فصل –1 المقدمة والتشفير التقليدي

1.1 التشفير Cryptography

- يعرّف التشفير وفقا لقاموس اوكسفورد (2006) على أنه "**فن كتابة الشفرات او حلها**".
  - ✓ يركز هذا التعريف على **مسألة الاتصلات السرية** secret communications فقط ويهمل بقية التطبيقات الاخرى للتشفير
  - ✓ كما انه يعرّف التشفير بانه "**فن**".
    - في الواقع ، لغاية القرن العشرين (او بصورة ادق لوقت متأخر من ذلك القرن) كان تصميم الشفرات وحلّها يعتمد على الابداع والخبرات الفردية ، حيث كانت هناك نظريات محدودة للتشفير ولايوجد حتى تعريف مناسب لمعنى كون الشفرة جيدة.
- في نهاية القرن العشرين تغيرّت هذه الصورة للتشفير بصورة جذرية ، حيث تطورت نظريات غنية للتشفير مما مهّد للدراسة الدقيقة للتشفير بشكل "**علم**".
- فضلا عن ذلك ، اصبح مفهوم التشفير يشمل مواضيع اكثر من الاتصال السري:
  - ✓ **كتوثيق الرسائل**
  - ✓ **التوقيع الرقمي**
  - ✓ **بروتوكولات تبادل المفاتيح**
  - ✓ **بروتوكولات كشف الهوية**
  - ✓ **التصويت والمزاد الالكتروني**
  - ✓ **والنقد الرقمي**.
- وبذلك فإن التشفير الحديث يشمل جميع المشاكل الناجمة عن الاحتساب التوزيعي distributed computation والتي تخضع لمهاجمات خارجية او داخلية.
- في ضوء ذلك ، يمكن تعريف التشفير بانه " **الدراسة العلمية لتقنيات تأمين المعلومات الرقمية ، المعاملات ، والاحتساب التوزيعي**".
- يتعلق الفرق الأخر الرئيسي بين التشفير التقليدي والتشفير الحديث بمن يستخدم التشفير.
  - كان اكثر مستخدمي التشفير هم الجهات العسكرية والوكالات الاستخبارية.
  - في الوقت الحالي ، يوجد التشفير في كل مكان ، حيث توجد الآليات الأمنية التي تعتمد على التشفير في كل نظام حاسوبي. مثلا:
    - يعتمد المستخدمون على التشفير في كل وقت للوصول الآمن لمواقع الويب
    - يطبّق التشفير للتحكّم عن بعد بأبواب السيارات
    - تستخدم الشركات طرق التشفير والتوثيق لحماية نسخ البرامجيات
    - يحمي التشفير عملية شراء السلع عبر الانترنت ببطاقات الاعتماد
- باختصار ، فإن التشفير انتقل من كونه فن للتعامل مع الرسائل العسكرية الى علم لتأمين الانظمة للمستخدمين في كل انحاء العالم. واصبح بذلك موضوع حيوي في مجال علوم الحاسوب.

1.2 التشفير بالمفتاح الخاص

- يهتم **التشفير بالمفتاح الخاص** private key cryptosystem بتصميم **مناهج تشفير** encryption schemes (تعرف احيانا ciphers) توفّر الاتصال السري بين طرفين يشتركان مسبقا بمفتاح التشفير.
- يعرف المفتاح الذي يتفق عليه الطرفان قبل بدء الاتصال **بالمفتاح الخاص** private-key.
- يستخدم الطرفان هذا المفتاح في كل مرة يرغبان بالتواصل السري.
- يقوم الطرف المرسل باستخدام المفتاح **لتشفير** الرسالة encrypt قبل ارسالها ،
- يستخدم الطرف المستلم نفس المفتاح **لفك شفرة** decrypt الرسالة المستلمة.
- تعرف الرسالة الاصلية **بالنص الصريح** plaintext ، اما الرسالة المرسلة فتعرف **بالنص المشفّر** ciphertext.
- تسمّى طريقة التشفير هذه ايضا **بالتشفير المتناظر** symmetric cryptosystem لان نفس المفتاح يستخدم في عمليتي التشفير و فك التشفير.
- تختلف هذه الطريقة عن طريقة **التشفير غير المتناظر** asymmetric cryptosystem ، التي تستخدم مفتاحين مختلفين ، احدهما يستخدم لعملية التشفير والاخر يستخدم لعملية فك الشفرة (سوف نذكره بالتفصيل لاحقا بأذن الله).
- تفترض طرق التشفير بالمفتاح الخاص انّ الاطراف المتواصلة تتفق مسبقا ، وبصورة سرية ، على المفتاح الخاص قبل البدء بالتواصل ،
  - مثلا عن طريق اللقاء المباشر

- هذا الافتراض يعتبر مصدر قلق للاستخدامات الحديثة ويحد من تطبيق مناهج التشفير ذات المفتاح الخاص.

1.2.1 صيغة التشفير بالمفتاح الخاص

- يتكون منهج التشفير بالمفتاح الخاص من ثلاث خوارزميات: تستخدم الأولى لتوليد المفاتيح ، تستخدم الثانية للتشفير ، و تستخدم الثالثة لفك الشفرة.

1- خوارزمية توليد المفتاح **Gen**: هي خوارزمية احتمالية probabilistic تنتج المفتاح k اعتمادا على توزيع المفاتيح المستخدم.

2- خوارزمية التشفير **Enc**: تأخذ كمدخلات المفتاح الخاص k والنص الصريح m وتخرج النص المشفّر c. نرمز لعملية التشفير بالشكل $Enc_k(m)$.

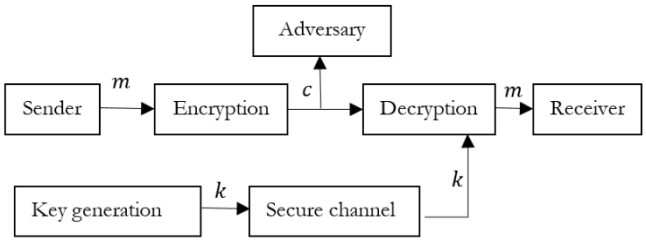
3- خوارزمية فك الشفرة **Dec**: تأخذ المفتاح السري k ، والنص المشفّر c وتعطي النص الصريح m. نرمز لعملية فك الشفرة بالشكل $Dec_k(c)$.

- تعتمد خوارزمية توليد المفاتيح على **فضاء المفاتيح** $\mathcal{K}$ key space (جميع القيم المحتملة للمفاتيح).
- تستخدم خوارزمية التشفير **فضاء النصوص الصريحة** (او الرسائل) $\mathcal{M}$ message space.
- يشمل **فضاء النصوص المشفرة** $\mathcal{C}$ cipher space جميع احتمالات النصوص المشفّرة.
- لاحظ ، ان اي منهج تشفير يتم صياغته بتعريف الخوارزميات الثلاث: (Gen, Enc, Dec).
- ينص **شرط الصحة** correctness لأي نظام تشفير على انّه لكل مفتاح k تنتجه Gen ، ولكل رسالة $m \in \mathcal{M}$ يجب ان يتحقق

$$Dec_k(Enc_k(m)) = m$$

بمعنى ان منهج التشفير يجب ان يلتزم بكون فك شفرة رسالة (بمفتاح معين) يؤدي الى الرسالة الاصلية التي تم تشفيرها.

- يستخدم التشفير بالمفتاح الخاص لاجراء الاتصال السري بين طرفين كما يلي: يتفق الطرفان على تنفيذ Gen للحصول على المفتاح المشترك k ، يقوم الطرف المرسل بتشفير الرسالة m عن طريق حساب $c = Enc_k(m)$ وارسال النتيجة الى الطرف المستلم الذي يقوم بدوره بفك الشفرة عن طريق حساب $m = Dec_k(c)$.



شكل (1.1): التشفير بالمفتاح الخاص

- يوضّح الشكل 1.1 طريقة عمل التشفير بالمفتاح الخاص.

1.2.2 المفاتيح ومبدأ Kerckhoffs

- يتضّح مما سبق ان سرية الرسالة المرسلة تعتمد على سرية المفتاح k فلو كان يعرف الخصم adversary المفتاح وطريقة عمل خوارزمية فك الشفرة Dec فأنه سيفك الشفرة بسهولة ويعرف الرسالة المرسلة. لذا يجب ان تتفق الاطراف المتواصلة على تبادل المفتاح بصورة سرية وتعمل على ابقاء تلك المفتاح سريّة تماما عن اي طرف اخر. قد يبالغ البعض بالتفكير باخفاء عمل خوارزميتي التشفير وفك الشفرة ايضا لزيادة الأمنية.
- يتناقض هذا التفكير مع مبدأ Kerckhoff ، الذي وُضع في القرن التاسع عشر ، والذي ينص على ضرورة الاحتفاظ بسرية المفتاح فقط دون منهج التشفير المستخدم. بمعنى ان جميع الاطراف ، بما فيهم الخصم ، يعرفون التفاصيل الداخلية لعمل منهج التشفير.
- وبموجب هذا المبدأ تعتمد أمنية منهج التشفير على سرية المفتاح فقط.
- هناك سببان رئيسان لاستخدام هذا المبدأ:
  - اولا ، ان الاحتفاظ بسرية مفتاح التشفير القصير تكون اسهل بكثير من الاحتفاظ بسرية خوارزميات التشفير الطويلة.
  - ثانيا ، ان عملية استبدال المفتاح المكتشف تكون اسهل من عملية استبدال خوارزمية التشفير.

1.2.3 سيناريوهات المهاجمة

نذكر فيما يلي سناريوهات المهاجمة attack scenarios الاساسية لمناهج التشفير مرتّبة حسب خطورتها ومقسّمة اعتمادا على المعلومات المتوفّرة لدى الخصم. سوف نفترض في جميع المهاجمات ان الخصم يعرف منهج التشفير المستخدم.نطلق على النص المشفّر الذي يرغب الخصم بكسر شفرته **بنصّ التحدي challenge text**:

1- المهاجمة بالنص المشفّر فقط **ciphertext-only attack**: تعتبر المهاجمة الاساسية ، وتمثّل السيناريو الذي **يتنصت eavesdropping** فيه الخصم على نص التحدي ويحاول معرفة نصّه الصريح.

2- المهاجمة بالنص الصريح المعروف **known-plaintext attack**: هنا ، يتوفر لدى الخصم عدد من النصوص الصريحة ونصوصها المشفّرة. تعرف هذه المعلومات بالازواج ويشترط ان تكون جميع النصوص الصريحة مشفّرة بنفس المفتاح. يهدف الخصم من دراسة هذه المعلومات الى معرفة المفتاح الخاص وبالتالي فك شفرة نص التحدي.

3- المهاجمة بالنص الصريح المختار **chosen plaintext attack (CPA)**: في هذه المهاجمة ، تتوفر لدى الخصم القدرة على تشفير اي نص صريح هو يختاره. بعد ذلك ، يحاول الخصم استثمار المعلومات المتوفّرة لفك شفرة نص التحدي.

4- المهاجمة بالنص المشفّر المختار **chosen ciphertext attack (CCA)**: هنا ، يتمكن الخصم من فك شفرة اي نص مشفّر من اختياره ، وبالتالي يحاول – ايضا- معرفة النص الصريح الذي يقابل نص التحدي.

- من المهم ملاحظة ان المهاجمات الأولى والثانية تُعد مهاجمات خاملة passive attacks ، من حيث كون الخصم يستلم فقط نصوص مشفّرة ويطلق مهاجمته لكسر شفرة تلك النصوص. في المقابل ، تعتبر المهاجمات الاخيرة مهاجمات فعّالة active attacks ، لأن الخصم يطلب بصورة متكررة تشفير او فك شفرة نصوص معينة هو يختارها.
- تعتبر المهاجمة الأولى هي الاسهل من حيث عمل الخصم ، كونها تتطلب فقط التنصت على النص المشفّر ، ولكنها ، في نفس الوقت ، تعتبر المهاجمة الاخطر وذلك لأن الخصم الذي يكسر الشفرة بالمعلومات القليلة المتوفرة (النص المشفّر فقط) لديه ، فإن ذلك يُعد مؤشر على ضعف أمنية منهج التشفير المستخدم.
- نؤكد على مسألة ، وهي انّه ليس من الضروري دائما استخدام منهج التشفير الذي يقاوم المهاجمة الاقوى ، وانما ذلك يعتمد على طبيعة التطبيق ، حيث تمتاز المناهج القوية بكفاءة اقل من المناهج التي تقاوم مهاجمات اضعف. قد تكون المناهج الضعيفة مفضلة وكافية لبعض التطبيقات.
- تعرف محاولة الخصم "**لكسر**" شفرة منهج التشفير **بكسر الشفرة** cryptanalysis.
- اوضح نوع من انواع كسر الشفرة هو محاولة **تخمين المفتاح**.
- تدعى المهاجمة التي يحاول بها الخصم جميع الاحتمالات الممكنة للمفتاح **ببحث المفتاح الشامل** exhaustive key search. عند تجربة المفتاح الصحيح سوف يظهر نص صريح مفهوم ، ولكن عند تجربة فك الشفرة بمفتاح خاطئ سوف يظهر نص عشوائي وغير مفهوم.

1.3 التشفير بالمفتاح المعلن

قُدّمت فكرة **التشفير بالمفتاح المعلن** public-key cryptosystem في سبعينيات القرن الماضي من قبل Diffie و Hellman. كانت فكرتهم هي بتصميم منهج تشفير ذو مفتاحين. يستخدم **المفتاح المعلن** public key لتشفير النص الصريح ويستخدم **المفتاح الخاص** private key لفك شفرة النص المشفّر.

- لاحظ ان المفتاح المعلن يكون معلنا "للجميع" بينما يكون المفتاح الخاص معلوما لصاحب ذلك المفتاح فقط. وبذلك فإن التشفير بالمفتاح المعلن يمكّن اي شخص من تشفير رسالة وارسالها الى المستلم. يستطيع فقط ذلك المستلم من فك شفرة تلك الرسالة.
- اشهر مثال على التشفير بالمفتاح المعلن هو منهج تشفير RSA الذي طُوّر من قبل Rives ، Shamir ، و Adleman.
- يعفي التشفير بالمفتاح المعلن الحاجة الى الاتفاق المسبق على المفتاح الخاص والمشترك بين الطرفين.

1.4 تطبيقات التشفير الاخرى

1.4.1 سلامة الرسالة

يعمل التشفير على توفير **السرية** secrecy (confidentiality) ضد الخصم المتنصت ، الذي يعرف عادة بالخصم الخامل passive adversary ، والذي يقتصر دوره على مشاهدة المعلومات المرسلة بين الطرفين.

- على كل حال ، توجد هناك تهديدات اخرى تتطلب الحماية منها ، خصوصا عندما يكون لدينا خصم فعّال active adversary. لنفترض ان الطرف المرسل يدعى Alice والطرف المستلم يدعى Bob. يعمل الخصم الفعّال على:
  - تغيير المعلومات المرسلة بين Alice و Bob.
  - ارسال معلومات الى Bob تجعله يظن انها مرسلة من Alice.

- يمنع وصول المعلومة من Alice الى Bob

- يجب ان نلاحظ ان التشفير ، بحد ذاته ، لايمنع هذه المهاجمات. مثلا ، باستطاعة الخصم قلب ثنائية (استبدال 0بـ1او بالعكس) من المعلومات المرسلة بين الطرفين بدون ان يؤثر ذلك على فك الشفرة.
- لذا ينبغي توفير آليات لحماية **سلامة الرسالة** message integrity المرسلة ضد مهاجمات الخصم الممكنة. يمكن للخصم ايضا ان يزوّر forge رسالة ويرسلها الى Bob ليقنعه انها مرسلة من قبل Alice.
- هناك أداتين من ادوات التشفير تحمي من مثل هكذا مهاجمات:
  - نستخدم في سياق التشفير بالمفتاح الخاص مايعرف **بشفرات توثيق الرسالة** message authentication code (MAC).
  - اما في سياق التشفير بالمفتاح المعلن فتستخدم اداة مشابهة تعرف **بالتوقيع الرقمي** digital signature.

شفرات توثيق الرسالة. تتطلب تقنية شفرة توثيق الرسالة من الطرفين Alice و Bob الاتفاق على مفتاح سري.

- عندما تريد Alice ان ترسل رسالة الى Bob ، تقوم باستخدام المفتاح السري لخلق شفرة MAC tag ترسل هذه الشفرة مع الرسالة.
- عندما يستلم Bob الرسالة والشفرة معا ، يستخدم المفتاح لاعادة حساب الشفرة ويفحص فيما اذا كانت تطابق الشفرة المستلمة.
- عندما يوجد تطابق فإن Bob "يقبل" الرسالة ، والا فانه "يرفض" الرسالة لكونها غير صالحة.

التوقيع الرقمي. في سياق التشفير بالمفتاح المعلن ، يوفّر التوقيع الرقمي تأكيدا مشابها للتأكيد الذي تقدّمه MAC.

- في منهج التوقيع ، يتم استخدام المفتاح الخاص لتوقيع الرسالة بخوارزمية تعرف بخوارزمية التوقيع signing algorithm.
- يتم الحاق التوقيع الناتج بالرسالة.
- في الطرف المقابل ، يستخدم المستلم خوارزمية فحص verification algorithm ، والتي تعتمد على المفتاح المعلن.
- تأخذ خوارزمية الفحص الرسالة والتوقيع كمدخلات وتقرر اما قبول او رفض الرسالة ، لتشير فيما اذا كان التوقيع صالحا ام لا.
- الشئ اللطيف في التوقيع الرقمي ان اي شخص يعرف المفتاح المعلن للمرسل يستطيع فحص التوقيع على العكس من شفرات MAC التي تقيّد فحص الشفرة بالطرف المستلم فقط.
- يمتاز التوقع الرقمي بكونه اقل كفاءة من MAC ، لذا فمن الافضل عدم استخدام التوقيع الرقمي لتوقيع رسائل "طويلة".
- في الجانب العملي ، يتم نحت hash الرسالة قبل توقيعها. تستخدم لهذا الغرض **دالة نحت** hash function ، والتي تعمل على ضغط الرسالة متغيرة الطول الى بصمة fingerprint ثابتة الطول وعشوائية الشكل. تمتاز دوال النحت بعدم استخدامها للمفتاح.
  - بعد ان تقوم Alice بنحت الرسالة ، تقوم بتوقيع البصمة الناتجة باستخدام مفتاحها الخاص. يتم ارسال الرسالة الاصلية والتوقيع الى المستلم Bob. تعرف هذه العملية **بأنحت ثم وقّع** hash-then-sign. يقوم Bob بنحت الرسالة المستلمة ثم يستخدم المفتاح المعلن لفحص صلاحية التوقيع على البصمة المحسوبة.
- من الجدير بالذكر ، ان دوال النحت المستخدمة في مجال التشفير تعرف بدوال النحت المقاومة للتصادم resistant collision hash functions ، والتي تعرف ايضا بدوال نحت التشفير cryptographic hash functions. يحدث التصادم للدالة h عدما $h(x) = h(y)$ بحيث $x \neq y$.

1.4.2 بروتوكولات التشفير

تستخدم ادوات التشفير كطرق التشفير ، شفرات MAC ، التوقيع الرقمي ، دوال النحت ، وغيرها في تصميم بروتوكولات معينة. ونقصد **بالبروتوكول** protocol سلسلة المعلومات المتراسلة بين طرفين (او اكثر) لتأسيس معلومة مشتركة محددة او تأكيد امتلاك معلومة مسبقة.

منهج كشف الهوية. احد اهم البروتوكولات هو منهج كشف الهوية identification scheme ، والذي يستخدمه احد الاطراف لكشف هويته او تأكيد امتلاكه لكلمة سر مثلا.

ادارة المفاتيح. تستخدم بروتوكولات التشفير في **ادارة المفاتيح** المستخدمة في التشفير. هناك اسلوبين لادارة المفاتيح:

- ✓ اسلوب **تبادل المفاتيح** key distribution ، حيث تستخدم مؤسسة موثوقة trusted authority لتوليد ونقل المفاتيح بين الاطراف المتراسلة في الشبكة ،
- ✓ واسلوب **الاتفاق على المفاتيح** key agreement ، حيث يشترك الطرفان Alice و Bob ، مثلا ، في عملية الاتفاق على المفتاح.

1.5 طرق التشفير التقليدية

- سوف نستعرض بعض طرق التشفير التقليدية ونبيّن عدم أمنة تلك الطرق ، مما يمهّد لصياغة مبادئ التشفير الحديث. اثناء استعراض تلك الطرق ، سوف نقدّم بعض مبادئ principles التشفير الاساسية والتي نتعلمها من نقاط ضعف الطرق التقليدية.
- سنكتب النصوص الصريحة بالحروف الصغيرة lower case، اما النصوص المشفّرة فتكتب بالحروف الكبيرة UPPER CASES. نفترض ان الخصم يعرف تفاصيل منهج التشفير (طبقا لمبدأ Kerckhoff). تكون النصوص المطلوب تشفيرها مكتوبة باللغة الانكليزية ، حيث نمثّل الحروف بشكل ارقام ضمن المجموعة $\{0, \dots, 25\}$.
- قبل البدء باستعراض الطرق التقليدية تجدر الاشارة الى ان هذه الطرق تقسم الى قسمين:

أ. طرق تعتمد على مبدأ **التعويض** substitution حيث يتم تعويض حروف النص الصريح بحروف اخرى.

ب. طرق تعتمد على مبدأ **البعثرة** permutation حيث تبقى نفس حروف النص الصريح ولكن تتغيير مواضعها في النص المشفّر.

1.5.1 شفرة قيصر

- تعتبر **شفرة قيصر Caesar cipher** اقدم طريقة تشفير ، حيث تعمل على تدوير حروف الابجدية الانكليزية ثلاث مواضع. حرف a يصبح D ، حرف b يصبح E وهكذا للبقية. تدوّر الحروف الموجودة في اخر الابجدية للبداية فحرف x يصبح A ، حرف y يصبح B.

مثال(1.1): سوف يشفّر النص "begin the attack now" بعد ازالة الفراغات بالشكل:

."EHJLQWKHDWWDFNQRZ"

- ✓ تمتاز هذه الطريقة بكونها ثابتة fixed ولاتستخدم اي مفتاح سري ، بحيث ان اي شخص يعرف طريقة عمل شفرة قيصر يستطيع فك شفرة النص المشفّر مما يجعلها غير آمنة.
- ✓ هناك طريقة اخرى تشبه شفرة قيصر ولها نفس مستوى الأمنية تعرف بـROT-13 حيث تعمل على تدوير الحروف بمقدار 13 موضع.

1.5.2 شفرة التزحيف

- تعاني شفرة قيصر من حقيقة كون التشفير يعمل دائما بنفس الطريقة ولاتتضمن مفتاح سري.
- تشابه **شفرة التزحيف shift cipher** شفرة قيصر ولكن يوجد فيها مفتاح تشفير k تتراوح قيمته بين 0 و 25 ، بحيث يتم التشفير بتدوير الحروف k من المواضع.
- تعتمد شفرة التزحيف على التعامل مع زمرة الاعداد الصحيحة $\mathbb{Z}_{26}$ والتي تتضمن مجموعة الاعداد $\{0, \dots, 25\}$ ، كما تستخدم عملية باقي القسم $\bmod 26$ لاختزال الاعدد الناتجة ضمن $0 - 25$.
- يوضّح المنهج (1.1) كيفية عمل شفرة التزحيف.

Construction (1.1): Shift cipher

Input: plaintext, $m \in \mathbb{Z}_{26}$.

Output: ciphertext, $c \in \mathbb{Z}_{26}$.

Key generation: choose random $k \leftarrow \mathbb{Z}_{26}$.

Encryption: $c = k + m \bmod 26$.

Decryption: $m = c - k \bmod 26$.

- لاحظ ان شفرة قيصر تعتبر حالة خاصة من شفرة التزحيف عندما يكون المفتاح $k = 3$.

كسر شفرة التزحيف. بما انه يوجد 26 احتمال لقيم المفتاح ، فإن الخصم يستطيع تجربة جميع تلك القيم لفك شفرة النص المشفّر وملاحظة اي من النتائج "يعطي معنى".

- يعرف هذا الاسلوب من المهاجمة بمهاجمة بحث المفتاح الشامل او مهاجمة **القوة القاسية** brute force attack.
- لكي يكون اي منهج تشفير آمنا فانه يجب ان يكون محصّن ضد محاولة كسر الشفرة هذه ، وألا فإن هذا المنهج سيكسر بصورة كاملة بغض النظر عن تعقيد خوارزمية التشفير المستعملة. وهذا يقودنا الى مبدأ مهم يدعى "**مبدأ كفاية فضاء المفتاح**":

اي منهج تشفير يجب ان يمتلك فضاء مفتاح محصّن ضد هجمة بحث المفتاح الشامل.

- في هذه الايام ، يستخدم الخصوم في مهاجمة brute force حواسيب قوية ، وربما يستخدمون عدد من الحواسيب. لذا ينبغي ان يكون حجم فضاء المفتاح كبيرا (على الاقل 2^{60}).
- ينبغي التأكيد على ان هذا المبدأ هو **شرط ضروري** necessary للأمنية ولكن ليس **شرط كافٍ** sufficient ، حيث سوف نرى ان بعض مناهج التشفير تكون غير آمنة رغم امتلاكها فضاء مفتاح كبير جدا.

مثال(1.2): لكسر شفرة النص المشفّر بطريقة التزحيف:

JBCRCLQRWCRWCRVNBJENBWRWN

JBCRCLQRWCRWCRVNBJENBWRWN
IABQBKPQVBQVBQUMAIDMAVQVM
HZAPAJOPUAPUAPTLZHCLZUPUL
GYZOZINOTZOTZOSKYGBKYTOTK
FXYNYHMNSYNSYNRJXFAJXSNSJ
EWXMXGLMRXMRXMQIWEZIWRMRI
DVWLWFKLQWLQWLPHVDYHVQLQH
CUVKVEJKPVKPVKOGUCXGUPKPG
BTUJUDIJOUJOUJNFTBWFTOJOF
ASTITCHINTINTIMESAVESNINE

فانه يتم تجربة جميع قيم المفتاح ... **0,1,** لحين الحصول على نص مفهوم

وهي عبارة "a stitch in time saves nine " عند المفتاح $k = 9$.

1.5.3 التعويض احادي الابجدية

- تعمل شفرة التزحيف على تحويل كل حرف الى حرف اخر ، لكن عملية التحويل تتم في كل مرة بنفس مقدار التزحيف.
- فكرة **التعويض احادي الابجدية** mono-alphabetic substitution هي بتحويل كل حرف صريح الى حرف أخر عشوائي ، مع ملاحظة خضوع التعويض لمبدأ واحد-لواحد ، لفرض تمكين عملية فك الشفرة.
- عمليا ، تتم عملية التشفير وفك التشفير باستخدام بعثرة الابجدية. يوضّح المثال التالي بعثرة عشوائية والتي يمكن استخدامها كطريقة تشفير. حيث يتم تشفير حرف a الى X ، وهكذا ، مما يشفر النص tellhimaboutme الى GDOOKVCXEFLGCD.
- لفك التشفير ، نستخدم معكوس البعثرة وذلك باخذ الحرف الموجود في الصف الثاني من الجدول واخراج الحرف المناظر له في الصف الاول ، فمثلا فك

a b c d e f g h I j k l mnopq r s t u v wx yz
X E U A D N B K V M R O C Q F S Y H W G L Z I J P T

شفرة الحرف المشفّر X هو a ، وهكذا.

- يتكون فضاء المفتاح ، عندما نتعامل مع الابجدية الانكليزية ، من جميع احتمالات بعثرة الابجدية والبالغ عددها $26!$ ، مما يجعل مهاجمة القوة القاسية متعذرة حتى في حالة استخدام اقوى الحواسيب المعروفة حاليا. في الواقع ، سوف نرى ان هذه الطريقة غير آمنة رغم كبر حجم فضاء المفتاح. يوضّح المنهج (1.2) طريقة عمل التشفير احادي الابجدية.

Construction (1.2): Substitution cipher

Input: plaintext, $m \in \mathbb{Z}_{26}$

Output: ciphertext, $c \in \mathbb{Z}_{26}$

Key generation: Generate random permutation, ρ.

Encryption: $c = \rho(m)$.

Decryption: $m = \rho^{-1}(c)$.

كسر شفرة طريقة التعويض احادي الابجدية. نفترض انه تم تشفير نص مكتوب باللغة الانكليزية. عندها ، من الممكن مهاجمة طريقة التشفير احادي الابجدية باستخدام الانماط الاحصائية statistical patterns لتلك اللغة.

- هناك صفتين في طريقة التشفير احادي الاجدية تمكّن اجراء هذه المهاجمه ، هما:

أ. عملية تحويل كل حرف تكون ثابتة ، وبذلك فإن حرف e عندما يحّول الى حرف D فإن جميع تكرارات الحرف e في النص الصريح سوف تنتج الحرف Dفي النص المشفّر.

ب. تكون التكرارات القياسية لحروف اللغة الانكليزية معروفة ، حيث تم حسابها وفق احصائات اجريت على مقالات لمجلات وصحف. يوضّح الجدول (1.1) احتماليات تكرار حروف اللغة الانكليزية.

جدول (1.1) احتمالات تكرار حروف اللغة الانكليزية

A	8.15%	N	7.10%
B	1.44%	O	8.00%
C	2.76%	P	1.98%
D	3.79%	Q	0.12%
E	13.11%	R	6.83%
F	2.92%	S	6.10%
G	1.99%	T	10.47%
H	5.26%	U	2.46%
I	6.35%	V	0.92%
J	0.13%	W	1.54%
K	0.42%	X	0.17%
L	3.39%	Y	1.98%
M	2.54%	Z	0.08%

- تجري المهاجمة الاحصائية بتشكيل احتماليات حروف النص المشفّر (بتقسم عدد تكرارات الحروف على طول النص المشفّر) ومقارنتها مع جدول الاحتمالات القياسية.
- عندها يتم تخمين حروف النص الصريح اعتمادا على هذه التكرارات.
- مثلا ، بما ان حرف e هو اكثر حروف اللغة الانكليزية تكرارا ، فاننا نخمّن ان الحرف الاكثر تكرار في النص المشفّر هو حرف e ، وهكذا.
- مالم يكن النص المشفّر طويلا ، فمن المحتمل ان تكون بعض التخمينات خاطئة. يمكن تحسين طريقة الحزر باستخدام بعض العلاقات بين الحروف ، مثلا حرف u يأتي عادة بعد q.

1.5.4 شفرة Vigenere

- كما رأينا ، تعمل المهاجمة الاحصائية على التعويض الاحادي الابجدية لكون تحويل كل حرف يكون ثابتا.
- بذلك ، فإن هذه المهاجمة يمكن ان تحبط بتحويل نفس الحرف الى عدة حروف مشفّرة مختلفة.
- يعمل هذا الاسلوب على "تنعيم احتماليات" حروف النص المشفّر.
- على سبيل المثال ، يحوّل الحرف e مرة الى الحرف G ، مرة الى الحرف M ، ومرة الى الحرف L. بهذا الاسلوب تم توزيع تكرارات الحرف e على ثلاث حروف مما يصعب فكّ شفرة تلك الحروف الثلاث كونها تستلم ايضا تكرارات من حروف النص الصريح الاخرى.
- يسمى هذا الاسلوب "التعويض متعدد الابجدية"poly-alphabetic substitution.
- تعمل **شفرة** Vigenere بتطبيق شفرة التزحيف عدد من المرّات ، حيث يتم اختيار كلمة قصيرة بمثابة المفتاح الخاص ، ويتم تكرار كلمة المفتاح عدد من المرات لحين تغطية طول النص الصريح. تجري عملية تشفير النص الصريح "بجمع" كل حرف صريح مع الحرف المقابل له في المفتاح ، مع اجراء عملية تدوير للنتيجة عند الحاجة.
- على سبيل المثال ، تشفير النص tellhimaboutme بالمفتاح cafe يعمل كما يلي:

```
Plaintext:  tellhimaboutme
Key:        cafecafecafeca
Ciphertext: WFRQKJSFEPAYPF
```

- وهو تماما تزحيف الحروف (الاول ، الخامس ، التاسع ، وبقية التسلسل) بمقدار 3، تزحيف الحروف (الثاني ، السادس ، العاشر ، وهكذا) بمقدار 1 ، وتزحيف الحروف (الثالث ، السابع ، وهكذا) بمقدار 5. بمعنى انه يتم تكرار شفرة التزحيف بمفاتيح مختلفة.
- لاحظ ، انه في المثال السابق تم تحويل الحرف l مرة الى الحرف R ومرة الى الحرف Q. كما ان الحرف المشفّر F يأتي احيانا من e واحيانا من a. وبذلك ، فانه تم تنعيم التكرارات في النص المشفّر ، كما هو مطلوب. يوضّح المنهج (1.3) شفرة Vigenere.

Construction (1.3): Vigenere cipher

Input: plaintext, $m = m_1, \dots, m_l : m_i \in \mathbb{Z}_{26}$

Output: ciphertext, $c = c_1, \dots, c_l : c_i \in \mathbb{Z}_{26}$

Key generation:

Generate $k = k_1, \dots, k_l : k_i \leftarrow \mathbb{Z}_{26}$.

Encryption: $c_i = m_i + k_i \bmod 26 : \forall i = 1, \dots, l.$

Decryption: $m_i = c_i - k_i \bmod 26 : \forall i = 1, \dots, l.$

كسر شفرة Vigenere. نفترض في البداية ان طول المفتاح $k = k_1, k_2, \dots, k_t$ المستخدم في التشفير يكون معلوم بطول t.

- يقسّم النص المشفّر الى t من الاجزاء ، بحيث كل جزء يفترض ان يكون مشفّرا بشفرة تزحيف خاصة.
- لو كان لدينا النص المشفّر $c_1, c_2, \dots$ ، فإن لكل مقطع $j \in \{1, \dots, t\}$ نعرف بأن مجموعة الحروف $c_j, c_{j+t}, c_{j+2t}, \dots$ كانت مشفّرة بالمفتاح k_j.
- المطلوب هو معرفة القيمة الصحيحة لذلك المفتاح الصحيح k_jبتجربة جميع القيم 26 المحتملة. وهذه ليست بالعملية البسيطة ، كما في كسر شفرة التزحيف ، حيث يتطلب تجربة جميع احتمالات المفتاح تجربة 26^tمن الاحتمالات للوصول الى نص صريح مفهوم.
- بقيت مسألة مهمة وهي كيفية تحديد طول المفتاح.
- احد الاساليب المستخدمة لمعرفة طول المفتاح هي باستخدام طريقة Kasiski. تعمل هذه الطريقة كما يلي: يتم فحص النص المشفّر لايجاد مقاطع متطابقة بطول ثلاث على الاقل. يتم تسجيل المسافات $\delta_1, \delta_2, \dots$ بين مواضع بداية تلك المقاطع. يحسب طول المفتاح وفق العلاقة: $t = \gcd(\delta_1, \delta_2, \dots)$ ، حيث تمثل عملية gcd القاسم المشترك الاعظم بين قيم المدخلات.

مثال (1.3): افترض ان لديك النص المشفّر بطريقة Vigenere:

نحاول الحل بطريقة Kasiski. يتكرر المقطع CHR في النص خمس مرات عند المواقع 1 ، 166 ، 236 ، 276 ، 268. تكون المسافات بين اول ظهور مع بقية التكرارات

CHREEVOAHMAERATBIAXXWTNXBEEOPHBSBQMQEQERBW
RVXUOAKXAOSXXWEAHBWGJMMQMNKGRFVGXWTRZXWIAK
LXFPSKAUTEMNDCMGTSXMXBTUIADNGMGPSRELXNJELX
VRVPRTULHDNQWTWDTYGBPHXTFALJHASVBFXNGLLCHR
ZBWELEKMSJIKNBHWRJGNMGJSGLXFEYPHAGNRBIEQJT
AMRVLCRREMNDGLXRRIMGNSNRWCHRQHAEYEVTAQEBBI
PEEWEVKAKOEWADREMXMTBHHCHRTKDNVRZCHRCLQOHP
WOAIIWXNRMGWOIIFKEE

للمقطع هي 165 ، 235 ، 257 ، 285 ، على التوالي. يكون القاسم المشترك الاعظم بين تلك المسافات هو 5 ، والذي يمثل طول المفتاح.

1.5.5 شفرة المفتاح التلقائي

- تشبه شفرة **المفتاح التلقائي** autokey cipher شفرة Vigenere ، باستثناء انه بدلا من تكرار كلمة المفتاح ، يتم توليد المفتاح اعتمادا على حروف النص الصريح المراد تشفيره. حيث يستخدم المفتاح لتشفير اول عدد قليل من الحروف ومن ثم يستخدم النص الصريح لتشفير البقية. تمتاز هذه الشفرة بأنه لايوجد طول للمفتاح
- وبالتالي لايمكن تطبيق طريقة kasiski و فهرس الصدفة لكسر شفرتها. يوضّح المنهج (1.4) طريقة عمل شفرة المفتاح التلقائي.

مثال (1.4): ليكن $k_1 = 8$ ، النص الصريح rendezvous ، يتم تحويل النص الصريح اولا الى ارقام صحيحة:

17 4 13 3 4 25 21 14 20 18

Construction (1.4): Autokey cipher

Input: plaintext, $m = m_1, \dots, m_t: m_i \in \mathbb{Z}_{26}$, private key $k \leftarrow \mathbb{Z}_{26}$

Output: ciphertext, $c = c_1, \dots, c_t: c_i \in \mathbb{Z}_{26}$

Key generation: $K_1 = k, K_i = m_{i-1}: \forall i = 2, \dots, t.$

Encryption: $c_i = m_i + K_i \bmod 26: \forall i = 1, \dots, t.$

Decryption: $m_i = c_i - K_i \bmod 26: \forall i = 1, \dots, t.$

يتولد المفتاح كما يلي:20 14 21 25 4 3 13 4 17 8

يجري التشفير بجمع الارقام المتقابلة مع تطبيق mod 26:

25 21 17 7 3 20 9 8 12

وهو يمثل النص المشفّر

ZVRQHDUJIM

1.5.6 شفرة Hill

- هي طريقة اخرى من طرق التشفير متعدد الابجدية.
- تم اختراعها من قبل Laster S. Hill عام 1929. فكرة هذه الطريقة هي تشفير كل t من حروف النص الصريح بشكل تحويل خطي لكي تنتج t من الحروف المشفّرة. جميع العمليات تكون $\bmod 26$.
- يكون المفتاح الخاص $K = (k_{i,j})$ بشكل مصفوفة مربعة بالابعاد $t \times t$.
- يجري تشفير النص الصريح $m = m_1, m_2, \dots, m_t$ باستخدام المفتاح الخاص K كما يلي:

$$(c_1, c_2, \dots, c_t) = (m_1, m_2, \dots, m_t)\begin{pmatrix} k_{1,1} & \cdots & k_{1,t} \\ \vdots & \ddots & \vdots \\ k_{t,1} & \cdots & k_{t,t} \end{pmatrix}$$

- حيث تطبّق عملية ضرب مصفوفتين ، وذلك بضرب المتجه m مع كل عمود من مصفوفة k ، وتجمع حواصل الضرب لكل عمود لنحصل على متّجه النص المشفّر. يكتب التشفير بالصيغة

$$c = Mk$$

- لغرض فك الشفرة يجب استخدام معكوس المصفوفة K^{-1} ، بحيث ان النص الصريح يحسب كما يلي:

$$m = cK^{-1}$$

- يكون K^{-1} معكوس للمصفوفة Kاذا تحققت العلاقة: $KK^{-1} = I$ ، حيث ان Iهي مصفوفة الوحدة ، والتي تكون مصفوفة مربعة بالابعاد $t \times t$ وجميع عناصرها صفرية باستثناء جميع عناصر القطر الرئيسي التي تأخذ القيمة واحد. المثال التالي يبين مصفوفة الوحدة بالابعاد 2×2

$$I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

مثال(1.5): ليكن لدينا المصفوفة $K = \begin{pmatrix} 11 & 8 \\ 3 & 7 \end{pmatrix}$ ، فإن معكوسها $K^{-1} = \begin{pmatrix} 7 & 18 \\ 23 & 11 \end{pmatrix}$ ، وذلك بسبب:

$$\begin{pmatrix} 11 & 8 \\ 3 & 7 \end{pmatrix}\begin{pmatrix} 7 & 18 \\ 23 & 11 \end{pmatrix} \bmod 26 = \begin{pmatrix} 11 \times 7 + 8 \times 23 & 11 \times 18 + 8 \times 11 \\ 3 \times 7 + 7 \times 23 & 3 \times 18 + 7 \times 11 \end{pmatrix}$$

$$= \begin{pmatrix} 261 & 286 \\ 182 & 131 \end{pmatrix} \bmod 26 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

- نؤكد انه ليس بالضرورة وجود معكوس لجميع المصفوفات ، فبعض المصفوفات لايوجد لها معكوس وبالتالي لايمكن استخدامها في عملية التشفير لعدم امكانية فك شفرة النص المشفّر. تتبيّن صحة عمل شفرة Hill بالشكل الاتي:

$$cK^{-1} = (mK)K^{-1} = m(KK^{-1}) = mI = m$$

نذكر مثالا بسيطا لكيفية تشفير نص بشفرة Hill.

مثال(1.6): ليكن لدينا النص الصريح july ، والمفتاح $K = \begin{pmatrix} 11 & 8 \\ 3 & 7 \end{pmatrix}$.

سيكون لدينا عنصرين للتشفير ، احدهما $(9,20)$ (الذي يمثل الحرفين ju)، والاخر $(11,24)$ (الذي يمثل الحرفين ly). يتم التشفير كما يلي:

$$(9,20)\begin{pmatrix} 11 & 8 \\ 3 & 7 \end{pmatrix} \bmod 26 = (99+60\,,72+140) = (3,4)$$

و

$$(11,24)\begin{pmatrix} 11 & 8 \\ 3 & 7 \end{pmatrix} \bmod 26 = (121+72\,,88+168) = (11,22)$$

وبذلك فإن تشفير july هو DELW.

لفك الشفرة نستخدم المعكوس $K^{-1} = \begin{pmatrix} 7 & 18 \\ 23 & 11 \end{pmatrix}$

$$(3,4)\begin{pmatrix} 7 & 18 \\ 23 & 11 \end{pmatrix} \bmod 26 = (9,20)$$

$$(11,22)\begin{pmatrix} 7 & 18 \\ 23 & 11 \end{pmatrix} \bmod 26 = (11,24)$$

وبذلك حصلنا على النص الصريح.

معكوس المصفوفة. خارج نطاق الموضوع- يجب مراجعة موضوع الجبر الخطي.

Construction (1.5): Hill cipher

Input: plaintext, $m = m_1, \dots, m_t : m_i \in \mathbb{Z}_{26} \wedge t \geq 2$

Output: ciphertext, $c = c_1, \dots, c_t : c_i \in \mathbb{Z}_{26} \wedge t \geq 2$

Key generation:

Generate $K = \{t \times t \text{ invertible matrix over } \mathbb{Z}_{26}\}$.

Encryption: $c = mK$.

Decryption: $m = cK^{-1}$.

نلخص شفرة Hill بشكل رياضي في المنهج (1.5).

مهاجمة شفرة Hill. يتم مهاجمة شفرة Hill باستخدام مهاجمة النص المصريح المعلوم known-plaintext attack.

- عندما يكون لدى الخصم t من الازواج (x_j, y_j)، بحيث ان x_j هو النص الصريح و y_j هو النص المشفّر.
- يتم اعادة كتابة المعطياة بالصيغة $X = KY$ بحيث ان Kهو المفتاح المطلوب ايجاده.
- بما ان المصفوفة X هي قابلة للعكس فإن الخصم يحسب المفتاح $K = X^{-1}Y$ وبالتالي يكسر النظام ككل.

يوضّح المثال التالي كيفية مهاجمة شفرة Hill.

مثال(1.9): افترض ان لديك نص مشفّر PQCFKU بشفرة Hill وان $t = 2$. ليكن لدينا ازواج من النصوص (المصريحة ، المشفّرة)

$$\{((5,17),(15,16)),((8,3),(2,5))\}$$

سوف نحصل على معادلة المصفوفة: $K\begin{pmatrix}15 & 16\\ 2 & 5\end{pmatrix} = \begin{pmatrix}5 & 17\\ 8 & 3\end{pmatrix}$

$$\begin{pmatrix}5 & 17\\ 8 & 3\end{pmatrix}^{-1} = \begin{pmatrix}9 & 1\\ 2 & 15\end{pmatrix}$$

وبذلك فإن $K = \begin{pmatrix}9 & 1\\ 2 & 15\end{pmatrix}\begin{pmatrix}15 & 16\\ 2 & 5\end{pmatrix} = \begin{pmatrix}7 & 19\\ 8 & 3\end{pmatrix}$

1.5.7 شفرة Playfair

- تعتبر طريقة تشفير Playfair واحدة من طرق التشفير التقليدية المشهورة.
- تعمل هذه الطريقة على تشفير النص الصريح بواقع حرفين في كل مرة.
- يتم في البداية تشكيل المفتاح بشكل مصفوفة بالابعاد 5×5 اعتمادا على كلمة سر معينة.
- تتم عملية تشكيل هذه المصفوفة بتخزين كلمة السر في المصفوفة بدون تكرار الحرورف ومن ثم نكمل المصفوفة بما تبقى من حروف الابجدية.
- يتم اعتبار الحرفين I و J كحرف واحد.
- مثلا لو كانت كلمة المفتاح هي CARS ، فإن مصفوفة المفتاح تكون بالشكل:

C	A	R	S	B
D	E	F	G	H
I/J	K	L	M	N
O	P	Q	T	U
V	X	X	Y	Z

- لتشفير النص الصريح نبدأ بتقسمة الى ازوج من الحروف وتشفّر هذه الازواج وفق القواعد التالية:
  - اذا كان حرفا الزوج متشابهان ، نضع بينهما حرف فاصل ، مثلا حرف X.
  - اذا كان الحرفين الصريحين يوجدان في نفس الصف من مصفوفة المفتاح ، فإن تشفيرهما يتم باختيار الحرف الموجود يمين كل حرف صريح منهما مع مراعاة التدوير نحو اليمين في حالة الحرف الاخير ، مثلا الحرفين PT يصبحان QL.
  - اذا كان الحرفين الصريحين يوجدان في نفس العمود من مصفوفة المفتاح ، فإن تشفيرهما يتم باختيار الحرف الموجود اسفل كل حرف صريح منهما مع مراعاة التدوير في حالة الحرف الاخير ، مثلا عند تشفير BH تكون النتيجة HN.
  - في حالة كون الحرفين بصفوف واعمدة مختلفة يكون تشفير كل حرف باختيار الحرف الموجود في صفّه وبعمود الحرف الاخر. مثلا عند تشفير الزوج AL فإن النتيجة هي RK
- وضّح كم هو حجم فضاء مفتاح هذه الطريقة ؟
- كيفية تشفير النص "wearesmartstudents"

1.5.8 شفرة البعثرة

- في **شفرة البعثرة permutation cipher** يتم استخدام مبدأ البعثرة لتغيير مواضع النص الصريح.
- ليكن لدينا المجموعة X ، فإن دالة البعثرة $\rho: X \to X$ هي دالة واحد-لواحد ، بحيث لكل $x \in X$ يوجد عنصر وحيد $x' \in X$، يحقق $\rho(x) = x'$.
يمكن تعريف معكوس البعثرة $\rho^{-1}: X \to X$ بالقانون:

$$\rho(x) = x' \text{ اذا وفقط اذا } \rho^{-1}(x') = x$$

مثال(1.10): افترض ان لديك بعثرة 6 عناصر معرفّة كما يلي:

x	1	2	3	4	5	6
ρ^{-1}	3	6	1	5	2	4

يمكن حساب معكوس البعثرة من الجدول اعلاه بالتبديل بين الصف الاول والثاني واعادة ترتيب العناصر بصورة مرتّبة حسب عناصر الصف الاول

x	1	2	3	4	5	6
$\rho(x)$	3	5	1	6	4	2

افترض انه لدينا النص الصريح:

shesellsseashellsbytheseashore

نقوم اولا بتقطيع النص الى مجاميع بطول ست حرورف:

shesel | lsseas | hellsb | ythese | ashore

الأن يتم اعادة ترتيب كل مجموعة اعتمادا على البعثرة ρ ، لتنتج التالي:

EESLSH | SALSES | LSHBLE | HSYEET | HRAEOS

وبذلك يكون النص المشفّر:

EESLSHSALSESLSHBLEHSYEETHRAEOS

لفك الشفرة نستخدم نفس الاسلوب ولكن باستخدام معكوس البعثرة ρ^{-1}.

1.6 شفرة Vernam

- تعرف ايضا بشفرة one-time pad ، وهي الشفرة الوحيدة التي توفّر الأمنية التامة عند تشفير عدد من الثنائيات ، خلافا للطرق التقليدية التي توفّر هذا المستوى من الأمنية عند تشفير عنصر واحد فقط.
- **الأمنية التامّة** perfect security: هنا يتم قياس أمنية منهج التشفير عندما يكون لدينا خصم بقدرة حسابية غير محدودة. يوفّر منهج التشفير أمنية تامّة (او أمنية بصورة غير مشروطة) اذا لم يوفّر النص المشفّر للخصم المعلومات الكافية للنجاح في المهاجمة ، بغضّ النظر عن قدرة الخصم الحسابية.
- ليكن $a \oplus b$ هو عملية XOR بين سلسلتين رمزيتين من الثنائيات ، بحيث لو كان $a = a_1, \dots, a_\ell$ و $b = b_1, \dots, b_\ell$ فإن $(a \oplus b) = a_1 \oplus b_1, \dots, a_\ell \oplus b_\ell$. يرمز للخيط الرمزي الذي يتكون من ℓ من الثنائيات بالرمز $\{0,1\}^\ell$. يوضّح المنهج (2.1) كيفية عمل شفرة one-time pad.

تعمل هذه الطريقة بصورة صحيحة وذلك لان:

$$\mathrm{Dec}_k(\mathrm{Enc}_k(m)) = k \oplus m \oplus k = m$$

Construction (2.1): One-time pad

Input: message, $m \in \{0,1\}^\ell$, of length ℓ

Output: ciphertext , $c \in \{0,1\}^\ell$

Key generation: Generate the key $k \leftarrow \{0,1\}^\ell$ randomly.

Encryption: $c = m \oplus k$.

لجميع قيم m ، k.

لايكشف النص المشفر c اي معلومة اطلاقا عن النص الصريح ، وبذلك لايستطيع اي خصم معرفة النص الصريح من النص المشفر.

- على الرغم من ذلك ، توجد بعض السلبيات لطريقة one-time pad.

اولا: تشترط هذه الطريقة ان يكون طول المفتاح بطول الرسالة المراد تشفيرها ، وهذا الشئ يكون غير عملي في بعض التطبيقات ، حيث يتحتم نقل هذا المفتاح ، الطويل ، بصورة آمنة الى الطرف الاخر كي يفك الشفرة. فلو كان حجم النص الطلوب تشفيره 1Gb ، فإن نفس كمية البيانات يشترط توليدها ونقلها بالنسبة للمفاتح. ولايقتصر الامر على هذا فحسب بل يجب توليد مفتاح جديد لتشفير كل رسالة ، وفقا لتعريف الأمنية التامة.

ثانيا: تقاوم هذه الطريقة مهاجمة النص المشفر فقط cipher-text only attack ولا تصمد بوجه مهاجمات اقوى مثل known plaintext attack ، حيث عندما يعرف الخصم نص مشفّر c ونصّه الصريح m فانه يعرف المفتاح ببساطة $k = m \oplus c$ ، لذا يتحتم استخدام مفتاح مختلف لتشفير كل نص كما ذكرنا في النقطة السابقة.

1.7 مناهج التشفير الضربية

- قدّم Shannon ابتكارا اخرا في مجال التشفير يدعى التشفير الضربي product cipher ، وهو فكرة الجمع بين منهجي تشفر عن طريق ضربهما. تمثّل هذه الفكرة اساس بناء الكثير من مناهج التشفير الحالية كمنهج DES ، حيث تكرّر فيها عملية التشفير 16 مرة على النص الصريح.
- لو كان لدينا منهجي تشفير S_1 و S_2 فإن منهج التشفير الضربي لهما هو $S_1 \times S_2$ ، بحيث تعرّف عملية التشفير بالشكل

$$Enc_{(k_1,k_2)}(m) = Enc_{k_2}(Enc_{k_1}(m))$$

حيث k_1 هو مفتاح التشفير للمنهج s_1 ، و k_2 هو مفتاح التشفير للمنهج s_2. بمعنى ان النص الصريح m يتم تشفيره اولا باستخدام $Enc_{k_1}()$ ومن ثم يتم "اعادة تشفير" النص المشفر الناتج بالعملية $Enc_{k_2}()$. تتم عملية فك الشفرة بالشكل

$$Dec_{(k_1,k_2)}(c) = Dec_{k_1}(Dec_{k_2}(c))$$

حيث نعكس فيها ترتيب عملية التشفير.

- من المهم الاشارة الى ان بعض (وليس جميع) مناهج التشفير الضربي تكون ابدالية ، بمعنى ان $S_1 \times S_2 = S_2 \times S_1$. من جانب اخر ، تكون هذه المناهج تشاركية اذا كان: $(S_1 \times S_2) \times S_3 = S_1 \times (S_2 \times S_3)$.
- عندما نضرب منهج تشفير Sمع نفسه ، نحصل على $S \times S = S^2$. عندما نكرّر عملية الضرب n من المرات نحصل على S^n.
- لاحظ ، ان الكثير من الطرق التقليدية للتشفير كشفرة التزحيف ، شفرة Hill ، شفرة التعويض ، شفرة البعثرة ، وشفرة Vigenere يكون فيها $S^2 = S$ ، بمعنى ان تكرار التشفير لايؤدي الى زيادة الأمنية. لذا ينبغي اختيار مناهج التشفير بعناية قبل ان نكّرر عليها عملية التشفير للحصول على زيادة في الأمنية.
- تعتبر تقنية ضرب منهج تشفير "تعويضي" بمنهج اخر معتمد على "البعثرة" من التقنيات الشائعة هذه الايام لتصميم منهج تشفير ضربي ذو أمنية عالية.