الفصل الثالث

3 التشفير بالمفاتيح العامة وتوثيق الرسائل

محتويات الفصل:

3-1 طرق توثيق الرسائل

3-2 دوال التحوير الآمنة وخوارزمية HMAC

3-3 مبادئ التشفير بالمفاتيح العامة

3-4 خوارزميات التشفير بالمفاتيح العامة

3-5 التوقيع الرقمي

3-6 إدارة المفاتيح

3-7 توصيات للمطالعة

3-8 مصادر للمعلومات على الويب

3-9 مصطلحات رئيسة

3-10 أسئلة للمراجعة ومسائل

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

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

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

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

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

1-3 طرق توثيق الرسائل

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

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

3-1-1 التوثيق باستخدام التشفير التقليدي

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

3-1-2 التوثيق بدون تشفير الرسالة

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

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

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

❖ كود توثيق الرسالة:

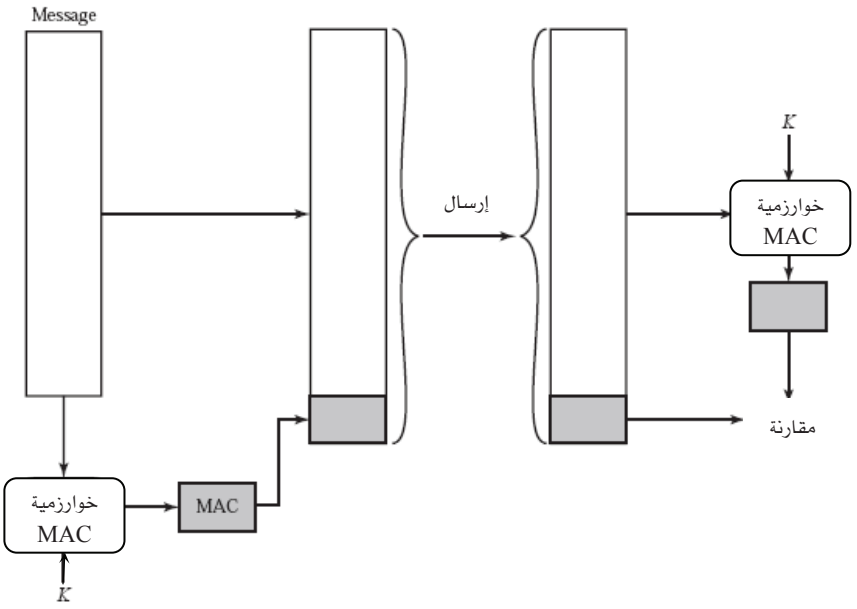
يتضمن التوثيق استخدام مفتاح سري لتوليد كتلة صغيرة من البيانات تُعرَف باسم "كود توثيق الرسالة" (Message Authentication Code (MAC)) تذيّل به الرسالة. ويفترض هذا الأسلوب أن الطرفين المتصلين A وB يستخدمان مفتاحاً سرياً مشتركاً K_{AB}. عندما يريد الطرف A إرسال رسالة إلى B، يقوم بحساب كود توثيق الرسالة من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

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

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

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

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



الشكل 3-1: التوثيق باستخدام كود توثيق الرسالة.

❖ دالة التحوير أحادية الاتجاه:

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

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

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

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

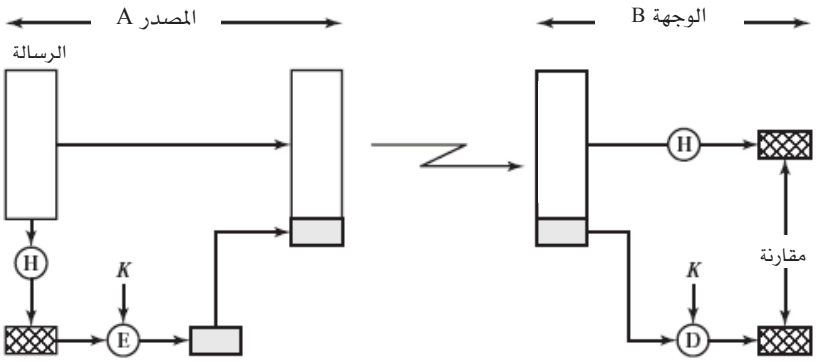
- برامج التشفير بطيئة جداً. بالرغم من صغر كمية البيانات التي تحتاج للتشفير في كل رسالة، فقد يكون هناك سيل متواصل من الرسائل الواردة للنظام والخارجة منه.
- كلفة المكوّنات المادية (العتاد) لأجهزة التشفير ليست بسيطة. وعلى الرغم من توفر شرائح إلكترونية رخيصة لتنفيذ خوارزمية DES، فإن محصّلة الكلفة ستكون عالية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

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

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

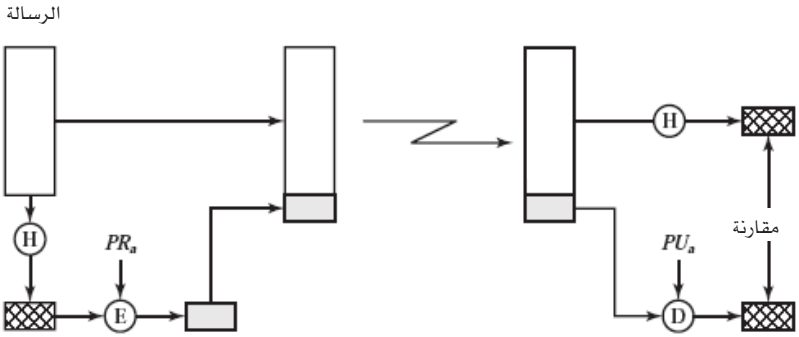
(يعني الرمز || عملية الوصل (concatenation))، ثم يرسل $[M||M_{DM}]$ إلى B. ولأن الطرف B يعرف S_{AB} فسيمكنه حساب $H(S_{AB}||M)$ للرسالة التي وصلته، ومن ثم التحقّق من MD_M. ولأنه لم يتم إرسال القيمة السرية نفسها فلن يتمكن المهاجم من تعديل الرسالة التي يعترضها. وطالما ظلت القيمة سرية، فلن يتمكن المهاجم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

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

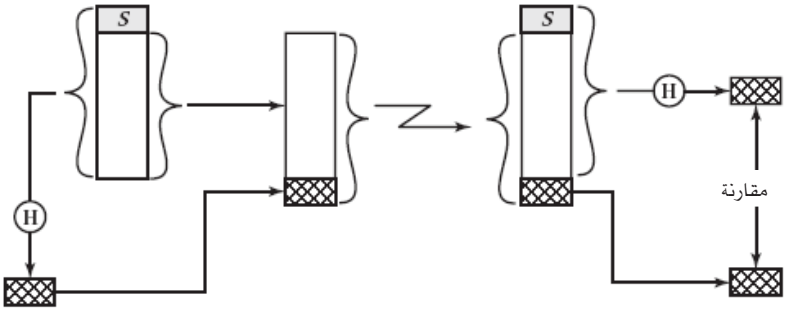


(a) استخدام التشفير التقليدي

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



(b) استخدام التشفير بالمفاتيح العامة



(c) استخدام قيمة سرية

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

2-3 دوال التحوير الآمنة وخوارزمية HMAC

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

1-2-3 المتطلبات من دالة التحوير

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

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

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

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

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

أولاً: ملاحظة أو اعتراض الرسالة والقيمة المُشفَّرة لكود التحوير.

ثانياً: توليد قيمة غير مُشفَّرة لكود التحوير من الرسالة.

ثالثاً: توليد رسالة بديلة لها نفس قيمة كود التحوير.

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

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

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

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

3-2-2 دوال تحوير بسيطة

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

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

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

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

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

حيث:

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

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

b_{ij} = البت i في كتلة البتات j.

$\oplus$ = دالة XOR.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

3-2-3 دالة التحوير الآمنة SHA-1

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

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

الجدول 3-1 مقارنة متغيرات SHA.

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

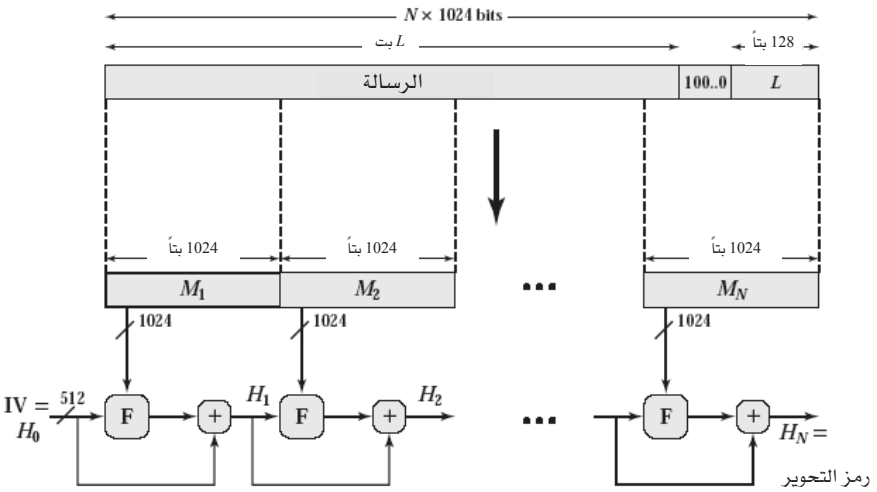
ملاحظات:

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

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

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

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



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

الخطوة 2: إضافة طول الرسالة.

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

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

الخطوة 3: تهيئة المخزن المؤقت لكود التحوير.

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

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

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

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

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

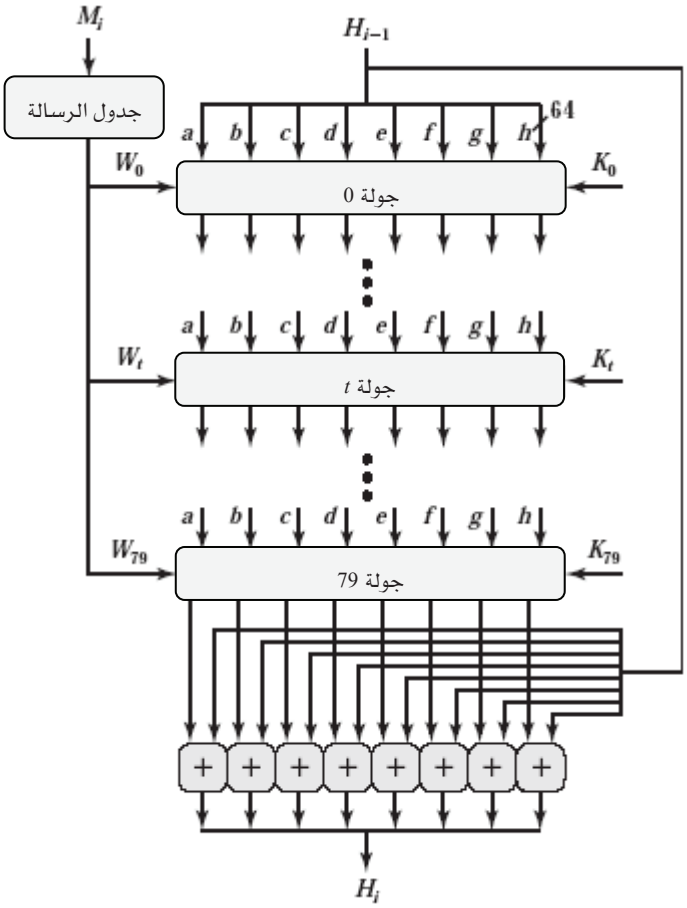
للجذور التكعيبية للأعداد الأولية الثمانين الأولى. توفر تلك الثوابت مجموعة "عشوائية" من أنماط من 64 بتاً، والتي يُتوقع أن تُزيل أي انتظام (نمطية) قد يوجد في البيانات الداخلة.

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

الخطوة 5: إخراج الناتج.

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

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



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

4-2-3 دوال التحوير الآمنة الأخرى

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

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

❖ خوارزمية خلاصة الرسالة (MD5):

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

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

❖ خوارزمية الدوّامة (Whirlpool):

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

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

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

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

5-2-3 خوارزمية HMAC

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

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

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

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

❖ الأهداف التصميمية لخوارزمية HMAC

يسرد طلب التعليقات RFC 2104 الأهداف التصميمية لخوارزمية HMAC كالآتي:

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

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

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

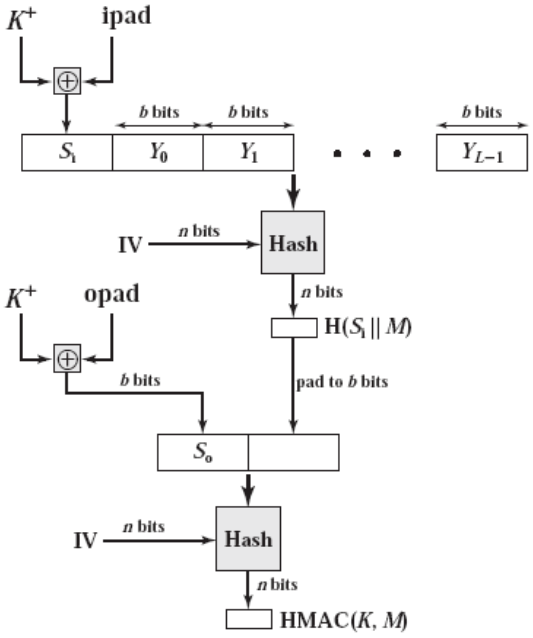
في الحقيقة، يُعدُّ هدف التصميم الأخير في القائمة السابقة الميزة الرئيسة لـ HMAC على كل الأساليب الأخرى التي اقتُرحت. يمكن إثبات أن خوارزمية HMAC آمنة بشرط توفر بعض جوانب القوة التشفيرية لدالة التحوير المضمّنة بدرجة معقولة. سنعود لهذه النقطة لاحقاً في هذا الجزء، لكننا سنتناول أولاً هيكل خوارزمية HMAC.

❖ الخوارزمية

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

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

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



الشكل 3-6: تركيبة HMAC.

يمكن التعبير عن الـ HMAC على النحو الآتي:

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

يمكن التعبير عن ذلك بشكلٍ آخر كما يأتي:

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

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

3-3 مبادئ التشفير بالمفاتيح العامة

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

3-3-1 تركيبة التشفير بالمفاتيح العامة

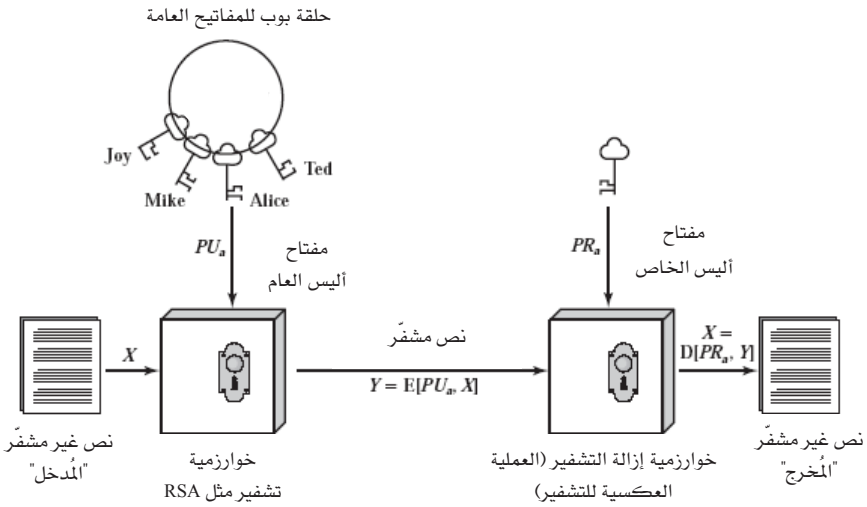
يُعدُّ التشفير بالمفاتيح العامة – الذي اقتُرح أولاً للجمهور من قِبَل ديفي وهيلمان في عام 1976 [DIFF76] – أول تقدم ثوري حقيقي في التشفير على مدى آلاف السنين. تعتمد خوارزميات المفاتيح العامة على دوال رياضية بدلاً من العمليات البسيطة على أنماط البتات كالتي تستخدم في خوارزميات التشفير المتماثل. كما أن التشفير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

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

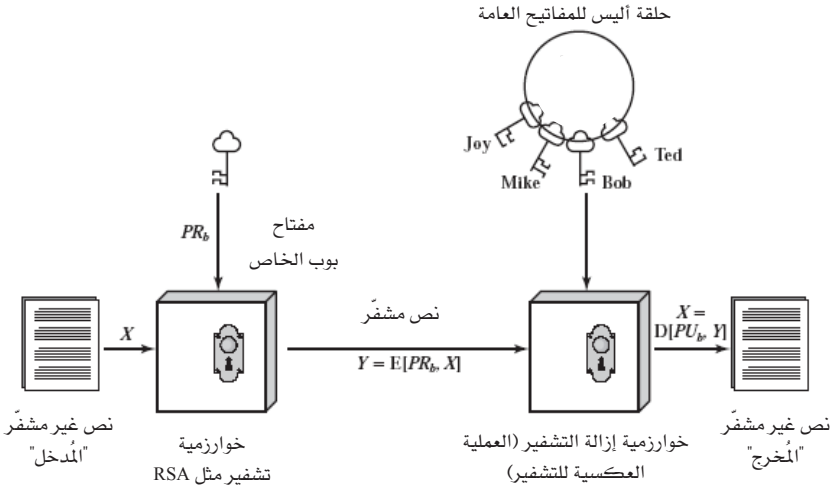
متقادمة. على عكس ذلك، بسبب عبء الحسابات الإضافية في الأساليب الحالية للتشفير بالمفاتيح العامة، لا يبدو في الأفق أي احتمال لإلغاء طرق التشفير التقليدية في المستقبل القريب. وأخيراً، هناك شعور بأن توزيع المفاتيح أمر بسيط عند استخدام التشفير بالمفاتيح العامة، مقارنةً بالمتاعب المرتبطة بالمصافحة المطلوبة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

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

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



(a) تشفير



(b) توثيق

الشكل 3-7: التشفير بالمفاتيح العامة.

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

تشمل خوارزمية التشفير بالمفاتيح العامة الخطوات الأساسية الآتية:

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

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

2-3-3 تطبيقات أنظمة التشفير بالمفاتيح العامة

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

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

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

الجدول 3-2: تطبيقات أنظمة التشفير بالمفاتيح العامة.

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

3-3-3 متطلبات التشفير بالمفاتيح العامة

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

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

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

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

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

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

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

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

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

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

4-3 خوارزميات التشفير بالمفاتيح العامة

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

1-4-3 خوارزمية RSA للتشفير بالمفاتيح العامة

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

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

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

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

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

توليد المفاتيح

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

احسب $n = p \times q$

احسب $\phi(n) = (p-1)(q-1)$

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

احسب d بحيث $de \bmod \phi(n) = 1$

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

المفتاح الخاص: $KR=\{d, n\}$

التشفير

النص غير المشفّر: $M < n$

النص المشفّر: $C = M^e \bmod n$

إزالة التشفير

النص المشفّر: C

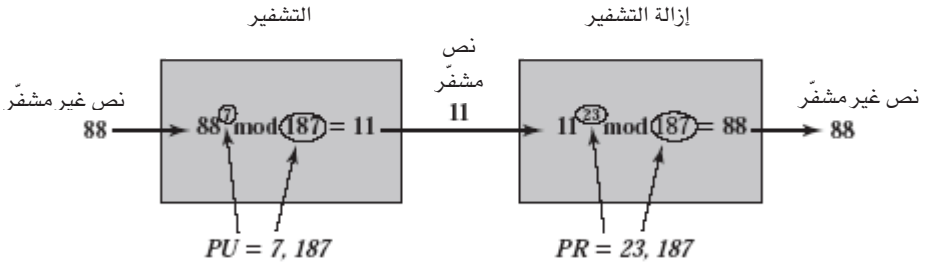
النص غير المشفّر: $M = C^d \bmod n$

الشكل 3-8: خوارزمية RSA.

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

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

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



الشكل 3-9: مثال لخوارزمية RSA.

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

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

ولإزالة التشفير نقوم بالآتي:

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

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

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

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

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

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

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

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

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

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

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

$b = a^i \bmod p$

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

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

❖ الخوارزمية:

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

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

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

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

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

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

ويحسب المستخدم B المفتاح من

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

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

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

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

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

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

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

العناصر العمومية للجميع

q عدد أولي

α عدد أقل من q ويمثل جذراً بدائياً لـ q

توليد مفتاح المستخدم A

اختر عدداً خاصاً X_A أقل من q

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

توليد مفتاح المستخدم B

اختر عدداً خاصاً X_B أقل من q

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

توليد المفتاح السري للمستخدم A

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

توليد المفتاح السري للمستخدم B

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

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

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

الطرف A:

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

الطرف B:

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

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

الطرف A:

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

الطرف B:

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

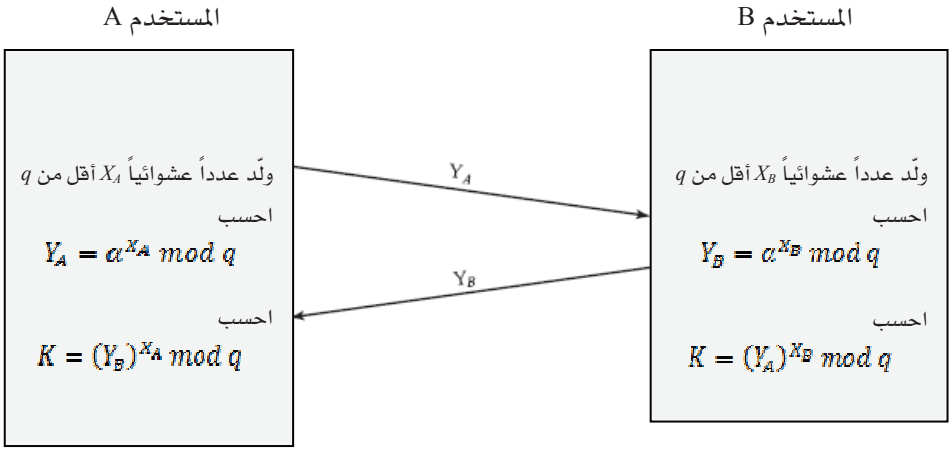
افترض أن مهاجماً E توفرت لديه المعلومات الآتية:

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

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

❖ بروتوكولات تبادل المفاتيح:

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



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

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

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

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

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

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

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

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

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

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

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

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

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

3-4-3 الخوارزميات الأخرى للتشفير بالمفاتيح العامة

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

❖ معيار التوقيع الرقمي (DSS):

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

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

❖ التشفير بالمنحنى الإهليليجي (ECC):

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

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

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

5-3 التوقيع الرقمي

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

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

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

6-3 إدارة المفاتيح

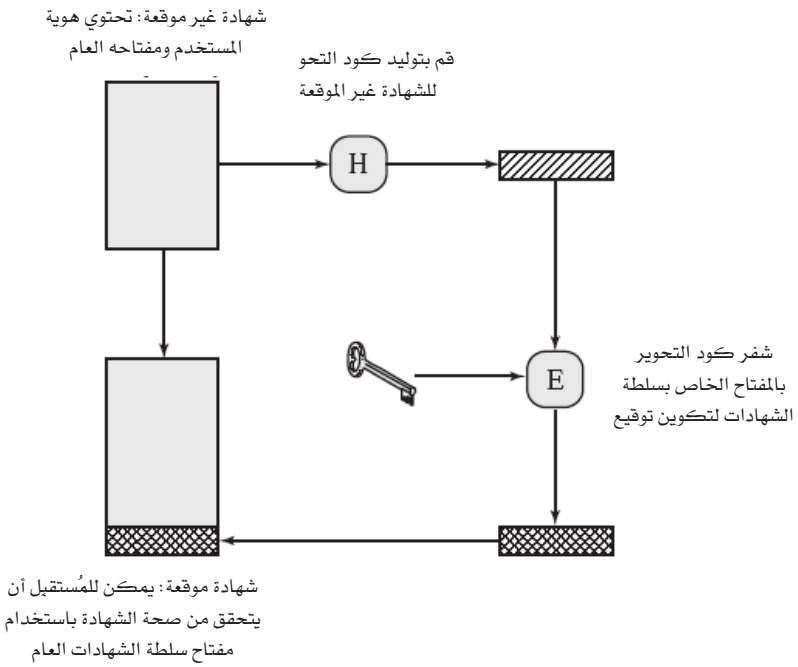
أحد الأدوار الرئيسة للتشفير بالمفاتيح العامة هو تناوله لمسألة توزيع المفاتيح. في الحقيقة هناك جانبان محدَّدان لاستخدام التشفير بالمفاتيح العامة لهذا الغرض:

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

سنتناول كلاً من هذين الموضوعين تباعاً.

1-6-3 شهادات المفاتيح العامة

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



الشكل 3-12: استخدام شهادة المفتاح العام.

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

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

3-6-2 استخدام لمفاتيح العامة لتوزيع المفاتيح السرية

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

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

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

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

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

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

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

7-3 توصيات للمطالعة

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

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

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

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

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

8-3 مصادر للمعلومات على الويب

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

9-3 مصطلحات رئيسة

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

10-3 أسئلة للمراجعة ومسائل

1-10-3 أسئلة للمراجعة

3-1 اذكر ثلاث طرق لتوثيق الرسائل.

3-2 عرِّف كود توثيق الرسائل.

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

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

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

3-6 ما العناصر الرئيسة لنظام التشفير بالمفاتيح العامة؟

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

3-8 ما الفرق بين المفتاح الخاص والمفتاح السرّي؟

2-10-3 مسائل

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

بعد تكوين الجداول الثلاث يتم إعلانها للجميع.

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

M1=

5
4
3
2
1

M2=

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

M3=

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

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

c. ناقش سبب كون ذلك الأسلوب آمناً.

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

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

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

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

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

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

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

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

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

a. اختر عدداً فردياً E.

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

c. اضرب P، Q للحصول على N.

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

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

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

وضح أن هذا الأسلوب ليس آمناً وذلك بحل المسألة الآتية:

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

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

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

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

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

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

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

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

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