5

الفصل الخامس

أمن البريد الإلكتروني

محتويات الفصل:

"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.

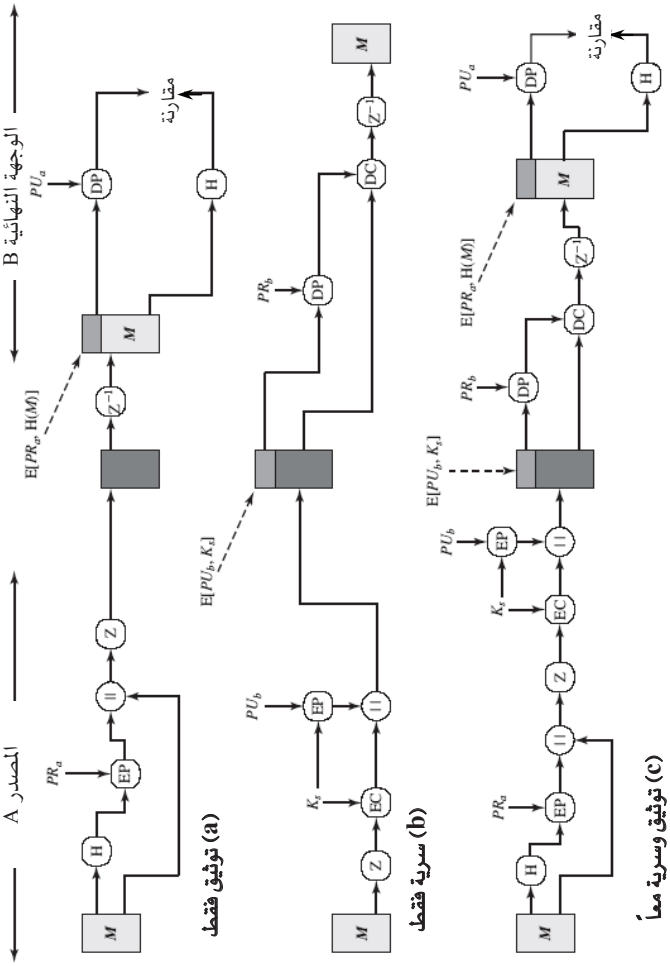
❖ التوثيق:

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

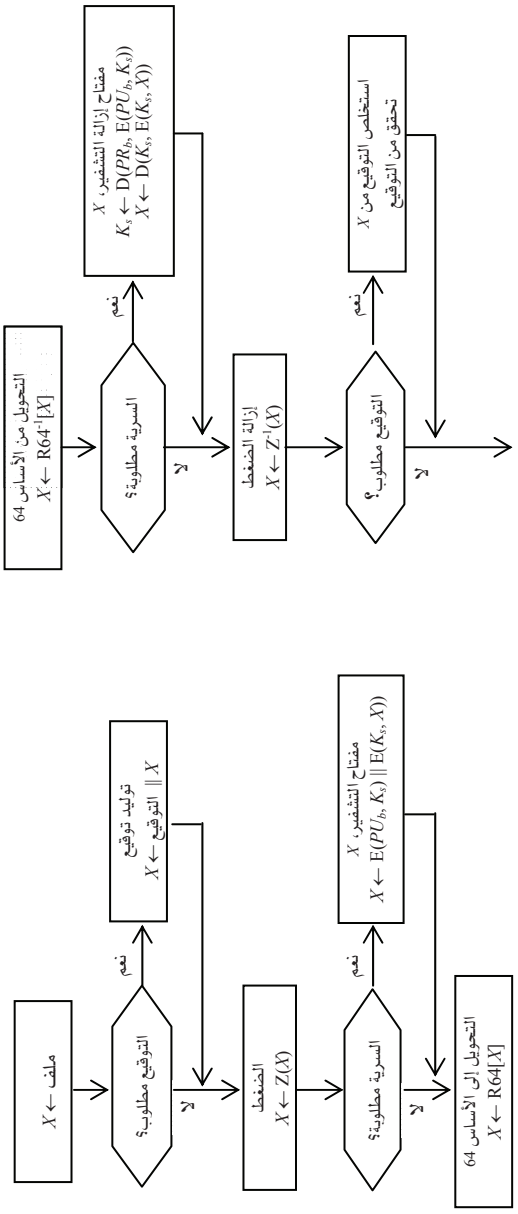
64. على سبيل المثال، يذكر [HELD96] أن متوسط نسبة الضغط باستخدام خوارزمية ZIP يبلغ حوالى 2.0. إذا أهملنا أجزاء التوقيع والمفتاح لصغرها نسبياً، فإن التأثير الكلي لضغط ملفٍ طوله X وإزالة ضغطه هو جعل طوله

$$1.33 \times 0.5 \times X = 0.665 \times X$$

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.

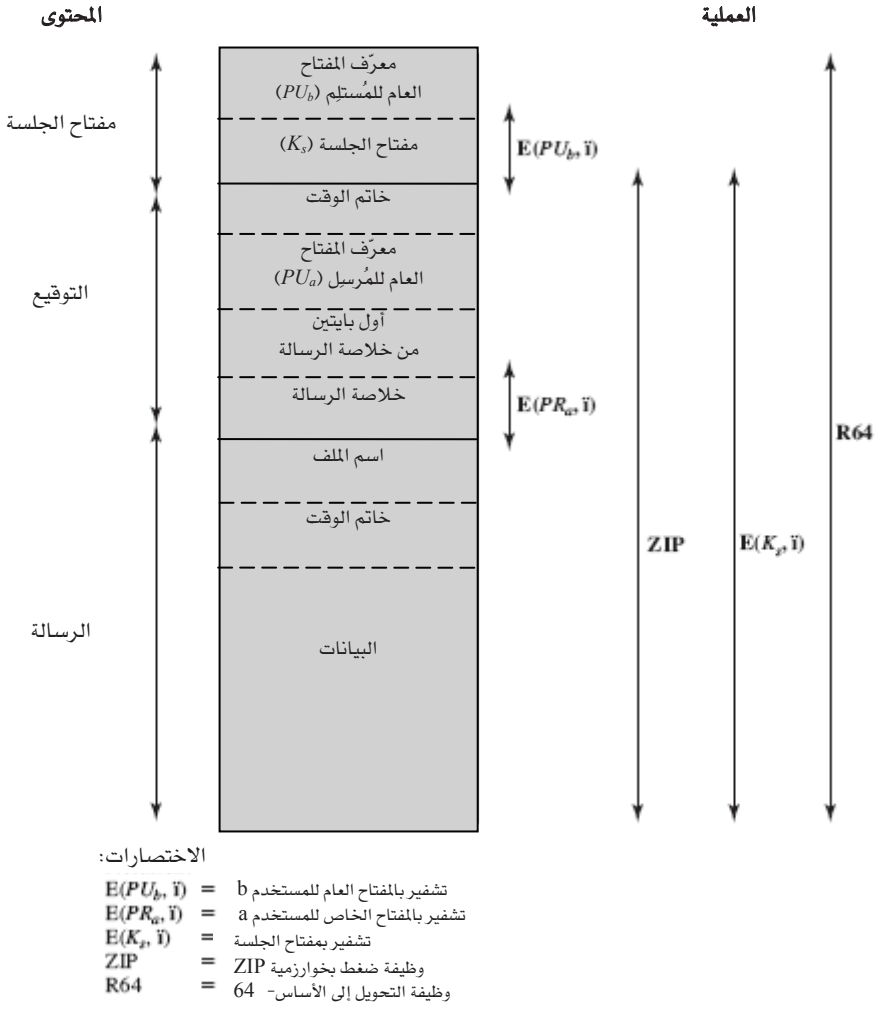
المُستلِم من أن معرِّف المفتاح يخص مفتاحاً عاماً يعرفه لذلك المُرسِل ثم يواصل عملية التحقق من التوقيع.

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.

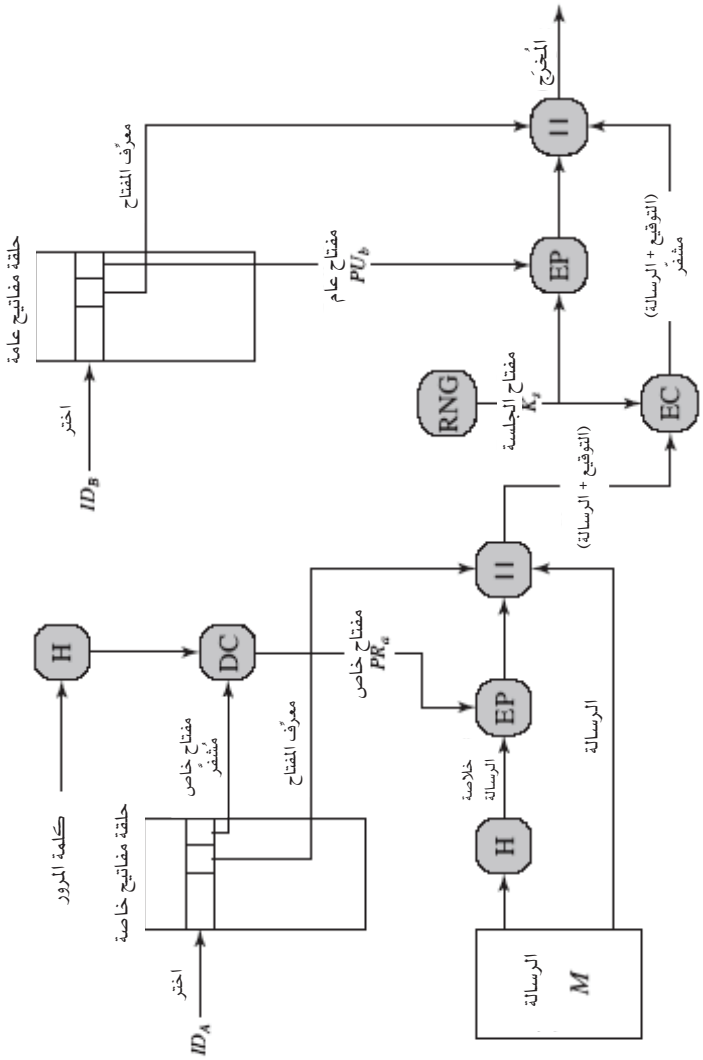
c. عندئذٍ يستخرج PGP مفتاح الجلسة ويستخدمه لإزالة تشفير الرسالة.

2. توثيق الرسالة

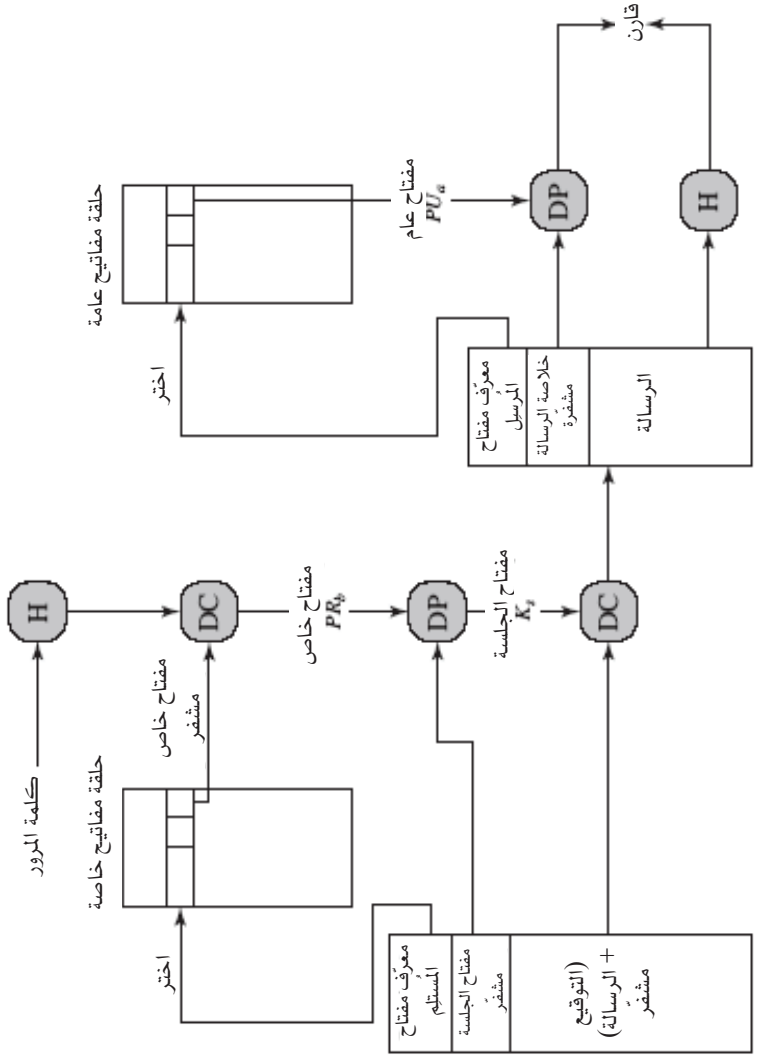
a. يستخرج PGP المفتاح العام للمُستلِم من حلقة المفاتيح العامة، وذلك باستخدام حقل مُعرِّف المفتاح في جزء مفتاح التوقيع ضمن الرسالة كمؤشر للفهرس.

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.

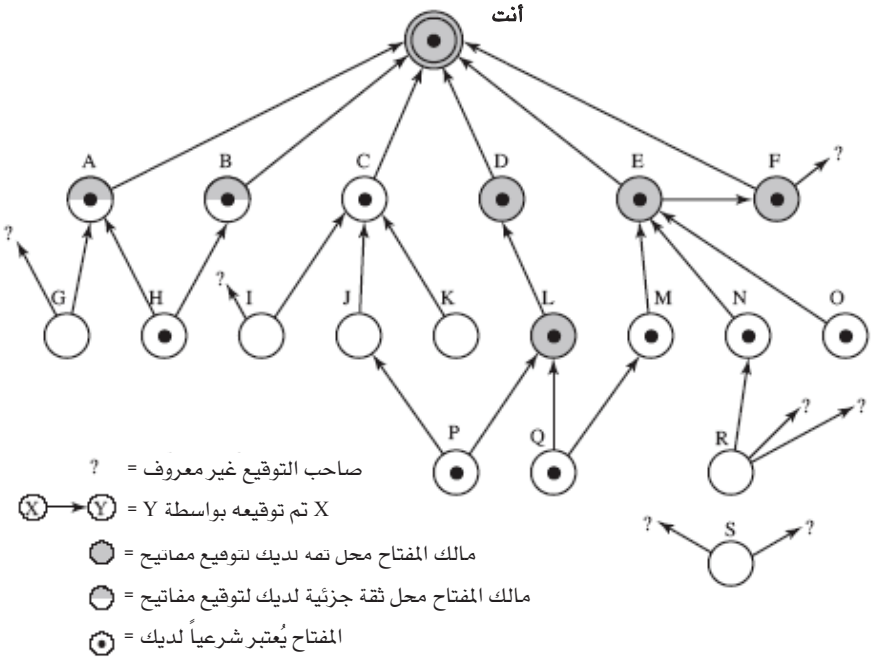
يوضّح الشكل 5-7 مثالاً لطريقة الربط ما بين الثقة في التوقيع وشرعية المفتاح.[2] ويبيّن الشكل هيكل حلقة مفاتيح عامة تُمكّن مستخدمها من تحصيل عددٍ من المفاتيح العامة، بعضها مباشرةً من مالكيها وبعضها الآخر من طرف ثالث كخادم مفاتيح (key server) مثلاً.

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.

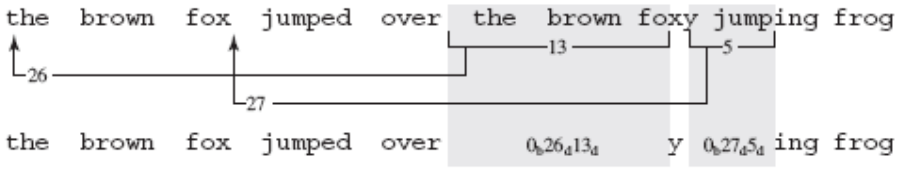
الأجزاء المتبقية من الرسالة المضغوطة هي الحرف y؛ والتسلسل $<00_b> <27_d> <5_d>$ يحل محل التسلسل الذي يضم حرف المسافة تليه jump؛ وتسلسل الحروف ing frog.

[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.



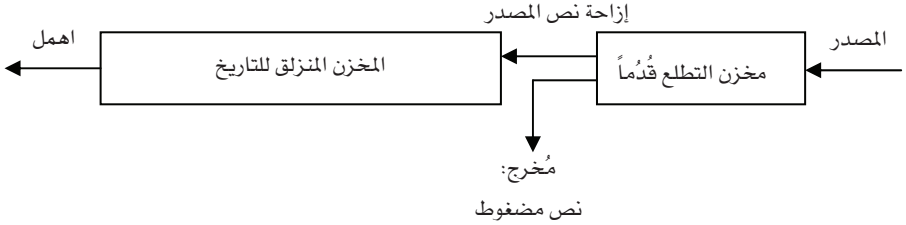
الشكل 5-9: مثال على نظام LZ77.

خوارزمية الضغط

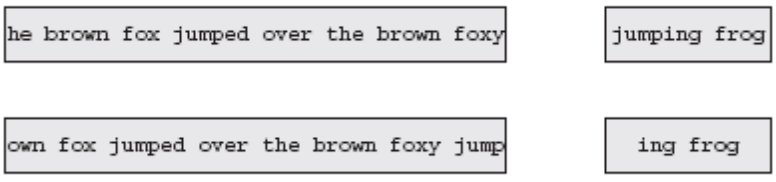
تستخدم خوارزمية الضغط LZ77 ومشتقاتها اثنين من المخازن المؤقتة (buffers). يضم المخزن المنزلق للتاريخ (sliding history buffer) آخر N حرف تم معالجتها من بيانات المصدر، بينما يضم مخزن التطلع قُدُماً (look-ahead buffer) الـ L حرف التالية (انظر الشكل 5-10 (a)). تحاول الخوارزمية إيجاد مطابقة بين اثنين أو أكثر 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

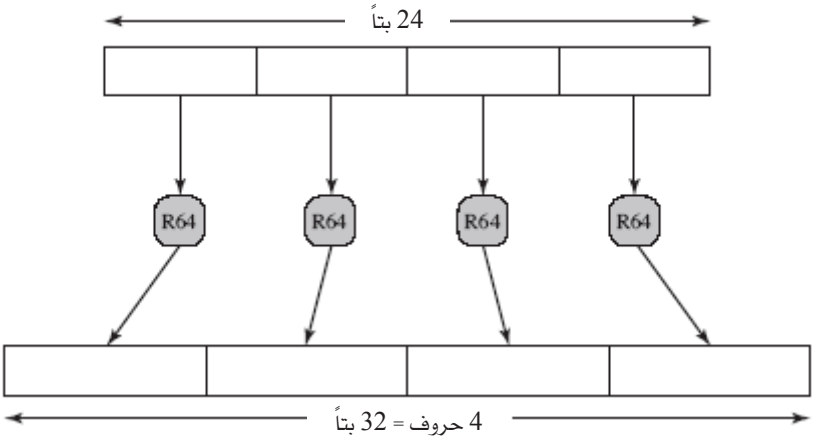
يستخدم كلٌّ من PGP وS/MIME أسلوب تكويد يُعرف بالتحويل للأساس 64، حيث يتم استبدال البيانات الثنائية الداخلة ذات القيم الاعتباطية بمُخرج يمثل التسلسل المقابل من الحروف القابلة للطباعة. تتسم عملية التكويد المستخدمة بالخصائص الآتية:

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؟]

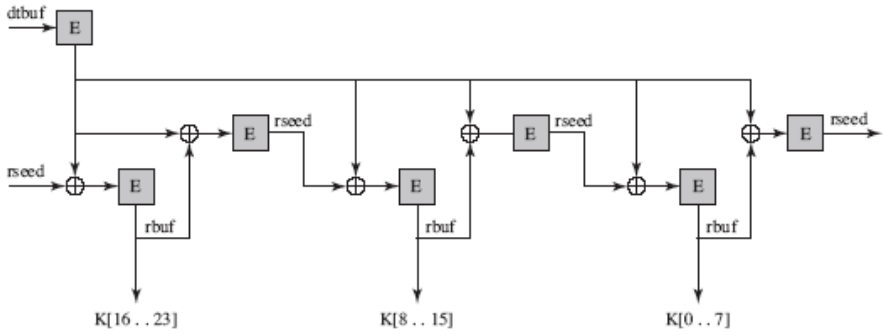
إذا كان k = 0 اذهب إلى G9 وإلا ضع k-1 ← k واذهب إلى G4.

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.