مقدمة

تقع على عاتق جميع أفراد المجتمع مسؤولية حماية وتأمين المعلومات (كالبرامج والبيانات). قد تتواجد المعلومات في الحواسيب المفردة أو في الشبكات أو في الانترنت. وبالتالي يتضمن أمن المعلومات: أمن الحواسيب، وأمن الشبكات، وأمن الانترنت.

بعض الجرائم الالكترونية:

يرتبط موضوع الجرائم الالكترونية مع بدء ظهور أجهزة الحاسوب، ويتطور مع تطورها، حيث تتعدد أساليب القرصنة وتتنوع لتواكب زمن حرب المعلومات التي وصلت إلى ذروتها مع مطلع القرن الحادي والعشرين . أما الدوافع وراء القرصنة الإلكترونية فكثيرة وغاياتها متعددة، بعضها يعود إلى الفضول الذي يلاحق القراصنة، إما لجمع المال وسرقة معلومات مهمة، وبعضها الآخر وهو الأهم، تقف وراءه دوافع سياسية غايتها اختراق الأنظمة الدفاعية، والتجسس على الشخصيات المهمة. وفي هذا السياق، نلقي الضوء على أشهر الجرائم التي تصدرت أخبارها الصحف العالمية، وكانت مصدراً لإثارة القلق، وأحيانا الشكوك حولها.

روبرت موريس يخترق الانترنت

بعد تخرجه في جامعة كورنيل في العام ،1988 صمم روبرت موريس أول دودة تجسس WORM، وكان دافعه الفضول في معرفة عدد الحواسيب المتصلة في الشبكة العنكبوتية، لكن حدث ما لم يكن في الحسبان، حيث فقد موريس السيطرة على الدودة التي بدأت تنسخ نفسها وتتكاثر على الشبكة ما أدى إلى حدوث أضرار جسيمة لعدد من أجهزة الحاسوب حول العالم، وقد تم إلقاء القبض عليه ليصبح موريس بذلك أول شخص يدان بموجب قانون الاحتيال الالكتروني وسوء استخدام الحاسوب، وحكم عليه بالسجن مدة ثلاث سنوات ودفع غرامة قيمتها 10 آلاف و50 دولاراً .

أدريان لامو يخترق (نيويورك تايمز)

في العام 2002 اخترق أدريان لامو البالغ من العمر 19 عاماً الشبكة الداخلية لصحيفة نيويورك تايمز ووصل إلى سجلات حساسة بما فيها قاعدة بيانات واسعة لبعض المقالات الافتتاحية والأوراق الأرشيفية

التي كانت تحتوي على أرقام هواتف وعناوين الأشخاص الذين كانوا يساهمون في الكتابة في الصحيفة، ومنهم على سبيل المثال جيمس بيكر وزير الخارجية الأمريكي السابق.

غاري ماكينون يخترق بيانات الجيش الأمريكي

بين عامي 2001 و2002 اتُهم الاسكتلندي غاري ماكينون باختراق أجهزة حاسوب خاصة بالجيش الأمريكي، وكان دافع "ماكينون" وراء ذلك هو جمع المعلومات التي بحوزة أمريكا عن الأجسام الغريبة الطائرة UFO !

وقال المسؤولون العسكريون أن من ضمن الأضرار الناجمة عن الاختراق حذف ملفات مهمة من أنظمة التشغيل، ما أدى إلى إغلاق شبكة المنطقة العسكرية التابعة للجيش في واشنطن والمكونة من ألفي جهاز حاسوب لمدة 24 ساعة.

وقام ماكينون أيضاً بحذف سجلات خاصة بالأسلحة في محطة "إيرل"، ويواجه حالياً حكماً بالسجن لمدة 60 عاماً إذا تمت إدانته بالتهم الموجهة إليه.

ألبرت غونزاليس يخترق شركة ذاتش.

أدين ألبرت غونزاليس رئيس عصابة مكونة من "هاكرز" بسرقة أكثر من 90 مليون بطاقة ائتمان وأرقام بطاقات السحب الآلي من TJX وغيرها من شركات تجارة التجزئة، بما فيها DSW وOfficeMax وسلسلة محال "ديف وبسترز" . وفي العام 2009 أدين غونزاليس بتهمة الاحتيال والسرقة وحكم عليه بالسجن لمدة 20 عاماً .

وفي نهاية سنة 2012 أكد مسؤول بالبيت الأبيض تسجيل محاولة قرصنة على نظام الحاسوب الخاص بالرئاسة الأمريكية، لكنها لم تبلغ إلا شبكة معلومات غير سرية ولا دليل على سرقة بيانات.

تتزايد المحاولات لتنفيذ الجريمة الالكترونية بنسب كبيرة لحسابات مستخدمين أو بطاقات ائتمانهم أو مهاجمة مواقع الكترونية حتى أصبحت تزيد عن الملايين في اليوم الواحد.

تعريف الأمان بشكل عام: هو طرق حماية نظام ما من جميع المخاطر التي قد يتعرض لها.

تعريف أمن المعلومات: هو طرق التحقق من كون الشخص الذي يقوم بتعديل أو قراءة البيانات المخزنة في النظام الرقمي هو شخص الذي يملك الصلاحية اللازمة لذلك، وأنه يقوم بذلك بالشكل الصحيح.

أهمية أمن الشبكات

تكمن أهمية أمن الشبكات في عدة نقاط منها:

1- حماية الخصوصية (Privacy) كالبيانات الشخصية... إلخ.

2- حماية المعلومات التي تملكها الشركة أو المؤسسة كالعقود والمناقصات وعلامات الطلاب... إلخ.

3- المحافظة على الوظيفة، فالإهمال في تحقيق الأمان ربما قد يتسبب بطرد الموظف المتسبب بالإهمال.

مفاهيم أمن الشبكات

هناك ثلاثة مفاهيم أساسية لأمن الشبكات هي:

1- السرية (Confidentiality): وتعني عملية الحفاظ على البيانات مخفية عن الأشخاص غير المرخص لهم. فمثلاً لتحقيق السرية نقوم بتشفير أو تغليف البيانات أثناء انتقالها.

2- السلامة (Integrity): وتعني ضمان عدم تعديل البيانات والخدمات من قبل جهة غير مرخص لها بذلك.

3- التوفر (Availability): أي ضمان أن تكون البيانات والخدمات متوفرة ومتاحة عند الحاجة لها.

الثغرة (Vulnerability): وهي نقاط الضعف الموجودة في النظام الرقمي والتي قد تستغل في التسبب بأذى له.

أمثلة عن نقاط الضعف:

1- الثغرات في البرمجيات ونظم التشغيل.

2- الاستخدام السيئ لبرمجية أو بروتوكول اتصال.

3- التصميم الضعيف للشبكة.

4- كلمة مرور غير آمنة.

التهديد (Threat): هو عبارة عن احتمال لخطر قد يستغل ثغرات النظام مما يؤدي إلى انتهاك مفاهيم أمن الشبكات.

مصادر التهديد: نوعان هما:

1- عرضي (Accidental): وهو غير متعمد ينتج بسبب أخطاء بشرية في تصميم النظام الرقمي أو استخدامه.

2- خبيث (Malicious) وهو متعمد: ومن يقوم بذلك هم:

-محبي الاختراقات.

-جهات منافسة.

-أعداء خارجيين.

-مجرمين إلكترونيين محترفين.

الهجوم (Attack): هو تنفيذ التهديد على النظام الرقمي بانتهاك الخدمات الأمنية والتهرب من الأليات الأمنية المطبقة عليه، ليتم الوصول لمعلومات ذلك النظام.

المخترق (Hacker): هو شخص خبير بعلوم الحاسوب ويستطيع الوصول بطريقة غير شرعية لبيانات خاصة بآخرين دون التسبب بآذى لهم.

المخترق الأخلاقي (Ethical Hacker): هو مخترق يقوم بتجريب اختراق الأنظمة الرقمية دون علم مالكها ودون تقاضي المال بهدف اكتشاف ثغرات النظام والنصح بتلافيها.

مختبر الاختراق (Penetration Tester): هو مخترق يقوم بتجريب اختراق الأنظمة الرقمية بعلم مالكها مع تقاضي المال منه بهدف اكتشاف ثغرات النظام والنصح بتلافيها.

المخرّب (Cracker): هو شخص خبير بعلوم الحاسوب ويستطيع الوصول بطريقة غير شرعية لبيانات خاصة بآخرين بقصد التسبب بآذى لهم.

الخدمات الأمنية (Security Services): هي مفاهيم في أمن الشبكات تعنى بالمحافظة على النظام الرقمي من أي هجوم.

من الخدمات الأمنية:

1- السرية. **2-** السلامة. **3-** التوفر[1].

[1] وهذه هي نفسها المفاهيم الأساسية الثلاثة في أمن الشبكات.

4- التوثيق[2] (Authentication): وتعني عملية التحقق من الهوية. وتعتبر خط الدفاع الأول ضد أي هجوم أو اختراق لنظام الشبكة.

5- الترخيص (Authorization): وتعني أن من يقوم بعملية ما له الحق والصلاحية للقيام بها.

6- عدم الإنكار (Non- Repudiation): وتعني تحمل مسؤولية القيام بعملية ما عند القيام بها.

7- التحكم بالوصول (Access Control): أي عملية تقييد الوصول للمصادر.

الآليات الأمنية (Security Mechanisms): هي الخطط والتقنيات المصممة لمنع أو كشف أي هجوم والشفاء منه. تقوم الآليات الأمنية بتحقيق الخدمات الأمنية.

من الآليات الأمنية:

1- التشفير (Encryption)، ويعتبر من آليات سرية البيانات.

2- التوقيع الرقمي (Digital Signature) هو برهان لهوية المرسل ويعتبر من آليات سلامة البيانات.

3- تبادليات التوثيق (Authentication Exchanges): وهي الآلية المستخدمة للتحقق من هوية الكائن[3] عبر تبادل المعلومات بينه وبين من يقوم بالتحقق منه. هناك العديد من أنظمة التوثيق، كل منها يحتاج متطلبات ولديه إمكانات مختلفة . وسنتحدث عنها بشيء من التفصيل في الفقرة التالية.

4- آليات التحكم بالوصول (Access Control Mechanisms): كأن نطبق تقنية قائمة التحكم بالوصول (Access Control List: ACL) والتي تستخدم للسماح أو عدم السماح للزبائن بالوصول لموارد الشبكة.

5- تغطية النقل (Traffic Padding): تغليف البيانات الحقيقية ببيانات أخرى لإخفاء البيانات الحقيقية. تؤمن هذه الآلية نقل سري للبيانات.

تبادليات التوثيق

وهي خط الدفاع الأول ضد أي هجوم أو اختراق لنظام الشبكة.

بعض أنظمة[4] المصادقة المستخدمة في أمان الشبكات:

[2] يسمى التوثيق أيضاً بالمصادقة.

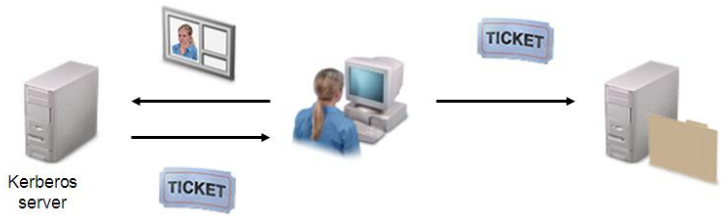
[3] نقصد بالكائن هنا: التجهيزات الشبكية - كالحواسيب و الموجهات والمخدمات- أو كل موقع ويب أو حسابات المستخدمين... إلخ.

[4] قد تكون هذه الأنظمة بروتوكولات أو تطبيقات (Applications)...إلخ

1- CHAP و MS-CHAP: ويستخدمان في عمليات التوثيق عندما يكون الاتصال من نوع نقطة لنقطة (Point-to-Point).

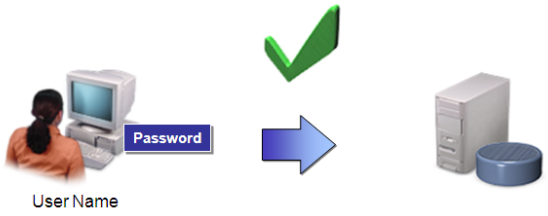
2- AAA: ويستخدم هذا النوع في عمليات التوثيق والترخيص والمراقبة. من التقنيات التي تطبق هذا النوع هي RADIUS أو TACACS+[5]

3- Kerberos: وفيها يتم منح تذكرة للمستخدم تسمح له باستخدام المورد المطلوب على الشبكة، بعد أن يتم التحقق منه. سيذهب المستخدم إلى المورد ويقدم له التذكرة (Ticket). يفحص المورد التذكرة ثم يسمح له بالدخول كما يوضح ذلك الشكل (1). تستخدم تقنية Kerberos في شبكات المجال (Domain).



الشكل (1)

تقوم عملية التوثيق بشكل عام على فكرة كلمة المرور (Password) كما يوضح ذلك الشكل (2).



الشكل (2)

كلمة المرور القوية (Strong Password): هي كلمة المرور التي تحقق المتطلبات المعقدة الموضوعة من مسؤول النظام (System Administrator) وموثقة في سياسة كلمة المرور (Password policy). تزيد كلمة المرور القوية من أمان النظام وتقلل من احتمالية تخمين كلمة المرور والكشف عنها.

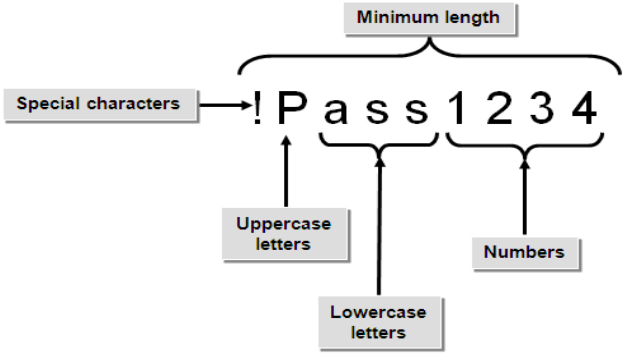
متطلبات كلمة المرور القوية:

- تحديد الحد الأدنى لطول كلمة المرور (Minimum Length).
- تتطلب تركيبة من الأحرف، الأرقام والرموز الخاصة (Special Characters) وحروف كبيرة (Upper Case) وصغيرة (Lower Case).

[5] TACACS+ خاص بتجهيزات Cisco.

- منع استخدام سلاسل الحروف مثل اسم المستخدم وكلمات القاموس.

انظر الشكل (3) لمشاهدة مثال عن كلمة مرور قوية.



الشكل (3)

عوامل التوثيق (Authentication Factors):

1- أشياء يجب أن تعرفها، مثل كلمة المرور (Password).

2- أشياء يجب أن تمتلكها، مثل بطاقة (Card).

3- أشياء لديك في ذاتك، مثل الشخصية الفيزيائية وبصمة الأصبع (Finger Print)..إلخ.

المصادقة متعددة العوامل (Multi-Factor Authentication): هي أي نمط يتطلب التحقق على الأقل من اثنين من عوامل المصادقة، ممكن أن يكون تركيبة من: (من أنت؟) و (ماذا لديك؟؟) و (ماذا تعرف ؟؟)

مثال: نظام يتطلب بطاقة هوية فيزيائية بالإضافة إلى كلمة مرور سرية كالحال مع بطاقة الصراف الآلي للبنوك.

السياسة الأمنية (Security Policy): هي وثيقة مفصّلة تشرح كيف ستقوم شركة ما بحماية مواردها وتجهيزاتها عبر مجموعة من الإرشادات والمعايير والاجرائيات التي يجب تطبيقها.

التشفير (Cryptography)

علم التشفير:

Cryptography: كلمة يونانية الأصل

kryptós تعني مخفي و gráphien تعني كتابة وبالتالي أصبح المصطلح هو الكتابة المخفية.

تعريف علم التشفير:

هو علم يهدف إلى حماية المعلومات بمزج محتوياتها وبذلك لن يستطيع أحد قراءتها إلا من لديه مفتاح تشفير يعيدها إلى أصلها.

مصطلحات لغة التشفير :

النص الواضح (Plain text): هو الرسالة الأصلية.

النص المشفر (Cipher text): هو الرسالة المشفرة.

خوارزمية التشفير (Cipher): هي خوارزمية لتحويل النص الواضح إلى نص مشفر.

المفتاح (Key): معلومة تستخدم في الشيفرة وتكون معروفة فقط للمرسل والمستقبل.

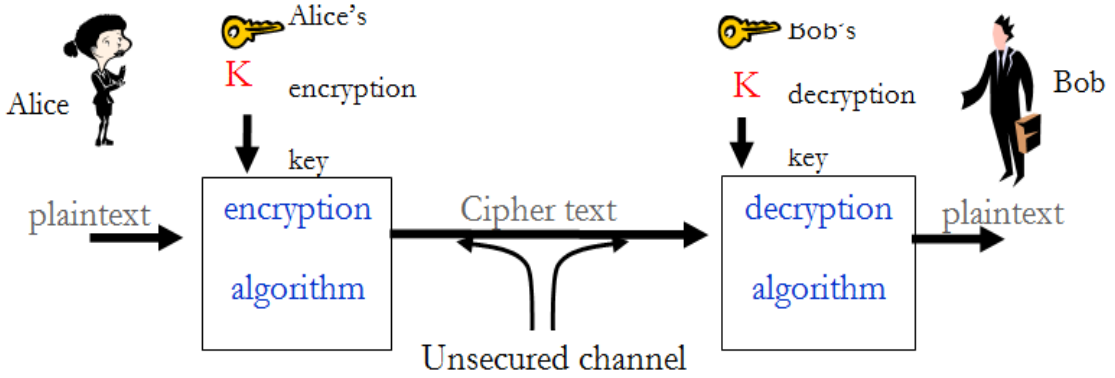
التشفير Encipher (Encrypt): تحويل النص الواضح إلى نص مشفر.

علم التشفير(Cryptography): دراسة مبادئ وطرائق التشفير.

فك التشفير(Decipher (Decrypt: إعادة النص المشفر إلى نص واضح بواسطة مفتاح.

كسر التشفير (Code breaking): ويسمى أيضاً تحليل التشفير (Cryptanalysis) وهو دراسة مبادئ وطرائق فك تشفير النص المشفر دون معرفة المفتاح.

يوضح الشكل (1) عملية تشفير البيانات عند المرسل ثم انتقالها عبر قناة غير آمنة ثم فك تشفيرها عند المستقبل.



الشكل (1)

أي عندما أرغب بنقل رسالة عبر وسط غير آمن إلى شخص آخر: أحول الرسالة إلى صيغة مشفرة (Cipher text) وأرسلها عبر الوسط إلى النظام في الطرف الآخر وسيكون هو الوحيد القادر على كشف محتوى هذه الرسالة.

ملاحظة (1): ليس بالضرورة أن يكون نقل المعلومات المشفرة عبر الانترنت فقط، فقد نحتاجه عبر شبكة خاصة.

تمثيل نظام التشفير(Cryptosystem) :

يمثل نظام التشفير عادة بخماسية مؤلفة من (M , C , K , E , D) حيث:

M: هي مجموعة منتهية من كل النصوص الواضحة (Plain Text) أي كل الرسائل أو المعطيات الأصلية وفق أبجدية معينة والتي تكون مدخلات للخوارزمية.

C: هي مجموعة كل الشيفرات الممكنة (Ciphertext) التي تخرجها الخوارزمية وذلك وفق أبجدية معينة.

K: فضاء المفاتيح أي مجموعة منتهية من كل المفاتيح الممكنة (Key).

E: خوارزمية التشفير.

D: خوارزمية فك التشفير.

وكمعنى رياضي :

من أجل كل k في K، يوجد قاعدة تشفير ek في E تقابلها قاعدة فك تشفير dk في D.

بحيث:

$ek: M \rightarrow C$

$dk: C \rightarrow M$

هي عبارة عن توابع تحقق:$dk(ek(m)) = m$ من أجل كل النصوص الواضحة m في M.

كسر الشيفرة (Code Breaking): أي اكتشاف الرسالة الأصلية من رسالة مشفرة دون معرفة المفتاح وذلك من قبل المختصين لاختبار قوة الخوارزمية أو من قبل المهاجمين.

الكندي هو أول عالم استطاع أن يكتشف طريقة لكسر شفرة معينة .

تصنيف الشيفرات[6]:

يمكن تصنيف الشيفرات بأكثر من معيار:

التصنيف حسب مفتاح التشفير

1- خوارزميات التشفير التناظري (Symmetric-key Algorithm):

المرسل والمستقبل لديهما نفس المفتاح وهما الوحيدان المالكان له أي يتم التشفير وفك التشفير باستخدام نفس المفتاح.

$$ek(m) = c$$

$$dk(c) = m$$

$$dk\ ek(m) = m$$

2- خوارزميات التشفير غير التناظري (Asymmetric-key Algorithm):

كل طرف لديه مفتاح مختلف عن الآخر.

بفرض أن مفتاح التشفير هو k1 وأن مفتاح فك التشفير هو k2:

$$ek1(m) = c$$

$$dk2(c) = m$$

$$dk2\ (ek1(m)) = m$$

التصنيف حسب فترة ابتكار الخوارزمية

[6] سيمر معنا في هذا المقرر تصنيفات أخرى للشيفرات.

1- التشفير بالطرق الكلاسيكية (Encryption in Classical Method)

الطرق الكلاسيكية هي الطرق القديمة التي استخدمت فيما مضى من قبل اختراع الحواسيب ، وبقيت الأساس لكثير من الخوارزميات الحديثة التي تستخدم اليوم ، حيث كانت تعتمد على أحلال أو أبدال حرف مكان أخر ، والخوارزميات الجيدة كانت تقوم بالاثنين ، لكن جميع هذه الخوارزميات تعمل على الحروف فقط **Character-Based** أي على مدى 26 حرف ، بعكس الطرق الحديثة التي تعتمد على التعامل مع الBit (0 أو 1) ، وتقسم الطرق الكلاسيكية إلى قسمين رئيسين :

شفرات الإحلال Substitution Cipher :
في هذا النوع من الشفرات ، التشفير يكون عن طريق إحلال حرف من النص الأصلي Plaintext بحرف أخر ليكون هو الحرف المشفر cipher char ، عمليه الإحلال هذه تكون طريق جمع مفتاح ما إلى الحرف من النص الأصلي .

شفرات الإبدال Transposition :
في هذا النوع التشفير يكون عن تغيير أماكن حروف النص الأصلي ، أي مجرد تبديل في المواقع . (بعض الكتب تطلق على هذا النوع اسم **Permutation** تقليب) .

تقسم شفرات الإحلال Substitution Cipher إلى أربعه أقسام رئيسيه :
النوع الأول : Monoalphabetic Substitution Cipher
النوع الثاني : Polyalphabetic Substitution Cipher
النوع الثالث : PolyGram Substitution Cipher
النوع الرابع : Homophonic Substitution Cipher

وسوف نتطرق لكل من هذه الطرق بالتفصيل ، أحب أن أنوه إلى أن هناك بعض الترجمات السيئة لهذه الأنواع ، لكني سأتحفظ عنها هنا ، وسنذكر المصطلح كما هو باللغة الأنجليزيه .

شفرات *Monoalphabetic Substitution Cipher* :

هذا النوع يعتبر من أقدم أنواع التشفير استخداما ، حيث نقوم في هذا النوع بإحلال Substitution حرف من النص الأصلي بحرف أخر جديد . وهو بالاضافه إلى قدمه يعتبر من أضعف أنواع التشفير ويسهل كسره باستخدام طريقه تسمى التحليل الإحصائي frequency analysis ، وهذه الطريقة من اكتشاف العالم العربي المسلم **أبو يعقوب الكندي** وهو أول من وضع أساسيات كسر الشفرات Cryptanalysis، حيث لاحظ وجود حروف تتكرر في القران الكريم أكثر من غيرها.

من أشهر شفرات هذا النوع Monoalphabetic Substitution :
Caesar Cipher
Affine Cipher
ROT13 Cipher
Abash Cipher

1- خوارزمية قيصر (Caesar Algorithm)

أول شيفرة إحلال تقليدية هي قيصر والتي تعتمد على الإزاحة بمقدار ثلاث حروف، ثم طورت لتشمل أي إزاحة معينة أخرى.

تتم وفق المراحل التالية:

1- إعطاء كل حرف رقم كما في الشكل (2):

A	B	C	D	E	F	G	H	I	J	K	L	M
0	1	2	3	4	5	6	7	8	9	10	11	12
N	**O**	**P**	**Q**	**R**	**S**	**T**	**U**	**V**	**W**	**X**	**Y**	**Z**
13	14	15	16	17	18	19	20	21	22	23	24	25

الشكل (2)

2- ثم تحسب خوارزمية قيصر كما يلي:

للتشفير $C = e(m) = (mi + k) \bmod (26)$: mi is the letter i of m

لفك التشفير $M = d(c) = (ci - k+26) \bmod (26)$: ci is the letter i of c

k:1…...25

مثال (1): لدينا النص الصريح التالي:

M= “meet me after the toga party”

الحل: نقوم بتشفيره حسب قيصر باستخدام الإزاحة بمقدار 3 إلى النص:

C= “phhwphdiwhuwkhwrjdsduwb”

كسر تشفير قيصر:

لدينا طريقتين :

1- التكرار الموجود ضمن اللغة البشرية لأن هذه الرسائل تستخدم أبجدية مقروءة ولأن أي لغة تحوي إحصائيات معروفة لتكرار أحرفها. أكثر الحروف تكراراً في اللغة الانجليزية هو الحرف E.

2- التجريب: عندما يكون المفتاح قصير أي يتكون من حرف واحد كما الحال في قيصر وفضاء المفاتيح صغير حيث لدينا 25 مفتاح فقط يمكن تجربتهم كلهم فعند توفر نص مشفر (Ciphertext) يمكن تجربة جميع الإزاحات للأحرف (مثلاً A يتم إسقاطها إلى الأحرف من B إلى Z) ونحتاج فقط إلى تميز النص عندما نصل إلى النص الواضح (Plaintext) .

مثال (2): اكسر الشيفرة التالية حسب خوارزمية قيصر معتمداً طريقة التجريب.

c = GCUA VQ DTGCM

الحل: Key 1 fbtz up csfbl

Key 2 easy to break

وهنا ظهر النص الصريح.

مثال (3): اكسر الشيفرة التالية حسب خوارزمية قيصر معتمداً طريقة التكرار.

fqjcb rwjwj vnjax bnkhj whxcq nawjv nfxdu mbvnu ujbbf nnc

الحل :

أول خطوة هي معرفة كم مرة تكرر كل حرف . نستطيع أن نعرف ما هو الحرف الذي تكرر أكثر من غيره، ومنه قد يكون هو الحرف E .

الآن نبدأ بعد جميع حروف الأبجدية .....

A:2 B:5 C:3 D:1 E:0 F:3 ……..j:7 … m:1 N: 7…... R:1 …. W:4 x:3 Y:0 Z:0

نعود إلى الحرف الأكثر تكراراً في النص المشفر ونجد أن لدينا حرفان هما j و n حيث تكرر كل منهم 7 مرات.

إن الفرق بين الحرف j والحرف E هو 5 ، وبالتالي المفتاح هو 5 ، ونبدأ بفك الشيفرة:

نطرح من كل حرف 5 حروف :

f - 5 = a
q - 5 = l
j - 5 = e
c - 5 = x
b - 5 = w

والناتج يكون alexw mrere ajevs نتوقف هنا بالطبع ، لأن النص ليس له معنى بتاتاً.

نأخذ الحرف الثاني تكراراً ، وهو الحرف N .

الفرق بينه وبين E هو 9 ، إذاً المفتاح في هذه الحالة يكون 9.

الآن نطرح من كل حرف 9 حروف .....

f-9 = w
q-9 = h
j-9 = a
c-9 = t
b-9 = s

وهكذا ليخرج لدينا النص الصريح التالي:

whats inana mearo sebya nyoth ernam ewoul dsmel lassw eet

وبعد ترتيب الكلمات نبدأ بمعرفة الجمل فيكون لدينا :

what's in a name a rose by any other name would smell as sweet

مثال (4) اكسر تشفير خوارزمية قيصر عن طريق التحليل الإحصائي:

لدينا الشيفرة التالية ونريد كسرها وإرجاعها إلى حالتها الأصلية .

```
WFIDZ JVORT KCPVD GKZEV JJVDG KZEVJ JVORT KCPWF IDJFZ KZJNZ KYJVE
JRKZF EGVIT VGKZF EDVEK RCIVR TKZFE REUTF EJTZF LJEVJ JRCCK YZEXJ
RIVVJ JVEKZ RCCPV DGKPE FKSFI EEFKU VJKIF PVUEF KJKRZ EVUEF KGLIV
NZKYF LKCFJ JNZKY FLKXR ZEKYV IVWFI VZEVD GKZEV JJKYV IVZJE FWFID
EFJVE JRKZF EGVIT VGKZF EDVEK RCIVR TKZFE FITFE JTZFL JEVJJ EFVPV
VRIEF JVKFE XLVSF UPDZE UEFTF CFIJF LEUJD VCCKR JKVKF LTYFS AVTKF
WKYFL XYKEF JVVZE XREUJ FFEKF EFKYZ EBZEX EFZXE FIRET VREUE FVEUK
FZXEF IRETV EFFCU RXVRE UUVRK YEFVE UKFFC URXVR EUUVR KYEFR EXLZJ
YTRLJ VFWRE XLZJY TVJJR KZFEG RKYEF NZJUF DREUE FRKKR ZEDVE KJZET
VKYVI VZJEF KYZEX KFRKK RZEKY VSFUY ZJRKK MRCZM VJKYL JNZKY EFYZE
UIRET VFWDZ EUEFY ZEUIR ETVRE UYVET VEFWV RIWRI SVPFE UUVCL UVUKY
FLXYK IZXYK YVIVZ JEZIM RER
```

الخطوة الأولى هي معرفة الحروف الأكثر تكراراً، ونظراً لطول الشيفرة فيفضل عد كل حرف يتكرر، نبدأ بالعد ، نقوم بعد الحرف الأول وهو W ونلاحظ كم مرة تكرر واحد ، اثنين ، ثلاثة .....إلى أن نصل إلى نهاية الشيفرة لنعرف أن الحرف W تكرر 9 مرات ، نأخذ الحرف الثاني وهو F ونبدأ بالعد ،وهكذا مع باقي الحروف في الشيفرة.

نتيجة تكرار الحروف بعد العد:

A: 1 B: 1 C: 16 D: 14 E: 82 F: 69 G: 10 H: 0 I: 27 J: 47 K: 61
L: 15 M: 3 N: 5 O: 2 P: 8 Q: 0 R: 45 S: 5 T: 21 U: 28 V: 69
W: 9 X: 15 Y: 28 Z: 47

نلاحظ في النتيجة أعلاه، أن الحرف E هو الحرف الأكثر تكراراً في النص المشفر(تكرر 82 مرة)، الآن كما ذكرنا سابقاً الحرف الأكثر تكراراً في الشيفرة قد يكون هو الحرف E ،و لأن في حالتنا هذه ، الحرف المشفر هو E إذا بالتأكيد الحرف E لن يكون هو الحرف البديل ، لذلك سوف نأخذ الحرف الأكثر تكراراً في الشيفرة أتى بعد الحرف E .

الآن لدينا حرفين هما V،F حيث تكرر كل منهم 69 مرة ، وقد يكون أحدهم هو الحرف E .

الآن نأخذ الحرف F ونشاهد الفرق بينه وبين الحرف E والنتيجة هي1.

الآن نأخذ الحرف V ونشاهد الفرق بينه وبين الحرف E والنتيجة هي17.

إلى هنا ، أصبح لدينا احتمالين ، الأول هو أن يكون المفتاح هو 1 ، والثاني هو أن يكون المفتاح هو 17.

نقوم بتجربة الأول ، ونطرح من كل حرف في الشيفرة 1، النتيجة ستصبح غير مفهومة وبالتالي الإزاحة بمقدار 1 خاطئة.

نجرب الإزاحة بمقدار 17، وسوف نحصل على هذا النص:

```
FORMI SEXAC TLYEM PTINE SSEMP TINES SEXAC TLYFO RMSOI TISWI THSEN
SATIO NPERC EPTIO NMENT ALREA CTION ANDCO NSCIO USNES SALLT HINGS
AREES SENTI ALLYE MPTYN OTBOR NNOTD ESTRO YEDNO TSTAI NEDNO TPURE
WITHO UTLOS SWITH OUTGA INTHE REFOR EINEM PTINE SSTHE REISN OFORM
NOSEN SATIO NPERC EPTIO NMENT ALREA CTION ORCON SCIOU SNESS NOEYE
EARNO SETON GUEBO DYMIN DNOCO LORSO UNDSM ELLTA STETO UCHOB JECTO
FTHOU GHTNO SEEIN GANDS OONTO NOTHI NKING NOIGN ORANC EANDN OENDT
OIGNO RANCE NOOLD AGEAN DDEAT HNOEN DTOOL DAGEA NDDEA THNOA NGUIS
HCAUS EOFAN GUISH CESSA TIONP ATHNO WISDO MANDN OATTA INMEN TSINC
ETHER EISNO THING TOATT AINTH EBODH ISATT VALIV ESTHU SWITH NOHIN
DRANC EOFMI NDNOH INDRA NCEAN DHENC ENOFE ARFAR BEYON DDELU DEDTH
OUGHT RIGHT HEREI SNIRV ANA
```

هناك احتمال كبير أن يصعب عليك ترتيب الحروف السابقة وإرجاعها إلى حالتها الأصلية بسبب عدم إتقان اللغة الانكليزية بشكل جيد . لكنها تبقى في النهاية هي النص الأصلي.

ملاحظة (1): يمكن اختيار جملة للتشفير (Key Phrase) بدلاً من المفتاح (الإزاحة)

مثال (1): لدينا جملة التشفير THE HILLS ARE ALIVE

اذا أردت أن أشفر الحرف A بهذه الطريقه ، سوف يكون الحرف A بعد التشفير هو الحرف T لأنها الأولى في جمله التشفير .
و B يصبح H ، و C يصبح E ، وهكذا...

وفي حال انتهت جملة التشفير ولم ينتهي النص الذي نريد تشفيره ، فنقوم بتكملة الحروف بالحروف الأبجدية المتبقية.

مثال (2): لدينا جملة التشفير BASIL

والمطلوب تشفير العبارة STOP FIRE

الحل: نقوم أولا بوضع الحروف الأبجدية المرتبة ومن أسفلها النص المشفر بالإضافة إلى باقي الحروف

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

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

الآن نقوم بتشفير النص المطلوب STOP FIRE لينتج لدينا النص المشفر التالي:

RTNO CFQL

شفره أتباش *Atbash Cipher*

هذه الشفرة أيضا من أبسط أنواع الشفرات ، و هي كانت في الأصل للغة العبرية ، ولكن يمكن استخدام المفهوم في باقي اللغات .
وطريقتها كالتالي ، وهي أن نجعل الحرف الأول في اللغة هو الحرف الأخير ، والحرف الثاني هو قبل الأخير ، وهكذا...

Plain: ABCDEFGHIJKLMNOPQRSTUVWXYZ
Cipher: ZYXWVUTSRQPONMLKJIHGFEDCBA
مثلا ، لتشفير الكلمة money ، يصبح لدينا الناتج nlmvb .

شفرات *Polyalphabetic substitution cipher*

الشفرات التي تندرج تحت هذا النوع ، هي مجرد تطبيق لطريقة Monoalphabetic عليها عده مرات ، أي أن المفتاح هنا يكون عبارة عن عده مفاتيح . مثلا اذا كان عدد المفاتيح4 ، يشفر الحرف الأول بالمفتاح الأول والحرف الثاني بالمفتاح الثاني ، وهكذا . وعندما تنتهي المفاتيح بعض الطرق تقوم بإعادة كتابتها مره أخرى ، وبعضها لا تقوم.

وأشهر الشفرات تحت هذا النوع ، هي شفرات عائلة فجينير Vigenere Cipher وقد طورت هذه الشفرات على مدى السنين من قبل أناس مختلفين.

من أشهر هذه الشيفرات:

1- شيفرة Simple Shift Vigenere Cipher

2- خوارزمية فيجينير الكاملة (Full Vigenere Cipher)

3- خوارزمية فيجينير تلقائية المفتاح (Auto-key Vigenere Cipher)

4- خوارزمية فيجينير طويلة المفتاح (Running-key Vigenere Cipher)

شفرات PolyGram Substitution Cipher

لاحظنا في الطرق السابقة أن أخذ حرف واحد وتشفيره بمفتاح إلى حرف مشفر ، هي طرق ضعيفة ويمكن كسرها بسهوله ، لكن هنا في الـ **POLYGRAM** التشفير سوف يكون بطول بلوك Block ، أي نأخذ البلوك الأول كاملا ونشفره ، ونضع البلوك المشفر . طبعا لا يشترط أن يكون بلوك النص الأصلي هو نفس حجم بلوك النص المشفر.

مثلا لدى خوارزمية تأخذ بلوك من 8 أحرف ، وتضع بدله بلوك مشفر من 8 أحرف ، كما هو موضح بالصورة التالية:

```
AAAAAAAA        maps to        ZXCIJCDV
AAAAAAAB        maps to        APQODFIM
...             ...            ...
ZZZZZZZZ        maps to        SSTFQQWR
```

اذا أردنا أن نطبق طريقة الهجوم العنيف[7] (Brute-Force Attack) لكسر كتلة واحدة فسوف نحتاج إلى $26^{\wedge}8$[8] احتمال، وهو ما يعادل 208,827,064,576 تخمين.

من أشهر شيفرات هذا النوع:

شفره بلافير Playfair
شفره هيل Hill Cipher
أسطوانة جيفيرسون Jifferson Cylinder

التشفير بطريقه الـ HOMOPHONIC SUBSTITUTION CIPHERS

يعتبر من الطرق الجيدة لإحباط التحليل الإحصائي.

التشفير بالإبدال

وطريقة الإبدال لها العديد من الأشكال والطرق والشفرات ومنها البسيط ومنها المعقد ، وسنأخذ أشهر وأسهل واحدة:

[7] الهجوم العنيف هو أحد أنواع الهجمات الشهيرة لتخمين النص المشفر عبر تجريب كل الاحتمالات الممكنة لتشكيله.

الطريقة الأولى طريقة العكس

الطريقة بسيطة للغاية، وكل ما في الأمر أننا سنبدل الحرف الأول مكان الحرف الأخير ، الحرف الثاني بالحرف ما قبل الأخير، وهكذا... ، وهي من أضعف أنواع الشفرات.

مثال:

النص الأصلي : Wajdy Essam Is Java Developer

بعد تشفيره : repoleveD vavJ sI massE ydjaW

لماذا لا نخترع خوارزمية ونبقيها سرية عن الجميع وبهذا لا يعرفها المخترق؟

المخترقون سوف يكتشفون الخوارزمية مهما فعلت ، ولا أي واحد في تاريخ التشفير تمكن من إبقاء خوارزميته سرية . لطالما تمكن الجواسيس في الحرب من كشف الخوارزميات سواء باستخدام عمليات رياضية أو أجهزه لكسر الشفرات، أو حتى يوظفوا جواسيس لدى العدو، أو يسرقوا الخوارزمية ، أو يسرقوا الجهاز المستخدم للتشفير.

مثال: ففي الحرب العالمية الثانية، تمكن الجنود البولنديين من سرقة الجهاز الألماني الذي كان الألمان يستخدموه للتشفير اسمه (Engima) وتم بيعه للحلفاء وبعدها تمكن هؤلاء الحلفاء من كسر معظم رسائل الألمان.

مثال آخر ، هناك خوارزمية اسمها RC4 اخترعت من قبل شركه RSA في عام 1987 لكن لم تنشر ، كل الـ cryptanalysts والمشفرين اجمعوا في ذلك الوقت أن هذه الخوارزمية آمنه جدا وتجعل البيانات سريه للغاية ، ولم تنشر تلك الخوارزمية لأغراض بيع برامج للتشفير (وليس لأغراض عسكريه) ، المهم في 1994 قام احد الهكرز بوضع الخوارزمية مشروحة بالتفصيل في الانترنت ! كيف عرف هذا الهكرز الخوارزمية ؟؟ بالتأكيد من خلال برامج Disassembly And Debugger، وهي برامج تستخدم لفتح الملفات التنفيذية وتتبعها سطر بسطر وتغيير وتتبع الشيفرات.

إذاً: لايمكن ابقاء الخوارزمية مخفية، وفرضاً إن استطعنا ذلك: فلا حاجة لنا بها... لأننا سنستطيع اخفاء النص أيضاً.

ما أهمية وجود المفتاح في الخوارزميات؟ لماذا نحتاج إلى مفتاح؟

إما أن نثق بالشركة منتجة الخوارزمية ونمنحها أسرارنا ، أو أن نبقي شيئاً ما مخفي.. كمفتاح سري، فالمفتاح يجعلنا نشعر بالارتياح التام ، لأننا اذا شفرنا الخوارزمية باستخدام المفتاح ، سوف تكون مهمتنا الحفاظ على المفتاح فقط ، بالتأكيد هو أسهل بكثير من الحفاظ على الخوارزمية.

أيضا في حال استخدمنا مفتاح تشفير لكل رسالة ، وتم كسر احد المفاتيح ، فان باقي الرسائل تكون سرية وغير مكشوفة. على العكس اذا استخدمت خوارزمية من تطويرنا وتم كشفها فان كل الرسائل سوف تنكشف أيضا.

التشفير بالطرق الحديثة (Encryption in Modern Method)

هناك نوعين من أنواع التشفير المتناظر الحديث هما:

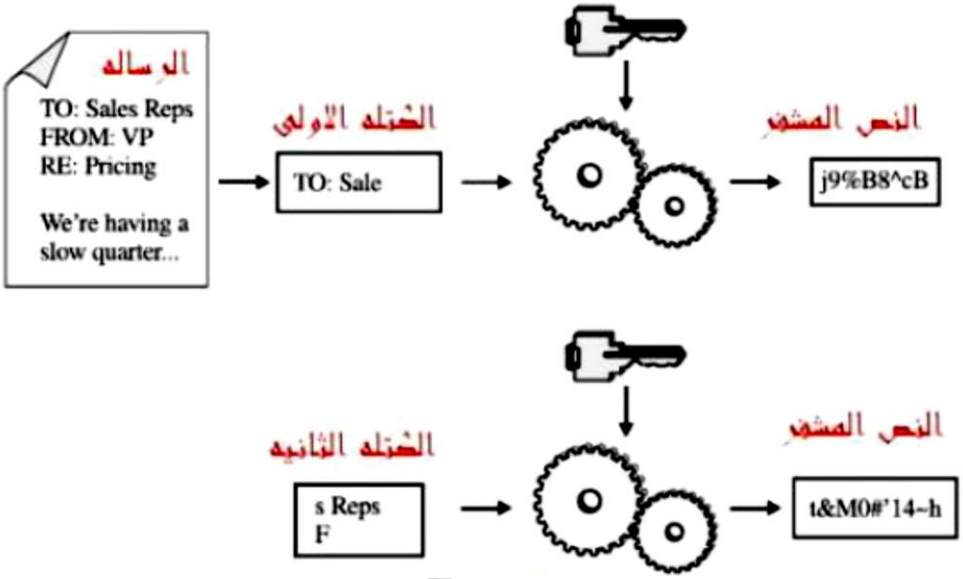
1- شيفرات الكتل (Block Cipher)

2- شيفرات التدفق (Stream Cipher)

لنتحدث عن كليهما:

1- شيفرات الكتل (Block Cipher)

شفرات الكتل (كما هو واضح من اسمها) تعمل على Block (كتله) من البيانات في كل مره ، عندما تزود الخوارزمية (أو برنامج التشفير) بالنص الأصلي هنا في هذا النوع يقوم بتقسيم النص إلى عده من الكتل ، كل كتله حجمها يكون 64 بت وأحيانا 128 بت (16 بايت) ، طبعا الحجم بيكون على حسب الخوارزمية المستخدمة



لنفرض أن طول النص الأصلي 227 بايت ، والخوارزمية تأخذ 16 بايت في كل مره ، الآن في مرحله التشفير ، ندخل المفتاح ، والكتلة الأولى (أول 16 بايت) ، ونبدأ عمليه التشفير ، والناتج هو نص مشفر بطول 16 بايت أيضا . بعدها نأخذ الكتلة الثانية (ثاني 16 بايت) ، و نبدأ في عمليه التشفير ، والناتج هو أيضا نص مشفر بطول 16 بايت ، وهكذا ستستمر العملية 14 مره (وهنا تكون شفرت 224 بايت من النص الأصلي).

الآن بقيت 3 بايتات ولكن الخوارزمية لا تعمل إلا اذا كانت الكتلة بطول 16 بايت ، اذا ما العمل ؟؟ هنا سوف تستخدم مفهوم جديد اسمه **الحشو Padding** وهو بكل بساطه يعمل على اضافه بايتات اضافيه إلى الكتلة الناقصة حتى تكتمل وتصبح بالحجم المطلوب. وهناك عده طرق للحشو سوف نذكر اشهرها ، على العموم في حال استخدمنا أي طريقة يجب أن تكون مفهومة للشخص الذي نريده أن يفك تشفير الرسالة أي يعرف هذه البايتات هي بايتات اضافيه.

الطريقه الأشهر للحشو : هي أن نعرف أولا عدد البايتات التي سوف نضيفها إلى الكتلة الناقصة ، في المثال السابق كانت (13 بايت) ، بعدها نقوم بتكرار هذا العدد في كل بايت في الكتلة ، أي انه سنحشو العدد "13" ثلاثة عشر مره ، وتسمى هذه الطريقه **PKC#5**

في عمليه فك التشفير ، سننظر إلى الكتلة الاخيره ونرى إن كان هناك عدد ما يتكرر في كل خانه من خانات الكتلة الاخيره ، ومنه سوف نعرف هل كان هناك حشو أو لا .

طبعا عند التشفير ، في حاله الكتلة الاخيره تساوي 16 بايت فلا يوجد داعي للحشو (أصلا لا يوجد مكان للحشو).

المشكلة في هذا النوع "شفره الكتل " Block Cipher هو انه في حاله كانت هناك كلمات تتكرر كثيرا في النص الأصلي ، فإنها بعد التشفير سوف يكون النص المشفر متشابه أيضا .

بفرض أن الاسم "Basil Dhimiesh" ظهر 3 مرات في مواقع مختلفة في النص الأصلي. مثلا ظهر في الكتلة الأولى) أول 16 بايت (والكتلة الرابعة) رابع 16 بايت (والكتلة العاشرة)عاشر 16 بايت.

الآن عند التشفير

دخلت الكتلة الأولى "Basil Dhimiesh" وتشفر إلى "Selrurjgqmf"

دخلت الكتلة الرابعة "Basil Dhimiesh" وتشفر إلى "Selrurjgqmf"

دخلت الكتلة العاشرة "Basil Dhimiesh" وتشفر إلى "Selrurjgqmf"

نلاحظ انه يمكن للمخترق معرفة أن الثلاثة "Selrurjgqmf" هي عبارة عن جملة واحده ، وهنا ممكن أن يكسرها بسهولة.

الحل: لتجنب مثل هذا التكرار هو استخدام أساليب تسمى أنماط العمليات (Mode Of Operation).وهي خمسة أنماط:

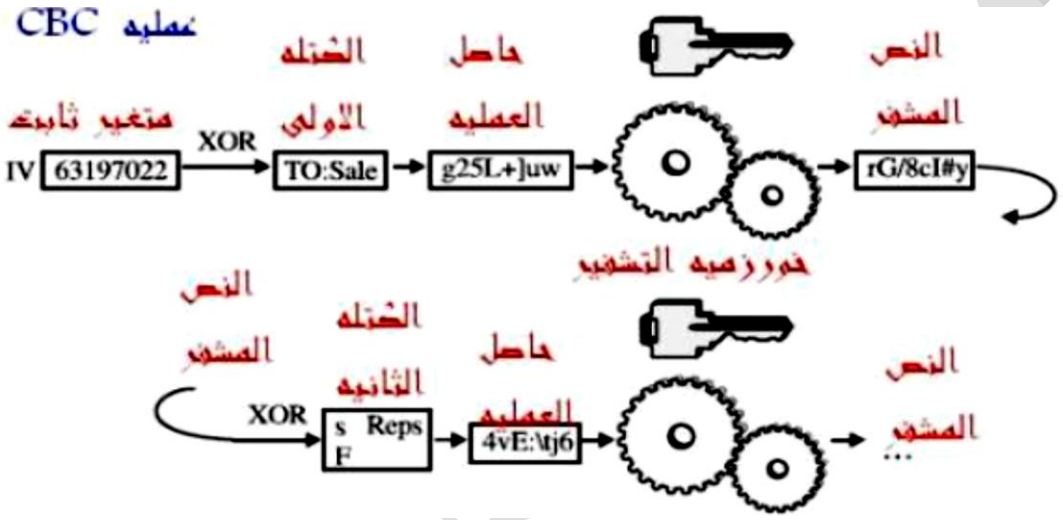
- Electronic codebook mode (ECB)
- Cipher block chaining mode (CBC) – most popular
- Output feedback mode (OFB)
- Cipher feedback mode (CFB)
- Counter mode (CTR)

من اشهر هذه الأساليب هو نمط سلسلة كتل التشفير (Cipher Block Chaining :CBC). في هذا النمط سوف نطبق العملية XOR على النص الأصلي الحالي مع النص المشفر السابق.

أي:

كتلة النص الأصلي الحالي XOR كتلة النص المشفر السابقة، وبعدها سوف نجري عملية التشفير على النص الناتج من عملية XOR.

بالنسبة إلى كتله النص الأول ، لن يكون هناك نص مشفر سابق لذلك سوف نطبق العملية مع متغير اسمه ***initialization vector*** اختصارا **IV** .



وهكذا سوف ننهي مشكلة تكرار البيانات بهذه الطريقة.

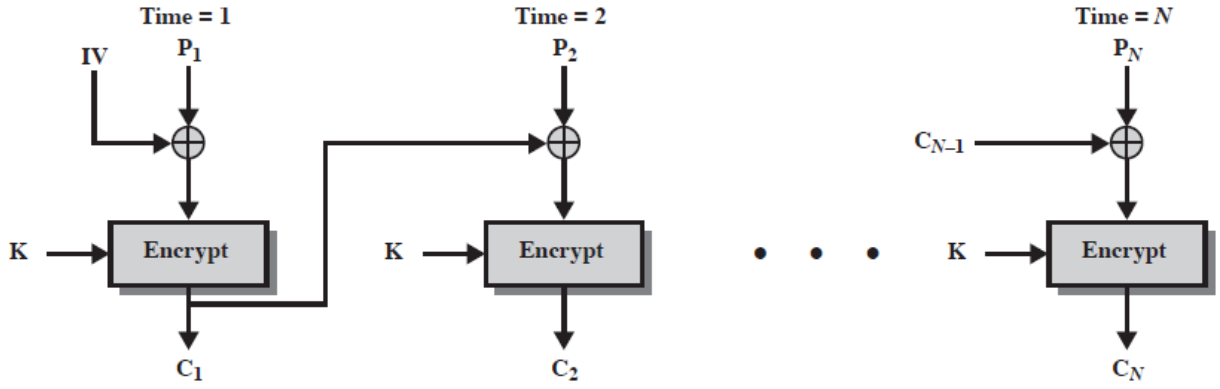
بطريقة أوضح يمكن تمثيل عملية CBC أثناء التشفير بالشكل (a) والممثل بالعلاقة:

$$C_i = \mathrm{E}_K\left(C_{i-1} \oplus P_i\right)$$

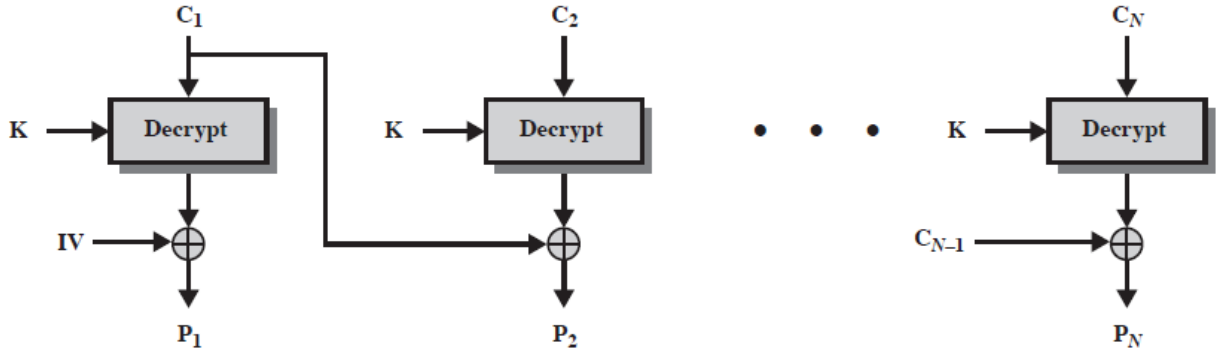
$$C_0 = \mathrm{IV}$$

وأثناء فك التشفير بالشكل (b) والممثل بالعلاقة:

$$P_i = C_{i-1} \oplus \mathrm{D}_K(C_i)$$



(a) Encryption



(b) Decryption

2- شفرات التدفق

النوع الثاني من أنواع التشفير بالمفتاح المتناظر هو شفره التدفق ، و هنا سوف نتعامل مع بت بت أو بايت بايت وليس كتله كتله ، وعمليه توليد المفتاح **Key Stream** من الممكن أن تعتمد على النص المشفر السابق ومن الممكن لا

أيهما أفضل: شيفرات التدفق أم شيفرات الكتل؟

شفره التدفق هي أسرع بكثير من شفرات الكتل وعمليه كتابه برامج سهله وأكودها اقل بكثير من الكتل ، واحد اشهر أنواع شفرات التدفق RC4 وهي أسرع بكثير من أي نوع من أنواع شفرات الكتل ، وتتطلب حوالي 30 سطر فقط في الكود . معظم شفرات الكتل تأخذ على الأقل 200-400 سطر.

شفرات الكتل من جهة أخرى تسمح باعاده استخدام المفتاح ، بعكس شفرات التدفق التي تستخدم المفتاح مره واحده فقط ، في الكثير من الأحيان يجب أن نشفر العديد من الأشياء بمفتاح واحد.

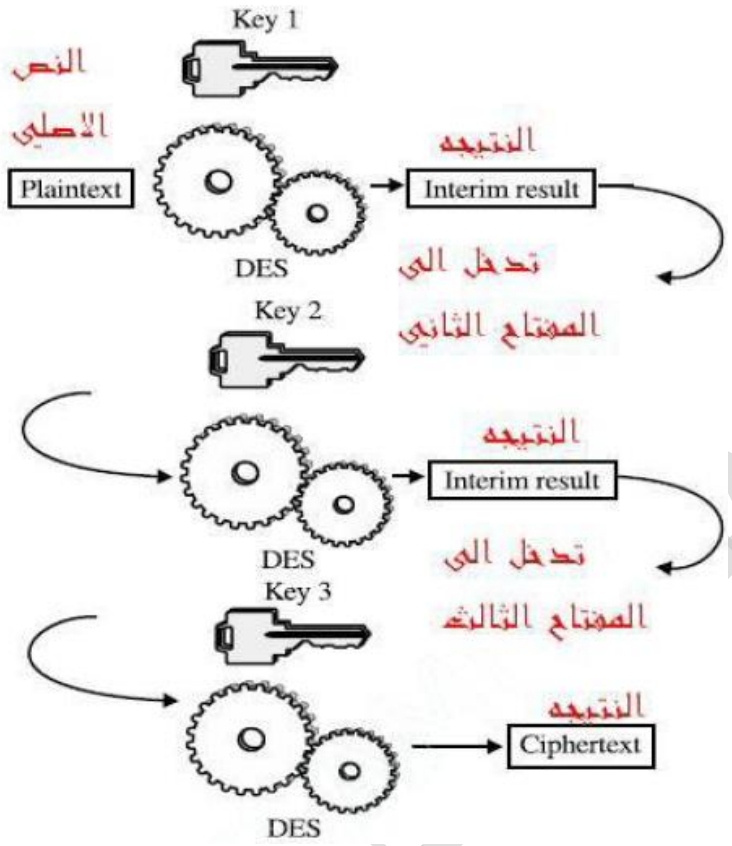
مثال ، شركه لديها قاعدة بيانات ضخمه للعملاء تحتوي معلوماتهم من أرقام هواتف وبطاقات ائتمانية وغيرها ، في حال استخدمت شفرات التدفق سوف تتطلب لكل مدخل (عميل) مفتاح خاص وهذا يتطلب مئات من المفاتيح وهو أمر غير عملي ، أما في حاله استخدمت شفرات الكتل فإنها تشفر جميع البيانات باستخدام مفتاح واحد ، ولفك تشفير بيانات أي عميل نستخدم نفس المفتاح . عمليه اداره المفتاح أسهل بكثير في هذه الحالة.
لذلك في معظم قواعد البيانات يتم استخدام شفرات الكتل Block Cipher وأيضا في برامج البريد الالكتروني ، وأيضا في برامج تشفير الملفات.

خوارزمية معيار التشفير الرقمي (Digital Encryption Standard: DES)

في بدايه السبعينات تم معرفه انه اغلب الشفرات القديمة لم تعد مجديه وغير نافعة للتشفير ، ولهذا قرر علماء في شركه IBM بعمل خوارزمية جديدة للتشفير تبنى على بنية قديمة تسمى Lucifer (نسبه إلى مخترعها Horst Feistel) ، ومن خلال مساعده وكاله الأمن القومي NSA تم عمل خوارزمية DES .
DES هي احد شفرات الكتل Block Cipher ، وتأخذ مفتاح بطول 56 بت ، وتعمل على كتله طولها 64 بت .
وفي الثمانينات لم يتم اكتشاف أي ثغره في DES لذلك كانت اقوي الخوارزميات في ذلك الوقت، ولكسر أي رسالة مشفره بها لم يكن هناك إلا استخدام هجوم الـ brute-force ، ولأن طول المفتاح 56 بت (مداه من 0 إلى 72 كوارلديون) و الاجهزه بطيئة للغاية ، فكانت عمليه الكسر تتطلب سنه كاملة.
وفي 1999 وفي احد المؤتمرات تم كسر هذه الخوارزمية في 24 ساعة من قبل the Electronic Frontier Foundation اذا العالم يجب أن ينتقل إلى خوارزمية أخرى

Triple DES

احد البدائل كانت خوارزمية Triple DES أو البعض يسموها 3DES ، هي بكل بساطه DES ولكن ثلاثة مرات ، يعني سوف تدخل الكتلة الأولى (16 بايت) إلى الخوارزمية بالمفتاح الأول ، والناتج سوف يدخل إلى الخوارزمية مع المفتاح الثاني ، والناتج سوف يدخل مع المفتاح الثالث .



على العموم DES 3 لها مشكله وهي أنها بطئيه جدا ، الـ DES العادية هي بطيئة ، فما بالك بـ DES ثلاث مرات ، واغلب التطبيقات تتطلب السرعة في العمل ، وهذه الخوارزمية لا تنفع لأنها بطيئة جدا ، اذا العالم مره أخرى بحاجه إلى خوارزمية!!

Advanced Encryption Standard

نتيجة لهذا الأمر قام المعهد الوطني للمعايير National Institute of Standards and Technology اختصارا NIST ، باستدعاء جميع المهتمين بهذا الأمر وكلفت بكل منهم بعمل خوارزميته الخاصة وفي النهاية أقوى خوارزمية سوف تكون هي المقياس الجديد AES ، وقد قدمت 15 خوارزمية (منها القوي ومنها الضعيف) .

وفي 1999 قامت NIST باختيار أفضل 5 خوارزميات بعد إجراء العديد من الاختبارات ، وقد جعلت الأمر بالتصويت لأفضل خوارزمية ، وفي 2000 تم اختيار خوارزمية **Rijndael** كالمقياس الجديد AES .

إدارة المفتاح المتناظر Symmetric-Key Management

توصلنا سابقا إلى أن التشفير بالمفتاح المتناظر يقوم بتشفير الرسالة بمفتاح ما ، ثم يقوم بفك التشفير بنفس المفتاح ، لذلك عملية الحفاظ على المفتاح أمر في غاية الأهمية ، فإذا انكشف المفتاح انكشفت جميع الأسرار، لذلك يجب حفظ المفتاح في مكان امن جدا ، عملية الحفاظ على المفتاح تسمى بإدارة المفتاح (Symmetric-Key Management)

ربما الآن تتساءل " اذا كان هناك مكان أستطيع أن أحفظ في المفتاح ، فلماذا لا أحفظ الرسالة في ذلك المكان ولا احتاج إلى التشفير "؟

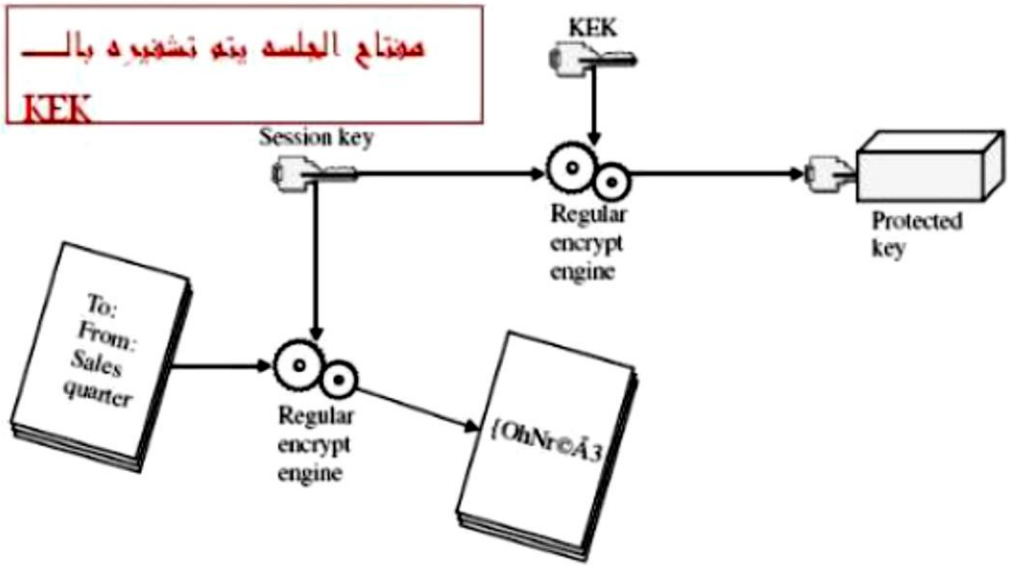
في الحقيقة حفظ مفتاح التشفير (56 بت مثلا) يكون أسهل كثيرا من حفظ الرسالة (بعض الأحيان حجمها يكون مئات من الميغا بايت MB) ، بالاضافه إلى هناك حلول لحفظ المفتاح عن طريق حفظها داخل أجهزه صغيره مصممة لهذا الغرض.

التشفير المعتمد على كلمة مرور (Password-Based Encryption)

إن المفتاح الذي كنا نستخدمه للتشفير وفك التشفير يسمى في الحقيقة "بمفتاح الجلسة session key" ، وأحد الطرق لحماية هذا المفتاح هي عن طريق تشفيره أيضا ، أي أن المفتاح (مفتاح الجلسة) يحتاج إلى مفتاح آخر لكي يتم تشفيره.

أحد تقنيات حماية مفتاح الجلسة وتشفيره هي التشفير المعتمد على كلمة مرور (**password-based encryption: PBE**)

يعني مفتاح الجلسة session key هو المفتاح الذي نستخدمه في التشفير وفك التشفير
ومفتاح تشفير المفتاح key encryption key هو المفتاح الذي نستخدمه لتشفير مفتاح الجلسة ، واختصارا يسمى KEK .

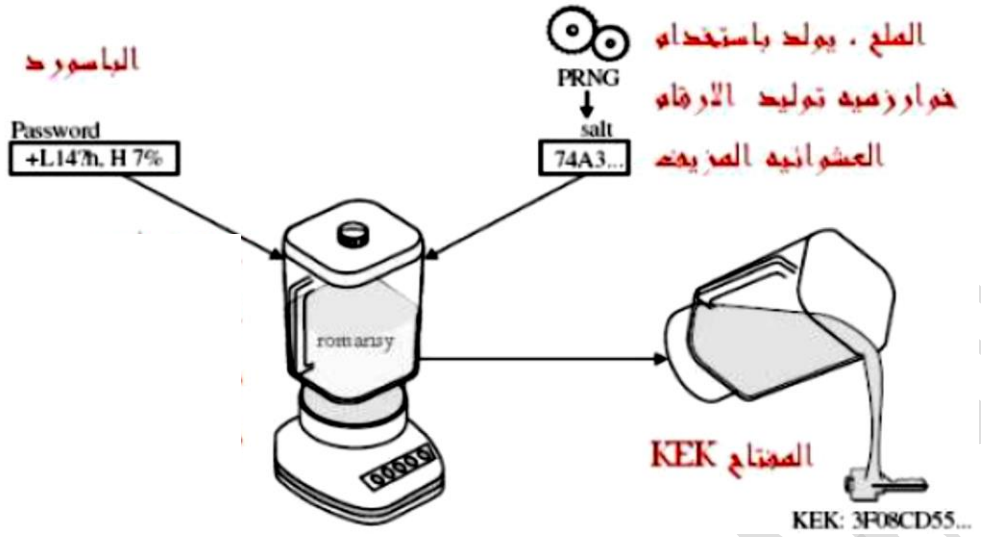


الآن بما أن المفتاح KEK (من الآن وصاعدا نسميه بهذا الاسم) هو الذي يستخدم لتشفير وفك تشفير مفتاح الجلسة ، السؤال هل أنا بحاجه إلى حماية هذا الـ KEK ؟
الجواب ، هو لا ، عندما نريد أن نشفر المعلومات نقوم بتوليد هذا المفتاح (بأحد طرق توليد الأرقام العشوائية) بعدها نقوم باستخدامه ومن ثم نحذفه ، وفي حاله فك التشفير نقوم بتوليد هذا المفتاح مره أخرى ونستخدمه ومن ثم نحذفه ، وفي مرحله توليد هذا المفتاح يجب أن ندخل باسورد معين سواء في مرحله التشفير أو فك التشفير .

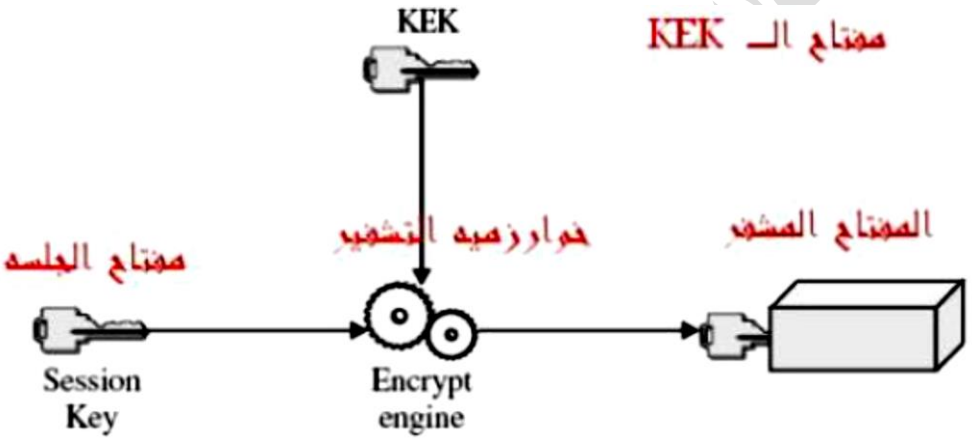
بصورة مبسطة ، مفتاح الجلسة Session Key هو الذي يشفر المعلومات
أما مفتاح KEK فهو الذي يشفر مفتاح الجلسة ونقوم بتوليده عن طريق password-based encryption

ويتم توليد الـ KEK :

-1- إدخال باسورد
-2- استخدام أي طريقه لتوليد أرقام عشوائية لتوليد الـ salt (الملح) .
-3- ندخل الباسورد والملح مع بعض داخل الخلاط blender والناتج هو خليط من البتات العشوائية .
-4- نأخذ ما يكفي من الخليط السابق ونضعه داخل المفتاح KEK ، وبعدها نستخدم الـ KEK لتشفير مفتاح الجلسة ثم نحذف هذا الـ KEK ، ونحتفظ بالملح .
-5-الآن تم تشفير مفتاح الجلسة ، ويجب آن تحفظ الملح لأنه سوف يستخدم في فك التشفير
يوضح الشكلان التاليان هذه المراحل



ثم يستخدم KEK كما يلي:



الآن لفك التشفير:

-1- ندخل الباسورد الذي أدخلناه في عمليه التشفير

-2- نأتي بالملح الذي احتفظنا به في مرحله التشفير

-3- ندخل الملح والباسورد في نفس الخلاط الذي استخدمناه في عمليه التشفير ، في حال اختلف احدهم سوف يكون الناتج عبارة عن KEK خاطئ ، وفي حال كانوا صحيحين فالناتج هو الـ KEKالصحيح

-4- نستخدم الـ KEK لفك مفتاح الجلسة ، وبعدها نستخدم مفتاح الجلسة لفك تشفير الرسالة..

االآن وبعد تشفير مفتاح الجلسة باستخدام مفتاح KEK ، هل تعتبر في أمان كامل من جميع االهجمات ؟

بالطبع لا ، لان المخترق بإمكانه عمل هجوم على المفتاح KEK (هجوم القوه العنيفة Brute Force attackويقوم بتجربة مفتاح مفتاح إلى أن يصل إلى المطلوب .

أو بإمكانه عمل هجوم على الباسورد (Brute Force Attack) ، ويقوم بإدخال الباسورد والملح في الخلاط ، بعدها يأخذ الناتج KEK ويفك تشفير مفتاح الجلسة وبعدها يفك تشفير البيانات ، وإذا لم يصلح الباسورد يقوم بتغييره واختيار واحد أخر .

قد تبدو العملية طويلة ، لكن في الحقيقة أساليب هجوم Brute Force قد تأخذ أساليب متطورة ، مثلا عمل البرنامج **بالتوازي in parallel** وهنا سوف يستفيد من عمل المعالج بشكل كبير ، أيضا من الممكن أن يعمل أكثر من جهاز في عمليه الكسر .

أيضا من الممكن أن يقوم المخترق باستخدام **هجوم القاموس dictionary Attack** وهنا يقوم بعمل قاعدة بيانات لأغلب الباسوردات في جميع اللغات ، بعدها يقوم بتجربة هذه الباسوردات . وهذا الهجوم بالطبع أسرع من هجوم الـ Brute Force لأنه يكون محدود على مجموعه من الباسوردات .

ملاحظة:

لقد رأينا قبل قليل أن احد طرق حفظ المفتاح هو استخدام الـ PBE ، احد الحلول الأخرى هي استخدام أجهزه خاصة لحفظ المفاتيح ، بعض هذه الاجهزه صغير جدا ويسمى **Token** ، وبعضها كبير ويسمى **crypto accelerators** .

مشكلة توزيع وإرسال المفتاح (The Key Distribution Problem)

تعرفنا قبل قليل على بعضا من أساليب حماية المفتاح (مفتاح الجلسة) ، ويكون إما عن طريق تشفيره مره أخرى (PBE) ، أو عن طريق تخزين المفتاح في احد الأجهزة الخاصة لذلك Token، إلى هنا الأمر تحت السيطرة ، لكن ماذا اذا أردنا أن نرسل المفتاح إلى شخص آخر حتى يقوم بفك تشفير الرسالة التي سوف أرسلها له (تذكر أن التشفير بالمفتاح المتناظر ، المفتاح نفسه يقوم بالتشفير وفك التشفير).

بعبارة مبسطه ، في حال قمت بتشفير رسالة ما بهذا المفتاح المتناظر ، بعدها أرسلت الرسالة إلى الشخص الذي أريد ، في حال وصلت الرسالة للشخص هذا سوف تكون غير مفهومه وذلك لان المفتاح الذي يفك التشفير معي والى الآن لم أرسله للشخص المراد ، أيضا في حال كان هناك مخترق ووصلت الرسالة إليه بطريقه ما (سواء قام باختراق جهاز الشخص الذي أرسلت له الرسالة ، أو قام بالتقاط الرسالة أثناء إرسالها) المهم سوف تكون الرسالة أيضا غير مفهومه لأنه لا يملك المفتاح.

اذا السؤال هنا ، كيف يمكن أن أرسل المفتاح بطريقه آمنه إلى الشخص الذي أريد ، وفي نفس الوقت لا يستطيع المخترق الحصول عليه ؟؟

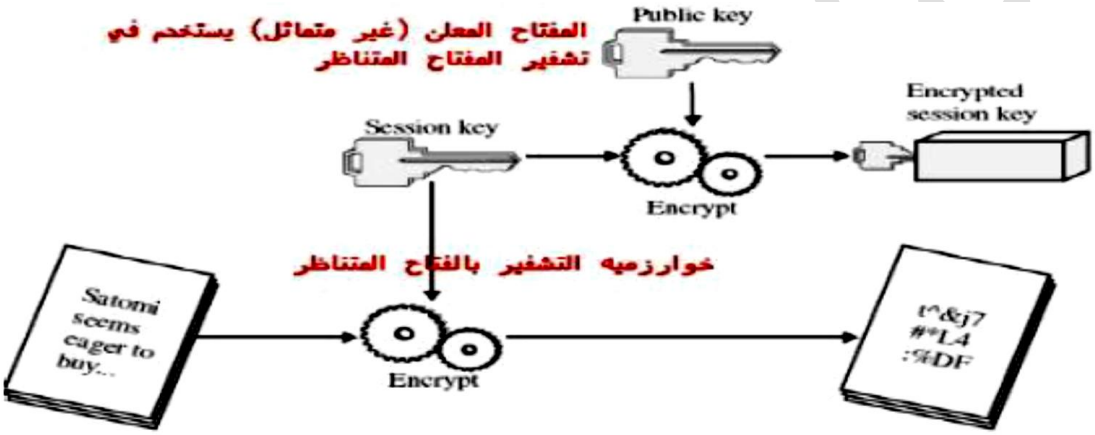
هذه المشكلة تسمى بمشكله **إرسال المفتاح Key Distribution Problem** ، والتي بسببها تم اختراع الطريقه الأخرى في التشفير وهي التشفير بالمفتاح غير المتناظر Asymmetric key Cryptography .

خوارزميات التشفير غير المتناظرة

في السبعينيات تم اختراع هذه الطريقة ، وهي تستخدم مفتاحين ، مفتاح عام public Key عادة يستخدم للتشفير ومفتاح خاص private Key عادة يستخدم لفك التشفير.

- **المفتاح العام public Key** يكون معروف للجميع وأي شخص يستطيع الحصول عليه (هو يستخدم للتشفير) .
- **المفتاح الخاص private Key** يكون غير معروف (معروف لشخص واحد) وهو يستخدم لفك التشفير

يستخدم المفتاح العام أيضاً لتشفير مفتاح الجلسة في التشفير المتناظر كما يوضح ذلك الشكل التالي:



لماذا قمنا بتشفير الرسالة بخوارزمية التشفير بالمفتاح المتناظر ؟

ولذلك بسبب **السرعة والأداء** ، فالتشفير بالمفتاح المتناظر أسرع بكثير من التشفير بالمفتاح الغير متناظر ،

التشفير بالمفتاح المتناظر يشفر 50 MB في الثانية الواحدة

التشفير بالمفتاح غير المتناظر يشفر 20-200 KB في الثانية الواحدة . –لاحظ الفرق- .

لذلك يستخدم المفتاح المتناظر لتشفير الرسالة ، ويستخدم المفتاح الغير متناظر (العام) لتشفير المفتاح المتناظر ، العملية السابقة تسمى بالظرف الرقمي Digital Envelope .

هذه العملية مشابه لعمليه Password Based Encryption ، لأنها في PBE نقوم باختيار خوارزمية لتشفير النص من نوع Symmetric ، بعدها نقوم بتوليد مفتاح الجلسة ، ونقوم بتشفير مفتاح الجلسة باستخدام PBE ، ولا يمكن لأي أحد فك تشفير مفتاح الجلسة إلا في حال كان يملك الباسورد الصحيح .

أما هنا في Digital Envelope ، نقوم باختيار خوارزمية لتشفير الرسالة من نوع Symmetric ، ونقوم بتوليد مفتاح الجلسة ، بعدها نحصل على المفتاح العام من الطرف الأخر الذي أريد إرسال الرسالة إليه ، وأقوم بتشفير مفتاح الجلسة بهذا المفتاح العام ، وأقوم بإرسال النص المشفر (بـSymmetric) أضافه إلى مفتاح الجلسة المشفر (بالمفتاح العام) للطرف الأخر ، و هنا يقوم الطرف الأخر باستلام الرسالة ، ويقوم بفك تشفير مفتاح الجلسة باستخدام المفتاح الخاص به ، وبعدها يحصل على مفتاح الجلسة ، ومنه يقوم بفك تشفير الرسالة .

لاحظ أن هذه الطريقه قد حلت مشكله توزيع المفاتيح ، حيث كل ما الأمر استخدام طريقه public Key Cryptography ، وبعدها لا حاجه لطرف ثالث ، أو توزيع المفاتيح قبل بدء الإرسال .

وفي عام 1977 قام البروفيسور Ron Rivest من معهد MIT ، مع زملائه , Adi Shamir Len Adleman بالاهتمام بهذه الخوارزمية DH ، وقاموا بتطبيقها لتكون أول خوارزمية من نوع Public key Cryptography وتمت تسميتها باسم **RSA** (الحرف الأول من كل اسم) .

ومن الأمثلة أيضاً خوارزمية Diffie-Hellman

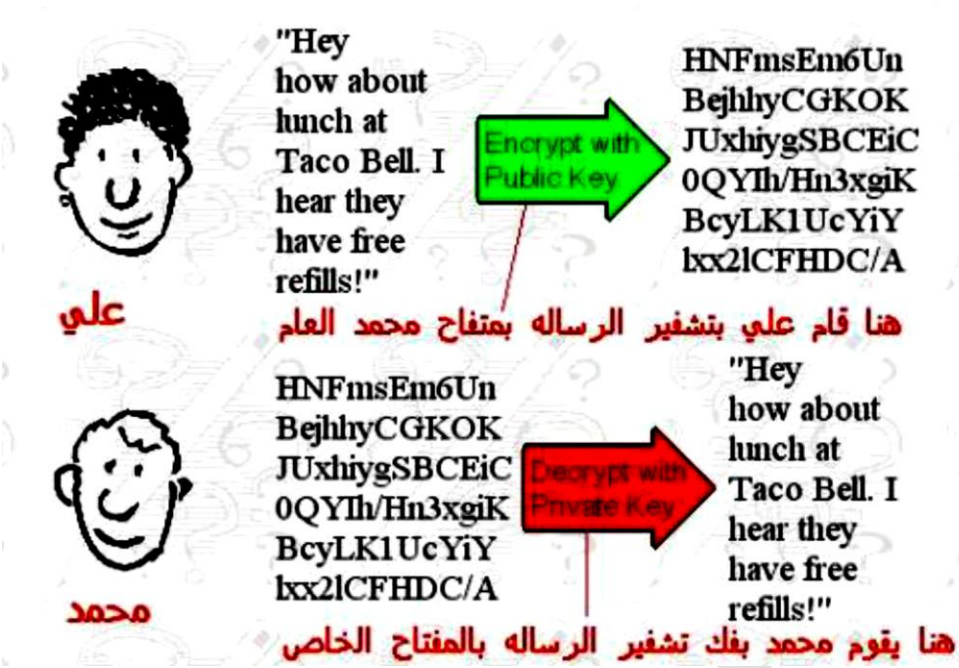
التوقيع الرقمي وتلخيص الرسائل بدوال الهاش[8] Digital Signature and Message Digest

نأخذ مثال بسيط لكي يوضح طريقة التشفير بالمفتاح العام والخاص ومن ثم كيفية استخدام التوقيع الرقمي المستخدمة في التوثيق (Authentication)، وسنتحدث عن تبادل رسائل بين شخصين هما محمد وعلي.

محمد لديه مفتاحين ، احدهما مفتاح عام والآخر مفتاح خاص (التشفير بالمفتاح غير المتناظر).

الآن هو يريد إرسال رسالة لأحد أصدقائه ، كل ما عليه إرسال المفتاح العام (يكون موجه للجميع) ، أما المفتاح الخاص فيحتفظ به لنفسه ، المفتاح العام يستخدم للتشفير والخاص لفك التشفير .

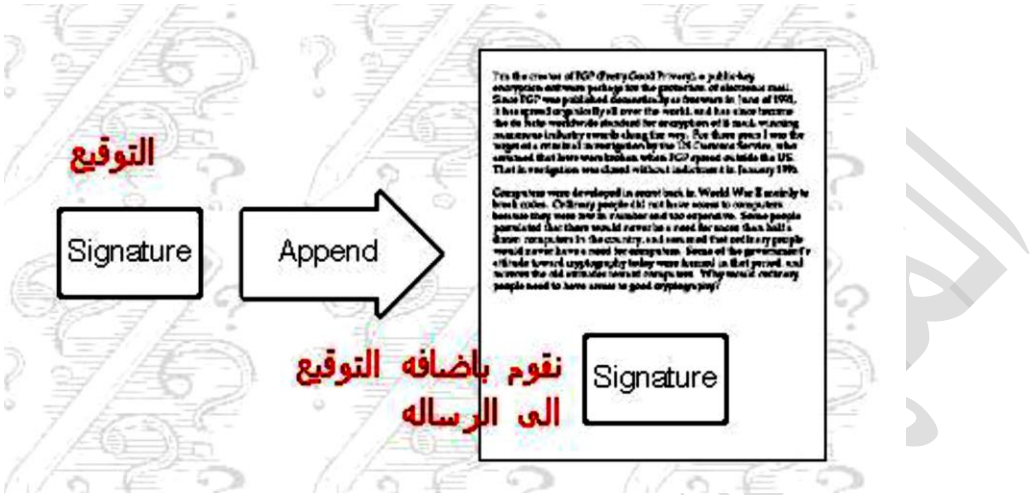
[8] تسمى خوارزمية الهاش بخوارزمية الإختزال أيضاً.



في حاله أراد محمد إرسال رسالة إلى علي وكان يريد أن استخدام التوقيع الرقمي (**الذي عن طريقه يستطيع علي التأكد من أن محمد هو مرسل الرسالة ، أيضا يمكن معرفه أي تغيير حصل على الرسالة أثناء إرساله**) ، كل ما على محمد هو أن يأخذ الرسالة بعد كتابتها ويدخلها إلى احد الدوال الهاشية (مثل md5 و SHA-1 وغيرها) لكي يخرج الناتج (يسمى بالهاش أو message digest).

بعد ذلك يقوم محمد بأخذ هذا الهاش ويقوم بتشفيره باستخدام المفتاح الخاص به ، وهكذا يحصل محمد على التوقيع الرقمي.

الآن يقوم محمد بإضافة التوقيع الرقمي (Digital Signature) إلى الرسالة التي يريد إرسالها .. ويرسلها إلى علي.



وصلت الرسالة إلى علي ، يقوم علي (أو البرنامج الذي يستخدمه) بفك تشفير التوقيع (الناتج هو الهاش) باستخدام المفتاح العام لمحمد ، في حال انفك بشكل صحيح ، يكون علي قد عرف أن المرسل هو محمد وليس أي احد آخر..

أيضا يقوم بتطبيق الدالة الهاشية (التي طبقها محمد) على الرسالة ، في حال تساوت مع الهاش ، اذا يكون علي قد عرف أن الرسالة لم تتغير أثناء إرسالها..

خوارزمية التشفير MD5

سنتحدث عن دالة الاختزال "Hash Function " أنواعها وسلسلة خوارزميات التشفير المسماة تلخيص الرسالة "Message Digest" والتي بدأت بـ "MD2 " ثم "MD4" واخيراً خوارزمية التشفير (MD5) والتي هي تطوير لسابقتها "MD4" والتي أصبحت فيما بعد من أشهر دوال الاختزال وتطبيقاتها في صحة الملفات "File Integrity " و التواقيع الرقمية "Digital Signature" و إثباتات الهوية الإلكترونية "Authentication " بالإضافة إلى طريقة عملها من حجز المساحة التخزينيه وتكون بعض الجداول بالإضافة إلى المعادلات المنطقية المستخدمة لتوليدها وخواصها العامة وأخيراً عن احتمالات الهجوم عليها وكسرها عن طريق الـ "Brute Force" أو عن طريق ملف "Rainbow Table" وغيرها من الطرق .

عبر الزمن تطورت هذه الأساليب كثيراً حتى وصلت بصمة الإصبع ، التوقيع والختوم المصدقة ومع ثورة المعلوماتية الرقمية الحديثة كان لابد لهذه الأساليب أن تواكب ركب التقنية وبناء وسائل رقميه لإثبات الهوية ، إثبات الهوية الرقمي عادة مايتم باستخدام إحدى طرق التشفير والتي يطلق عليها دالة الاختزال أو الـ (Hash Function) والتي هي بوجه العموم مجموعه من العمليات التي تنفذ على نص بطول متغير من الحروف وتنتج نص مشفر بطول ثابت دائما يستحيل فهمه والعودة منه سلفاً للنص الأصلي ، ومن أنواع دالة الاختزال الـ (Hash Function) سلسلة تلخيص الرسالة (Message Digest) والمختصرة بـ (MD) والسلسلة تتضمن MD2 و MD4 و أخيراً والتي هي محورنا التالي MD5.

خوارزمية التشفير (MD5): خوارزمية التشفير (Md5) لا تختلف كثيراً عن سابقتها MD4 ، هي دالة تشفير يدخل لها نص بأي طول ويتم تجزئته إلى نصوص قصيرة بحجم 512 bit وتنتج نص مشفر بطول 128 bit يمثل بـ (32 رقم ست عشري Hexadecimal) ، ناتج عملية التلخيص صعب الرجوع منه للنص الأصلي ، مثال : النص الأصلي : Message Digest 5

نتيجة الـ (MD5) له : b88402ac7072606ec70f190ba5dd0211 هذه الطريقة عادة ماتستخدم في إثبات صحة الملفات من التعديل و هويات المستخدمين والتواقيع الرقمية والتي سنرى تفصيلها لاحقاً .

خواص خوارزمية التشفير (Md5) :

1- طول نتيجة الخوارزمية ومنها دائما مانستدل على نوع الخوارزمية المستخدمة ، على سبيل المثال إذا عرفنا ان ناتج العملية هو 128 bit فبالتأكيد أننا استخدمنا الـ (MD5) .

2- خوارزمية التشفير (MD5) لا تعطي نتيجتين متماثلتين لملفين مختلفين ، حينما نجد دالتين للـ (MD5) متماثلتين فبالتأكيد أنهما نتيجة لملفين أو رسالتين متطابقتين تماماً.

3- إذا قمنا بعمل خوارزمية التشفير (MD5) لملف أو رسالة معينه أكثر من مره في أوقات مختلفة دون التعديل عليها فإن نفس نتيجة الخوارزمية سوف تتكرر في كل مره .

4- خوارزمية التشفير (MD5) ذات طريق واحد، أي انه من نص عادي تعطي نص مشفر لكن لا نستطيع أن نصل من النص المشفر إلى نص عادي.

طريقة عمل خوارزمية التشفير (MD5):

يتم تقسيم النص إلى أجزاء كل جزء بحجم bit512 ويسمى (Block) وإذا كان الجزء اقل من هذا الحجم يتم زيادة بعض الحروف إلا أن يصل إلى 512 bit هذه الحروف تسمى بالحشو (padding) .

كل جزء بحجم bit512 يتم تجزئته إلى 16 كلمه "Word" كل كلمه بحجم bit32.

تحجز مساحه تخزينيه لأربع كلمات A,B,C,D وتكون في بداية الأمر كلمات إنشائية تحوي القيم التالية

A: 01 23 45 67

B: 89 ab cd ef

C: fe dc ba 98

D: 76 54 32 10

يتم إنشاء جدول مكون من 64 خانه كل خانه يتم حساب محتواها عن طريق هذه المعادلة:

Ki=abs (sin(1+i))*232 حيث الـ i ترمز لرقم الخانة في الجدول .

حيث abs تعني القيمة المطلقة و sin هي دالة الجيب المثلثية ، أي أن جميع محتويات هذا الجدول بوحدة الراديان.

لدينا اربع معادلات منطقية جميع هذه المعادلات تطبق على كل جزء مكون من 512bit وتسمى كل مرحله يمر فيها هذا الجزء على احد هذه المعادلات بالدورة أو (Round) .

الدورة الأولى : يدخل لها الأربع كلمات انشائيه A,B,C,D

وباستخدام هذه المعادلة المنطقية

F(B,C,D)=(B ^ C)ᵛ(!(B) ^ D)

الدورة الثانية: يدخل لها ناتج الدورة السابقة للأربع كلمات وتتم هذه المعادلة

$G(B,C,D) = (B \wedge C) \vee (C \wedge \sim D)$

الدورة الثالثة: ويدخل لها ناتج الدورة السابقة وتتم هذه المعادلة

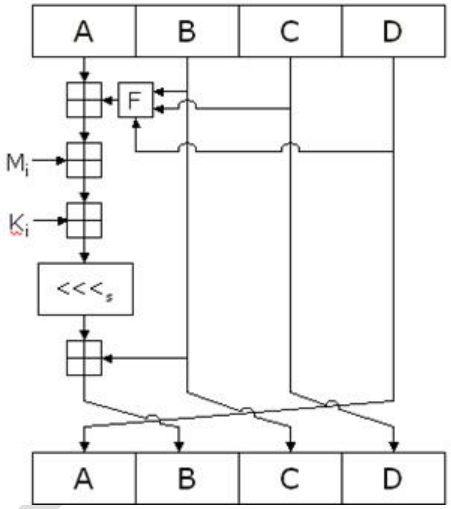
$H(B,C,D) = B \oplus C \oplus D$

الدورة الرابعة: يدخل لها ناتج الدورة السابقة وتتم هذه المعادلة

$I(B,C,D) = C \oplus (B \vee \sim D)$ حيث:

^ هي العملية المنطقية " و". v هي العملية المنطقية " أو " . $\oplus$ هي العملية المنطقية "أو الحصريه XOR " . ~ هي العملية المنطقية " نفي ".

تفصيل جميع العمليات السابقة والتي تنفذ على كل جزء 512bit بشكل عام تتم بالمعادلة التالية بالإضافة إلى الصورة التوضيحية



المعادلة هي

$A \leftarrow B + ((A + F(B, C, D) + M[i] + K[i]) <<< s)$

حيث:

A,B,C,D : هي المساحة التخزينية .

F: هنا تعني ناتج المعادلة المنطقية المنفذة في هذه الدورة .

Ki : هي قيمة الخانة في الجدول الذي تم انشاؤه في الخطوة رقم 4.

s <<< مقدار الإزاحه إلى اليسار بمقدار قيمة الـ S .

تطبيقات خوارزمية التشفير (MD5) :

إثبات صحة الملفات (File Integrity): حينما يكون لدينا ملف للمشاركة قراءة دون تعديل ونريد ان نتأكد دوماً أنه لم يتعرض لأي تحرير من قبل اشخاص غير مصرح بهم فكل ماعلينا هو حساب دالة الـ (MD5) قبل عرض الملف للمشاركة وحفظ النتيجه في قاعدة بينات ومن ثم يتم حساب الـ (MD5) بشكل دوري ومقارنة نتيجة الحساب بالقيمه المحفوظه سابقاً ، بمجرد اختلاف القيمتين عن بعضهم يعني ان الملف تعرض للتعديل عن محتواه الأصلي حتى وإن كان التغيير حرف واحد فقط! .

التواقيع الرقمية (Digital Signature): توقيع رقمي يستخدم لإثبات هوية المرسل او الكاتب للملف ، يضمن لنا عدم التعديل على الملف بعد الارسال بواسطة شخص متجسس، وهذه التواقيع من الممكن اضافتها على رسائل مشفره او بدون تشفير. وطريقه عمله أن يقوم المرسل بحساب الـ (MD5) لرسالته وتشفيره بواسطة المفتاح الخاص (private key) به وارسالها بالإضافه للرسالة إلى المستقبل ، وفي حال وصول الرسالة للمستقبل ، يقوم بفك التشفير بواسطة المفتاح المعلن (public key) وحساب الـ(MD5) للرسالة ومقارنتها بـ(MD5) المرسلة اذا تشابهت فيعني ان الرسالة سليمه من التعديل وبإستخدامه للمفتاح المعلن يضمن هوية المرسل .

كلمة المرور(Password):

حينما تضع كلمة مرور على جهاز الحاسب الخاص بك لتزيد الأمان والخصوصيه عليه فإن ما يحدث فعلاً هو تخزين القيمه الناتجه من حساب الـ (MD5) لكلمة المرور خاصتك في ملفات النظام وفي المره القادمه من محاولتك للدخول لحسابك فإن النظام يقوم بحساب الـ (MD5) لكلمة المرور المدخلة وبمقارنتها بالقيمه المحفوظه في ملفات النظام اذا تطابقت القيمتان سوف يسمح لك بالدخول والتعامل مع ملفاتك وفي حين اخطأت في ادخال كلمة المرور بالتالي ستختلف نتيجة الـ (MD5) عن المحفوظه سابقاً وستترك لك 3 محاولات لادخال كلمة المرور مره اخرى والا سيتم اغلاق المحاولات لفتره من الوقت .

كسر خوارزمية التشفير (MD5):

من الصعب جدا كسر خوارزمية التشفير (MD5) ولكن أكثر الطرق شيوعاً واستخداما في كسر الـ (Md5) خصوصاً في حالات كلمة المرور هي الـBrute Force" "والتي تعتمد على جمع عدد من الكلمات المتوقعه وتشفيرها بالـ (MD5) ومقارنة النتيجه بـ (MD5) الأصلي والمخزن كنص مشفر وهناك الكثير من البرامج لسطح المكتب والتي تساعد في عملية الـ Brute Force""، الطريقه الثانيه هي جمع عدد كبير من النصوص بالإضافه للـ (Md5) الخاصه بها في ملف يدعى " Rainbow Table" ويتم العودة لهذا الملف والمقارنه كلما دعت الحاجه ، بالإضافه إلى ان بعض المواقع على الشبكة العنكبوتية

تقوم بحساب (MD5) لنص معين او العكس لكن بعض هذه المواقع تقوم بحفظ النص الذي ادخله المستخدم و(MD5) الخاص به واضافتها لملف الكلمات" Rainbow Table " واستخدامه في عمليات الـ Brute Force لذا احذر من استخدام مواقع الويب في تجربة تشفير كلمات المرور خاصتك .

الخلاصة:

وتتلخص خوارزمية التشفير MD5 بأنها عمليه تشفير من نوع دالة الإختزالHash Function" " والتي يمر النص فيها بست خطوات تشمل تجزئة النص لأجزاء بحجم 512 bit و تكون الكلمات الإنشائيه و من ثم مرورها في أربع دورات وتميز كل دوره بمعادله منطقيه خاصه بها لتنتج نص غير مفهوم بطول 128 bit ، ويتم الاستفادة من هذه الخوارزمية في عدة تطبيقات منها التوقيع الرقمي واثبات صحة الملفات وفي إثبات الهوية وتشفير كلمة المرور"Password" (المرجع: مركز التميز لأمن المعلومات).