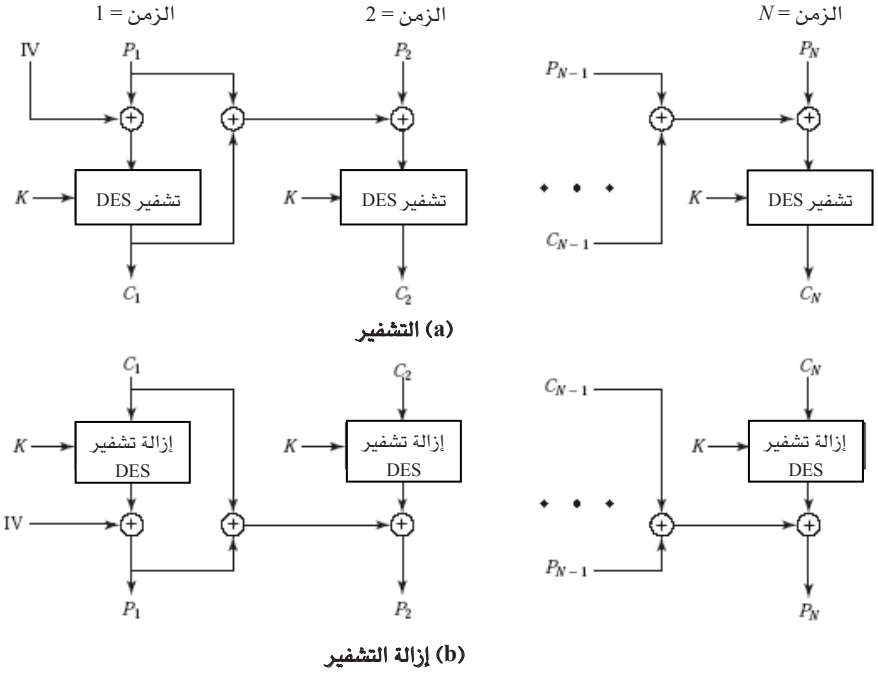
تذكَّر من الفصل الثاني أنه في نمط CBC لخوارزمية DES يتكون المُدخل في كل مرحلة من XOR للكتلة الحالية من النص غير المُشفَّر مع الكتلة السابقة من النص المُشفَّر، وأن نفس المفتاح يُستخدم في كل مرحلة (انظر الشكل 2-9). يتميز هذا النمط على نمط ECB (والذي يتم فيه تشفير كل كتلة بشكلٍ مستقل) بما يأتي: في CBC تُنتج 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).