

[ait.hemad.m@gmail.com](mailto:ait.hemad.m@gmail.com)

Pr. M. AIT HEMAD

Cryptographie

|  |
| --- |
|  |
|  |
| Cryptologie |
|  |

|  |
| --- |
|  |
| Introduction à la cryptologie   * Le terme "cryptologie" provient du grec «kruptos», caché et   «logos», étude   * La cryptologie est une science mathématique qui comporte deux branches : la cryptographie et la cryptanalyse.   Cryptologie  « Science du Secret »  Cryptographie Cryptanalyse |

|  |
| --- |
|  |
| Cryptographie   * Le terme "cryptographie" provient du grec   *Kruptos*, "caché", et de *graphein*, "écrire".   * C’est la "science des écritures secrètes" * C’est une science mathématique permettant d’effectuer des opérations sur un texte intelligible afin d’assurer une ou plusieurs propriétés de la sécurité de l’information |



Cryptographie

L'utilisation de la cryptographie a pour objectif d'assurer les grandes fonctions de sécurité suivantes :

▫ Confidentialité

▫ Authentification

▫ Intégrité

▫ Non répudiation

▫ …



Cryptanalyse

* *La cryptanalyse* désigne la science et l'art de l'analyse d'un système de chiffrement
* C’est la science permettant d’étudier les systèmes cryptographiques en vue de les tester ou de les casser.



Cryptanalyse

On distingue plusieurs types d’attaques cryptanalytiques :

▫ Texte chiffré connu (Ciphertext-only attack)

🞄 L’attaquant dispose seulement de textes chiffrés C.

▫ Texte en clair connu (Known-plaintext attack)

🞄 L’attaquant dispose de textes en clair et de leur chiffre.

▫ Texte en clair choisi (Chosen-plaintext attack)

🞄 En plus, l’attaquant peut faire crypter ce qu’il veut par la méthode de chiffrement et voir ce qu’elle produit.

▫ Texte chiffré choisi (Chosen-ciphertext attack)

🞄 L’attaquant peut faire déchiffrer ce qu’il veut par la méthode de déchiffrement et voir le texte en clair

▫ Texte en clair/chiffré choisi adaptative (Adaptative- plaintext/Ciphertext attack)

🞄 C’est un cas particulier de l’attaque à texte en clair/chiffré choisi

🞄 L’attaquant peut adapter ces choix en fonction des textes en clair/ chiffré précédents



Chiffrement

* C’est le processus de transformation d’un message (appelé *texte en clair*) de telle manière à le rendre incompréhensible.
* Le résultat de ce processus de chiffrement est appelé *texte chiffré* ou *cryptogramme*
* Le processus de reconstruction du texte en clair à partir du texte chiffré est appelé *déchiffrement*.



C=Ek(M)

Dk(C)= M

Chiffrement

* Dans la cryptographie moderne, les transformations en question sont des fonctions mathématiques, appelées *algorithmes cryptographiques*, qui dépendent d'un paramètre appelé *clé*.



Chiffrement

* la plupart des procédés cryptographiques reposent essentiellement sur des substitutions et des transpositions.



Chiffrement par substitution

* Un chiffrement par substitution est un algorithme par lequel chaque caractère du message clair (écrit dans un alphabet donné) est substitué par un autre caractère dans le message chiffré (qui peut être écrit dans un alphabet différent de celui du message clair).



Chiffrement par décalage

* Dans ce système de chiffrement, chaque lettre est représentée par un entier compris entre 0 et

25. Cela revient à travailler dans *ℤ*26.

▫ Plus précisément, les clés secrètes, les messages clairs et chiffrés sont dans *ℤ*26.

* Pour chiffrer un message clair M := x1x2…xN écrit avec les lettres de l’alphabet latin (les xi sont donc des lettres quelconques)



Chiffrement par décalage

* On commence par transformer chaque lettre xi de M en un entier ni entre 0 et 25, le message devient donc la suite des entiers n1n2 …nN.
* Puis, chaque entier ni est chiffré avec le chiffrement par décalage EK avec une clé secrète K.
* On obtient donc une suite d’autres entiers C = m1m2 …mN où mi = EK (ni).



Chiffrement par décalage

Pour déchiffrer C

* Calculer Dk (mi) = DK (Ek (ni)) = ni
* Récupérer M en remplaçant chaque entier ni par sa lettre correspondante.

▫ La correspondance lettre/entier généralement employée est donnée par

a b . . . z

0 1 . . . 25



▫ Ensuite on ajoute K = 11 à chaque valeur

28 15 24 14 15 36 32 25 31 29 11 23 19 24 31 19 30

Exemple

* Supposons que la clé soit K = 11. Supposons que le texte clair soit ”rendezvousaminuit”.

▫ On commence par convertir le message en une suite d’entiers :



Définition formelle du chiffrement par décalage

* Pour une clé K ∈ ℤ26, c’est- à-dire 0 ≤ K≤ 25, M ∈ ℤ26 un message secret et C ∈ ℤ26 un message chiffré, on définit :

EK (M) = M + K (mod 26)

et

DK (C) = C - K (mod 26)

* Il est facile de vérifier que DK (EK (M)) = M pour tout M ∈ ℤ26.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| r | e | n | d | e | z | v | o |  |
| 17 | 4 | 13 | 3 | 4 | 25 | 21 | 14 |
| u | s | a | m | i | n | u | i | t |
| 20 | 18 | 0 | 12 | 8 | 13 | 20 | 8 | 19 |



Exemple

28 15 24 14 15 36 32 25 31 29 11 23 19 24 31 19 30

▫ On calcule le reste modulo 26 :

2 15 24 14 15 10 6 25 5 3 11 23 19 24 5 19 4

▫ Enfin on convertit cette suite d’entiers en une suite de lettres :

c p y o p k g z f d l x t y f t e

▫ le message chiffré est ”cpyopkgzfdlxtyfte”.



Exemple

Déchiffrement :

* il faut d’abord convertir le texte en entiers
* soustraire K = 11 à chaque valeur
* calculer les restes modulo 26
* convertir les nombres en caractères alphabétiques.



*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*

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

Exemple : Le code de César

* L’un des premiers systèmes connus est celui attribué à César.
* Il consistait simplement à utiliser la clé K = 3, il se déduisent par simple décalage de 3 positions.



* Chiffre : ?

Exemple : Le code de César

Exemple :

* Message : bonjour tout le monde

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



* Chiffre : ERQMRXUWRXWOHPRGH

Exemple : Le code de César

Exemple :

* Message : bonjour tout le monde

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



Exercice: chiffrement par décalage

* Enoncé :

▫ Pour protéger des données confidentielles, on utilise un système de chiffrement par décalge. Montrer qu’il est très aisé de déchiffrer le message suivant (écrit en français) : Zsgashwsfrwbhsfbsh.



Exercice: chiffrement par décalage

Réponse :

Le code est le suivant : décalage de 14 lettres.

Clair : 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 Chiffré :o p q r s t u v w x y z a b c d e f g h i j k l m n

* En français, la lettre la plus fréquente est le e

▫ il y a 5 s et 3 h. Il est logique de tester s = e. Le reste vient tout seul ensuite puisque le décalage est constant…

▫ Le système est donc très facilement cassable dès lors qu’on connaît les fréquences des lettres dans la langue utilisée !

* Le message écrit est : LesmetiersdInternet



Chiffrement par décalage

* Sécurité

le chiffrement par décalage est peu sûr : en effet, on peut le cryptanalyser par la méthode de recherche exhaustive (ou force brute). Comme il n’y que 26 clés possibles, il suffit d’essayer toutes les clés jusqu’ à ce que l’on obtienne un message clair compréhensible.



Substitution simple

* Substitution simple : Un caractère du message clair est substituer par un caractère unique du message chiffré. Cela correspond le plus souvent à une permutation des caractères de l'alphabet des messages clairs.
* On l’appelle aussi monoalphabétique



Combien a-t-on de clés secrètes possibles ?

Substitution simple

* Exemple : Considérons deux alphabets :

A1: 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 A2: x n y a h p o g z q w b t s f l r c v m u e k j d i

Message : bonjour Clé = A2

Message chiffré : nfsqfuc



Substitution simple : Sécurité

* Combien a-t-on de clés secrètes possibles ?

▫ Pour la lettre ”a” on a 26 choix possibles de transformation en une autre lettre de l’alphabet (”a” compris).

▫ Pour la lettre ”b” on a 25 choix possibles de transformation en une autre lettre de l’alphabet.

▫ …

▫ Pour la lettre ”z” on a 1 choix possible de transformation en une autre lettre de l’alphabet.

* Soit un total de 26 x 25 x… x 1 = 26! = 403291461126605635584000000 clés
* Une recherche exhaustive de la clé devient très difficilement réalisable, y compris avec l’assistance d’un ordinateur.

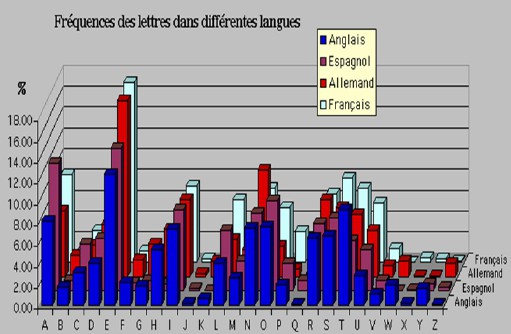


Substitution simple : Analyse de fréquence

* Exploitation des régularités du langage

▫ si la langue de départ et la technique de chiffrement sont connus

* Analyse de la fréquence d’une lettre
* Cette technique ne fonctionne bien que si le message chiffré est suffisamment long pour avoir des moyennes significatives.



Substitution simple : Sécurité

* Le chiffrement par substitution mono- alphabétique simple ne résiste pas à l'analyse des fréquences  Sécurité faible



Substitution homophonique

* substitution homophonique : fait correspondre à chaque lettre du message en clair un ensemble possible d'autres caractères



Substitution homophonique

* Il y a plusieurs techniques pour mettre en œuvre un chiffre homophone.

▫ se contenter de remplacer toutes les lettres par le même nombre de symboles.

▫ prévoir un nombre de symboles de substitution proportionnel à la fréquence de la lettre dans l'alphabet (l'idéal).

Lettre Fréquence Symboles



Exemple

|  |  |  |
| --- | --- | --- |
| A | 8,01 | 09-12-33-47-53-67-78-92 |
| B | 0,88 | 48 |
| C | 3,23 | 13-41-62 |
| D | 3,91 | 01-03-45-79 |

|  |  |  |
| --- | --- | --- |
| E | 17,52 | 14-16-24-25-31-39-44-46-55-57-64-74-82-87-  98-81 |
| F | 1,06 | 10 |
| G | 1,06 | 6 |
| H | 0,88 | 23 |
| I | 7,35 | 32-50-70-73-83-88-93 |
| J | 0,44 |  |
| K | 0,05 | 4 |
| L | 5,77 |  |
| M | 2,9 | 22-27-56 |
| N | 7,22 | 18-39-58-59-66-71-91 |
| O | 5,43 | 00-05-54-72-90 |
| P | 2,94 | 07-38-95 |
| Q | 1,14 | 94 |
| R | 6,69 | 29-35-40-42-77-80 |
| S | 8,17 | 11-19-36-43-65-76-86-96 |
| T | 7,07 | 17-20-30-49-69-75-97 |
| U | 6 | 08-52-60-61-63-99 |
| V | 1,41 | 34 |
| W | 0,02 | 89 |
| X | 0,47 | 28 |
| Y | 0,3 | 2 |
| Z | 0,12 | 21 |

15

26-37-51-84-88

* Message : bonjour tout le monde
* Chiffre : 480018157260……



Substitution homophonique

* Substitution homophonique : Un caractère du message clair correspond à plusieurs caractères du message chiffré.
* Le principe est qu'à chaque caractère de l'alphabet des messages clairs est associé une liste de lettre dans l'alphabet des messages chiffrés (qui est en général beaucoup plus gros que celui pour les messages en clair



Substitution polyalphabétique

* Il s'agit d'un ensemble de substitutions simples.
* Suivant la position du caractère dans le message clair, on applique une des substitutions simples.



Substitution polyalphabétique

Exemple : chiffrement par Vigenère

* Inventé en 1586 par le diplomate français Blaise de Vigenère.
* L’idée de ce système repose sur l’utilisation de plusieurs système monoalphabétiques



Exemple : chiffrement par Vigenère

Principe :

* Le chiffrement utilise une clé composée de lettres.
* Pour chiffrer, on prend la première lettre du message et la première lettre de la clé que l'on ajoute (les lettres ont une valeur de A=0 à Z=25, on prend la somme modulo 26), on note alors la lettre associée au résultat.
* Si la longueur de la clé est inférieure à celle du texte, alors on reprend au début de la clé.
* Plus la clé est longue, plus le code sera difficile à décrypter.



Exemple : chiffrement par Vigenère

* Texte en clair : LAVIEESTBELLE
* Clé : BONJOUR

Clé répétée : BONJOURBONJOU La première lettre vaut :

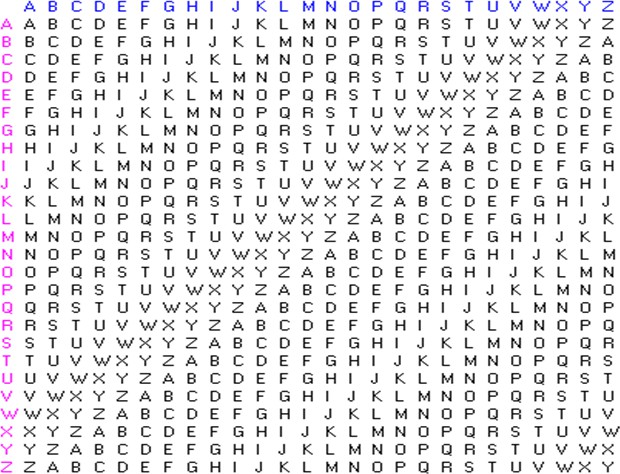
L + B = 11 + 1 = 12 mod 26 = 12 soit M.

**.**

**.**

**.**

* Texte chiffré : MOIRSYJUPRUZY



Exemple : chiffrement par Vigenère

* Pour utiliser le chiffrement de Vigenère, on a recours souvent au Carré de Vigenère.



* Réponse :

▫ Message chiffré : DHKMJCMHVMIILRPZI

Exemple : chiffrement par Vigenère

* Exercice :

▫ En utilisant le carrée de vigenère, chiffrer le texte "CHIFFRE DE VIGENERE" avec la clé "BACHELIER"



Substitution polygramique

le principe est de substituer des blocs de caractères, au lieu d'un seul caractère.



selon :

▫ C1  a\*P1 + b\*P2 (mod 26)

▫ C2  c\*P1 + d\*P2 (mod 26)

* Chaque digramme clair (P1 et P2) sera chiffré (C1 et C2)

dans Z26. Cependant, sa taille n’est pas fixée à 2. Elle

grandira selon le nombre de lettres à chiffrer simultanément.

* Les composantes de cette matrice doivent être des entiers positifs. De plus la matrice doit être inversible

Substitution polygramique : chiffrement de Hill

* Les lettres sont d’abord remplacées par leur rang dans l’alphabet. Les lettres Pk et Pk+1 deviennent Ck et Ck+1



5 7

* Pour chiffrer le message "je vous aime"

▫ Il faut d’abord remplacé les lettres par leur rang dans l’alphabet (a=1, b=2, etc.)

▫ Ensuite , on calcule

🞄 C1 = 9 \* 10 + 4 \* 5 (mod26) = 110 (mod 26) = 6

🞄 C2 = 5 \* 10 + 7 \* 5 (mod26) = 85 (mod 26) = 7

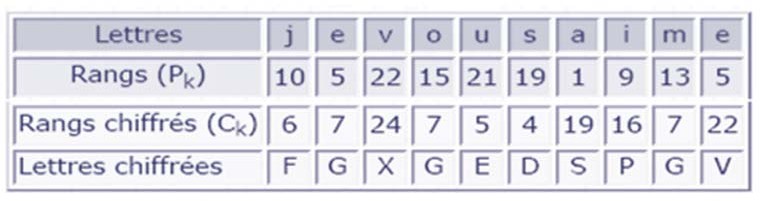
▫ On fait de même avec les 3e et 4e lettres, 5e et 6e, etc.

9 4

Substitution polygramique : chiffrement de Hill

Exemple de chiffrement :

* On prend comme clé de cryptage la matrice suivante:



Substitution polygramique : chiffrement de Hill

* On obtiendra finalement le résultat :



Chiffrement par transposition

* Le chiffrement par transposition consiste à appliquer une permutation des caractères sur le message clair en entier.

▫ le message chiffré est fait du même matériel que le message clair.

* Lors d'un chiffrement par transposition, seul l'ordre des lettres du texte en clair est modifié, les lettres elles-mêmes n'étant pas remplacées par d'autres lettres ou d'autres symboles



Chiffrement par transposition

* Combien de transpositions possibles peut-on constituer à partir d’un texte en clair de N lettres?
* Il existe N! transpositions possibles du même message.

▫ un nombre important lorsque le message est long.



chiffrement par transposition

* Pour de très brefs messages  il y a peu de variantes.

▫ un mot de trois lettres ne pourra être transposé que dans 6 (=3!) positions différentes.

🞄 Exemple : "col" ne peut se transformer qu’en "col", "clo", "ocl", "olc", "lco" et "loc".

* Lorsque le nombre de lettres croît, il devient de plus en plus difficile de retrouver le texte original sans connaître le procédé de brouillage.

▫ une phrase de 35 lettres peut être disposée de 35!

= 1040 manières différentes.



Transposition unidimensionnelle

* Consiste à permuter les caractères du texte clair en utilisant un tableau de dimension un.
* Exemple : considérons le tableau de permutation suivant:

S

A

L

A

M

L

S

M

A

A



Transposition unidimensionnelle

* Pour déchiffrer, on utilise la permutation inverse:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **3** | **1** | **5** | **2** | **4** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **2** | **4** | **1** | **5** | **3** |



Transposition unidimensionnelle

Exercice :

* Chiffrer le mots « cryptanalyse » en utilisant la permutation suivante {11, 1, 9, 4, 7, 6, 2, 10, 3, 5,

8, 12}

* Trouver la permutation inverse de décryptage.



* Le message chiffré est :

*'RNATSOISPTO NICPEHIRSS IIPELM'*

*PELM.*

*SS.II*

*PEHIR*

*O.NIC*

*OISPT*

Appliquer la transposition à chaque bloc

*MPLE.*

*IS.SI*

*IPHER*

*ION.C*

*POSIT*

Remplacer les espaces par un point.

1.

*TRANS*

2.

*RNATS*

Transposition unidimensionnelle

Exemple :

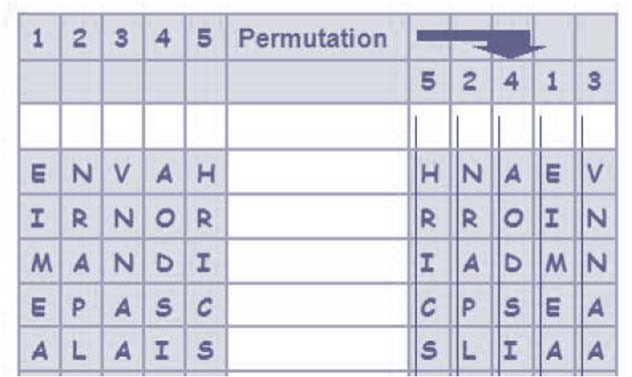
* Texte en clair : *'TRANSPOSITION CIPHER IS SIMPLE'.*
* Clé : la permutation (2, 4, 3, 1, 5)
* Découper le message en bloc de taille 5



Par exemple, écrivons "Envahir Normandie Pas Calais" dans une matrice 5x5

Transposition par colonne

* Il est possible d'utiliser des tableaux bidimensionnels pour créer des codages par transposition simple d'un message.



* Message chiffré : HRICSNRAPLAODSIEIMEAVNNAA

Transposition par colonne



Transposition par mot clé

* L'utilisation d'une clé est un moyen destiné à faciliter la mise en pratique : la clé permet de définir le nombre de colonnes ainsi que l'ordre dans lequel permuter les colonnes.
* On peut décider par exemple que l'ordre de permutation sera lié à l'ordre des lettres de la clé dans l'alphabet.



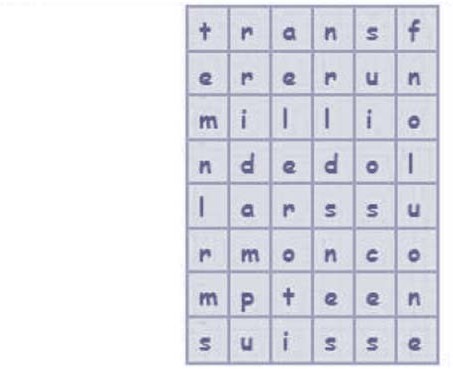
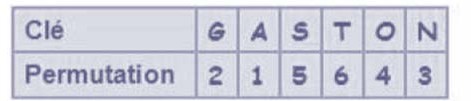
2. Donner la permutation de déchiffrement

Exercice

En utilisant la clé GASTON

1. Chiffrer le texte en clair suivant :

*TRANFERER UN MILLION DE DOLLARS SUR MON COMPTE EN SUISSE*



Transposition par mot clé

* Réponse :
  + Texte en clair *tranferer un million de dollars sur mon compte en suisse*
  + Texte Chiffré RRIDAMPUTEMNLR MSFNOLUONESUIOS CESAELEROTINRLD SNES



Principe de kerckhoffs

La sécurité repose sur le secret de la clé et non pas sur le secret de l’algorithme.

▫ Un mécanisme connu de tous sera testé, attaqué, étudié, et utilisé s’il est robuste



Taille de la clé

* Plus la taille d’une clé est grande et plus il est difficile de la trouver

▫ Les algorithmes qui utilisent des clés longues sont donc plus robustes

▫ La recherche de la clé à partir d’informations chiffrée par cette clé est plus difficile



Robustesse du système

La robustesse d’un système de chiffrement est déterminée par :

* La puissance de l’algorithme
* La taille de la clé
* Et la capacité à garder les clés secrètes de façon sécurisée



Cryptosystèmes

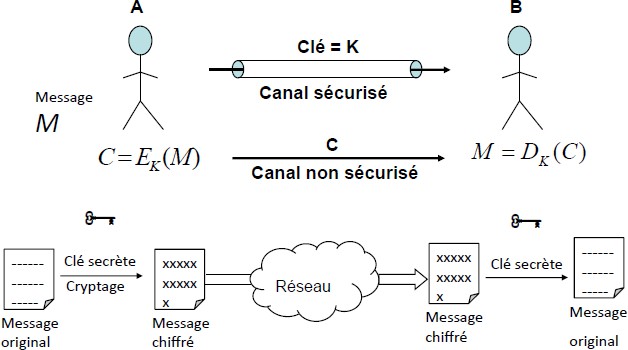
Il existe deux grandes classes de cryptosystèmes :

* Cryptosystèmes symétriques.
* Cryptosystèmes asymétriques.



Cryptosystème symétrique

* Les cryptosystèmes symétriques sont aussi appelé cryptosystèmes conventionnels ou cryptosystèmes à clé secrète.



Cryptosystème symétrique

* Schéma général du chiffrement symétriques

Déchiffrement

Chiffrement



One Time Pad

* Le masque jetable, également appelé chiffre de Vernam, est un algorithme de cryptographie à clé symétrique inventé par Gilbert Vernam en 1917.
* Principe :

▫ Chiffrement

C = MK.

▫ choisir une clé formée d’une suite de caractères aléatoires

▫ la clé a la même taille que le texte à chiffrer

▫ il ne faut utiliser la clé qu’une seule fois

▫ Déchiffrement

M= CK.



One Time Pad

Exemple :

* Chiffrement

M = 1 0 0 1 0 0 1 0 0

K = 1 0 1 0 1 0 1 1 1

C = 0 0 1 1 1 0 0 1 1

* Déchiffrement

C = 0 0 1 1 1 0 0 1 1

K = 1 0 1 0 1 0 1 1 1

M=DK(C) = 1 0 0 1 0 0 1 0 0



One Time Pad

Exercice : En utilisant OTP, Déchiffrer le texte chiffré suivant

cuskqxwmfwituk

* En employant le masque jetable suivant : bgfbcdfbfdecdg

▫ Résultat : BONJOURLATERRE

* En employant ce nouveau masque jetable : quauwtedbdisjg

▫ Résultat : MASQUESJETABLE

Il est donc impossible de déterminer le bon masque !



One Time Pad

* Algorithme de chiffrement parfait

▫ quel que soit M', il existe K' tel que C = M'K', c'est K' = M'C : tous les messages en clair sont équiprobables.

▫ sans information sur K, le cryptanalyste n'a donc aucun moyen de savoir quel est le bon texte en clair.

▫ ce système est inutilisable dans le cadre du chiffrement d'un flux important de données et reste limité à des applications extrêmes



Classes de chiffrements symétriques

Deux classes de chiffrements symétriques :

* Par flot :

▫ Traitement à la volée ; chiffrement à la one-time pad

▫ Substitution rapide ( typiquement )

▫ Générateur pseudo aléatoire

▫ Kerckhoffs : la sécurité repose sur le générateur de clé!

▫ Exemples : LFSR, RC4 (Rivest)

* Par bloc :

▫ Segmentation du message M à chiffrer

▫ M est scindé en un nombre de bloc de taille fixe

▫ Cryptage des blocs

▫ Le message chiffré est obtenu en concaténant les cryptogrammes des bloc

▫ Exemples : DES, IDEA, AES



Modes de chiffrement

* Objectifs des modes de chiffrement :

▫ Ils ne concernent que le chiffrement par bloc.

▫ Ils doivent masquer les blocs clairs identiques.

▫ Deux blocs identiques chiffrés avec la même clé ne donnent pas les mêmes chiffres si le reste du message diffère



Modes de chiffrement

* Les principaux modes de chiffrement :

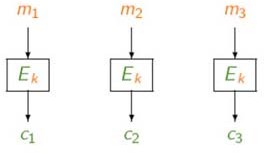
▫ ECB (Electronic CodeBook)

▫ CBC (Cipher Block Chaining)

▫ CFB (Cipher FeedBack)

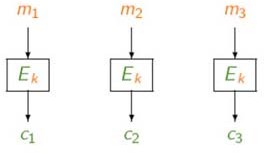
▫ OFB (Output FeedBack)

▫ CTR (CounTeR mode)



ECB (Electronic Code Book)

* Un bloc de texte se chiffre indépendamment de tout en un bloc de texte chiffré
* Chiffrement : chaque bloc clair mi est chiffré indépendamment et donne un bloc chiffré ci=Ek(mi)

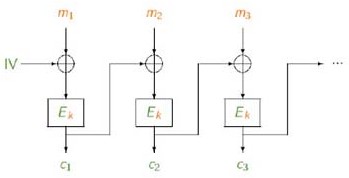


ECB (Electronic Code Book)

* Déchiffrement : chaque chiffre est déchiffré indépendamment pour donner le clair correspondant mi = Dk(ci).
* Conséquence : deux blocs clairs identiques

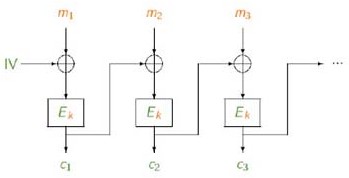
donnent toujours le même bloc chiffré pour une clé k fixée.

* Aucune sécurité



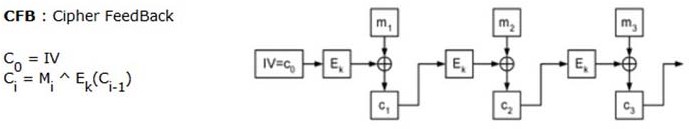
CBC (Cipher Bloc Chaining)

* Chaque bloc du cryptogramme dépend du bloc de texte en clair et de tous les blocs précédents
* Chiffrement : un vecteur d'initialisation IV est généré aléatoirement. ci = Ek(mi  ci-1).
* Le vecteur IV est transmis avec les blocs chiffrés.



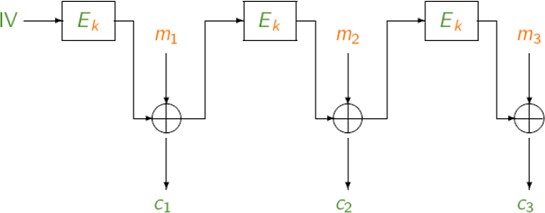
CBC (Cipher Bloc Chaining)

* Déchiffrement : mi = Dk(ci)  ci-1
* Conséquence : deux blocs clairs identiques chiffrés différemment.
* Mode le plus utilise



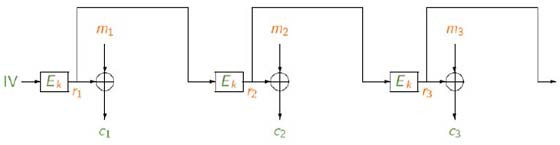
CFB (Cipher FeedBack)

* Chiffrement : un vecteur d'initialisation IV est généré aléatoirement.
* ci = ri  mi ; où r1= Ek(IV ) et pour i ≥2; ri = Ek(mi-1  ri-1).
* Le vecteur IV est transmis avec les blocs chiffrés.



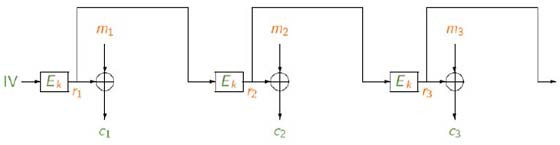
CFB (Cipher FeedBack)

* Déchiffrement : mi= ci  ri= ci  Ek(ci-1)
* Conséquence : deux blocs clairs identiques chiffrés différemment.
* Moins sûr, parfois plus rapide



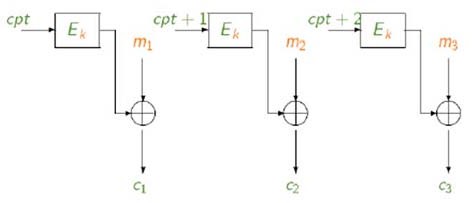
OFB (Output FeedBack)

* Chiffrement : un vecteur d'initialisation IV est généré aléatoirement.
* ci= ri  mi , où r0= IV et pour i ≥1; ri = Ek(ri-1).
* Le vecteur IV est transmis avec les blocs chiffrés.



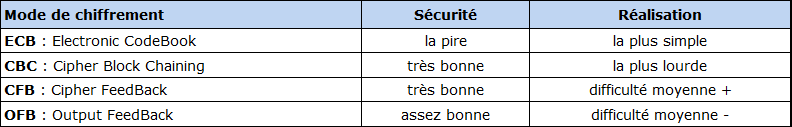
OFB (Output FeedBack)

* Déchiffrement : mi = ci  ri
* Conséquence : deux blocs clairs identiques chiffrés différemment.
* Totalement symétrique
* Moins de câblage



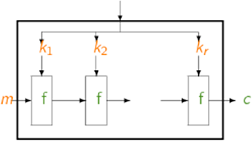
CTR (CounTeR mode)

* Chiffrement : ci= ri  mi , où r1= Ek(cpt) et pour i≥2; ri = Ek(cpt + i-1):
* Déchiffrement : mi = ci  ri
* Conséquence : deux blocs clairs identiques chiffrés différemment.



Modes de chiffrement

* Comparaison :



* Le bloc clair m est transforme r fois successivement à l'aide d'une fonction f qui dépend d'une sous-clé ki
* Le chiffre c est le résultat de la dernière transformation.

c = f (…f (f (m, k1), k2), …), kr)

* r est appelé nombre de tours ou de rondes.

Chiffrement itératif

* Tous les systèmes de chiffrement par blocs actuels suivent le schéma suivant :



DES

En 1977, le Bureau Fédéral des Standards des États-Unis (US Fédéral Bureau of Standards) a approuvé un algorithme cryptographique conçu par la firme IBM pour la sécurité des données. Le Standard de chiffrement de Données (DES, Data Encryption Standard)



DES

* Principe

🞄 DES est basé sur une ensemble de transformation, transposition, substitution et opérations non- linéaire.

🞄 Chiffrement de blocs de 64 bits grâce à une clé de 64 bits (56 bits effectifs).

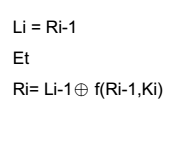
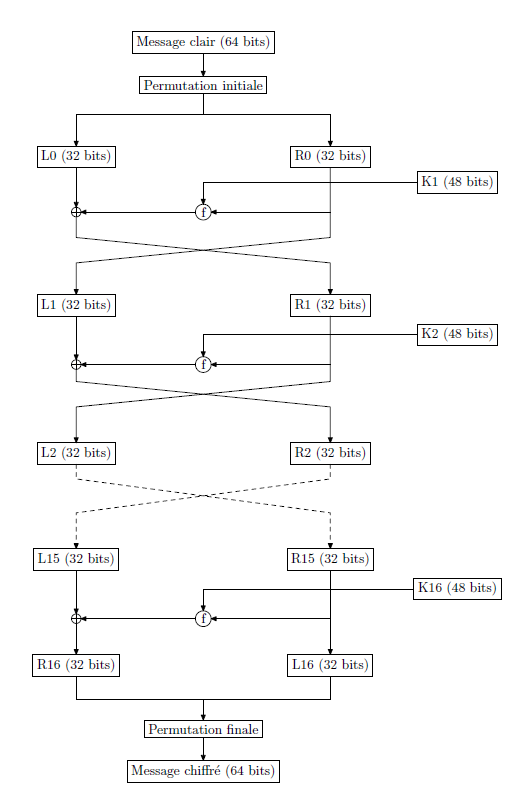
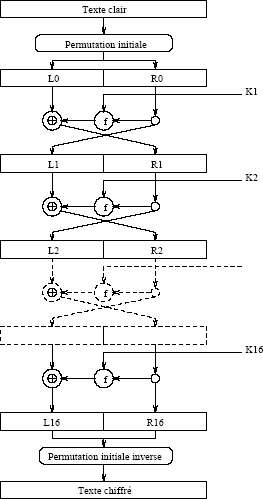
🞄 Processus de chiffrement : Permutation PI et PI-1, 16 rondes utilisant 16 clés secondaire K1 à K16 de 48 bits issues à partir de la clé K.



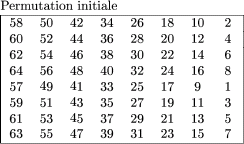
DES

Les grandes lignes de l'algorithme sont les suivantes :

* Fractionnement du texte en blocs de 64 bits (8 octets)
* Permutation initiale des blocs
* Découpage des blocs en deux parties: gauche et droite, nommées *G* et *D (*la notation anglo-saxonne étant *L* et *R* pour *Left and Right)*
* Étapes de permutation et de substitution répétées 16 fois (appelées rondes)
* Recollement des parties gauche et droite puis permutation initiale inverse.

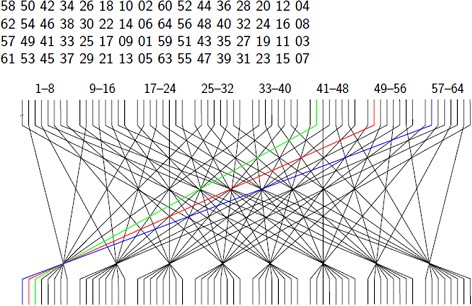


DES



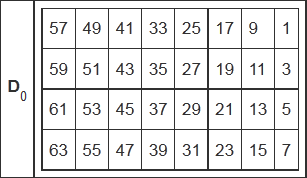
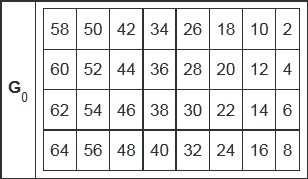
DES

* Étape 1 : les 64 bits du bloc sont soumis à la permutation initiale PI.



DES

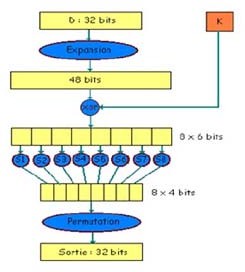
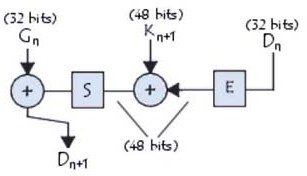
* Permutation initiale PI



DES

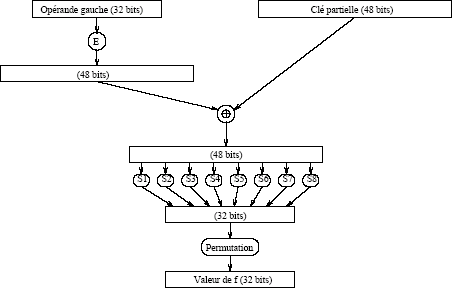
* Étape 2 : le bloc de 64 bits est scindé en deux blocs de 32 bits, notés respectivement G et D (pour gauche et droite).

On note G0 et D0 l'état initial de ces deux blocs

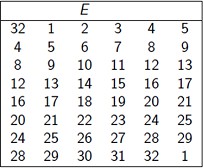


DES

* Les blocs Gn et Dn sont soumis à un ensemble de transformation itératives appelées *rondes*, explicitées dans ce schéma :



DES



DES

* Étape 3 : Les 32 bits du bloc D0 sont étendus à 48 bits grâce à une table (matrice) appelé *table d'expansion* (notée E), dans laquelle les 48 bits sont mélangés et 16 d'entre eux sont dupliqués :



DES

* Table d'expansion

▫ Ainsi, le dernier bit de D0 (c'est-a-dire le 7ème bit du bloc d'origine) devient le premier, le premier devient le second, ...

▫ De plus, les bits 1, 4, 5, 8, 9, 12, 13, 16, 17, 20, 21,

24, 25, 28 et 29 de D0 (respectivement 57, 33, 25,

1, 59, 35, 27, 3, 61, 37, 29, 5, 63, 39, 31 et 7 du bloc

d'origine) sont dupliques et disséminés dans la matrice.

* La matrice résultante de 48 bits est appelée D'0=E[D0]



DES

* Étape 4: on procède ensuite à un *OU exclusif*

entre la première clé K1 et D’0( c.à.d. E[D0]).

▫ Le résultat de ce *OU exclusif* est une matrice de 48 bits que nous appellerons D0 par commodité (il ne s'agit pas du D0 de départ).



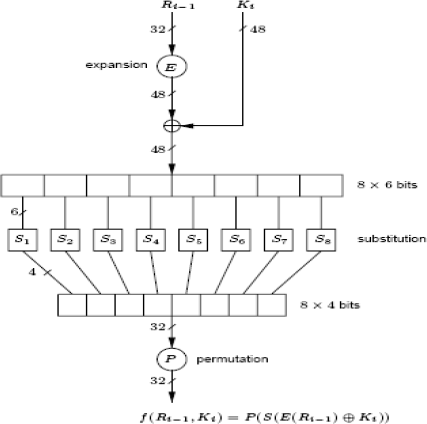
DES

Étape 5:

* D0 est ensuite divisé en 8 blocs de 6 bits chacun: D01, D02, D03,…D08.
* Pour un bloc quelconque D0i de 6 bits, la sortie d’une fonction de sélection Si est un bloc de 4 bits.

▫ fonctions de sélection Si : appelées parfois *boîtes de substitution* ou *fonctions de compression*

* Les résultats sont combinés pour former un D0 de 32 bits



DES



**Quatre bits de sortie**

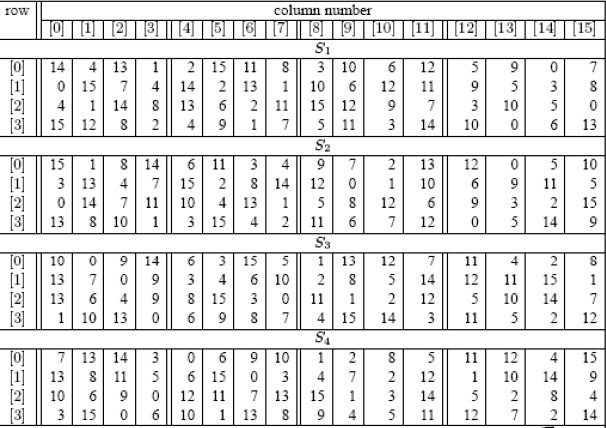
b2b3b4b5 détermine le numéro de la colonne du tableau

**S\_box**

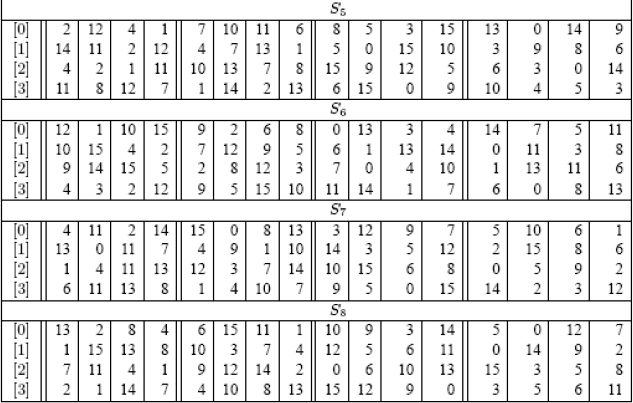
b1 et b6 détermine le numéro de la ligne du tableau

**B = b1 b2 b3 b4 b5 b6**

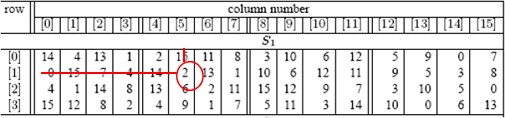
DES



DES



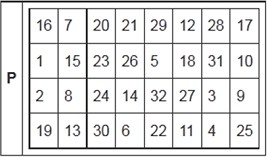
DES



DES

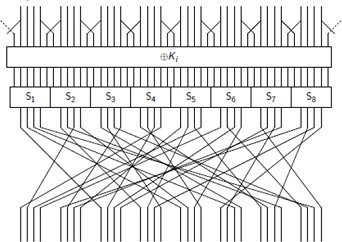
Exemple :

* Si D01 = 001011, alors le premier et le sixième bit (01=1) déterminent la ligne dans S1 et les bits de 2 à 5 ( 0101 = 5) déterminent la colonne.
* S1 (001011) = (2 = 0010)



DES

* Étape 6 : Le bloc de 32 bits obtenu est enfin soumis à une permutation P



DES

* la fonction f est une fonction de 32 bits vers 32 bits



DES

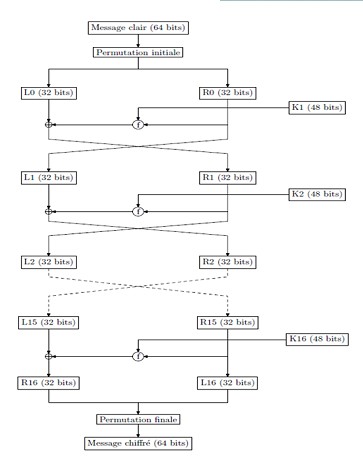
Ces choix peuvent paraître arbitraires mais :

* Toutes les briques sont très simples à coder et efficaces en hardware
* Les boîtes S apportent la non-linéarité
* Expansion et permutation garantissent une diffusion rapide

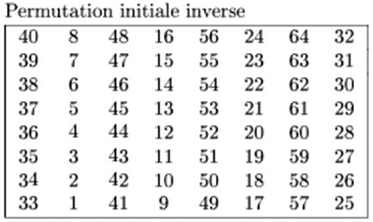


DES

* Le second cycle est simplement une répétition du premier avec D1 et G1 comme blocs d’entrée et K2 comme clé.
* L’opération se poursuit sur tous les cycles, mais à la dernière itération, les moitiés de gauche et droite ne sont pas échangées.

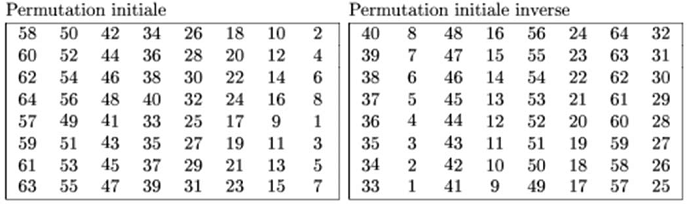


DES



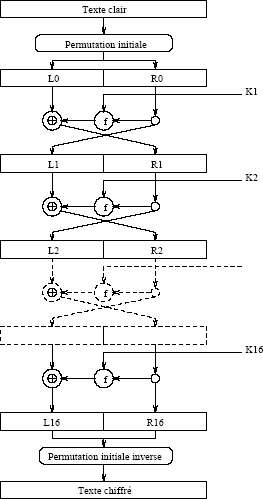
DES

* À la fin des itérations, les deux blocs G16et D16 sont recollés, puis soumis à la permutation initiale inverse PI-1 pour en donner un texte codé de 64 bits

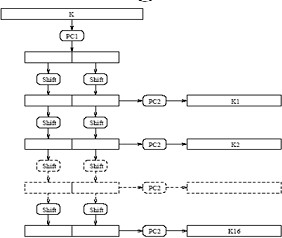


DES

* Permutation



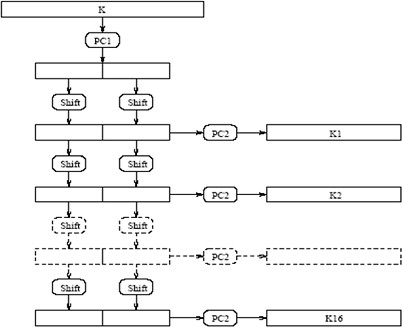
DES



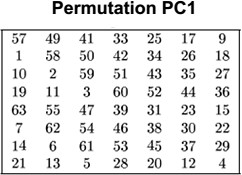
DES : Génération des clés

Génération des clés :

* Le schéma suivant montre comment obtenir à partir d'une clé de 64 bits, 8 clés diversifiées de 48 bits chacune servant dans l'algorithme du DES



DES : Génération des clés



DES : Génération des clés

* La première étape consiste en une permutation notée PC1
* les bits de parité de la clé sont éliminés afin d'obtenir une clé d'une longueur utile de 56 bits.

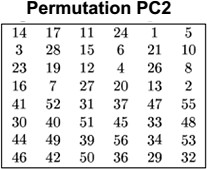


DES : Génération des clés

* La clé K est ensuite divisée en deux parties notées par L (à gauche) et R (à droite).
* L0 et R0 indiquent les états initiaux de L et R
* Ces deux blocs subissent un « Left Shift » c'est à- dire un décalage des bits à gauche

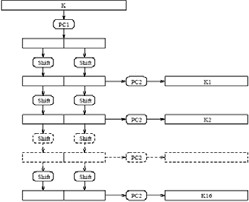
▫ les bits en seconde position prennent la première position, ceux en troisième position la seconde, ...

▫ Les bits en première position passent en dernière position.



DES : Génération des clés

* Les 2 blocs de 28 bits sont ensuite regroupes en un bloc de 56 bits.
* Celui-ci passe par une permutation, notée PC2, fournissant en sortie un bloc de 48 bits, représentant la clé K1.



DES : Génération des clés

* Au deuxième cycle, K2 est générée en décalant L1 et R1 d’un cran vers la gauche et en formant le bloc de 56 bits L2R2 à partir de L2 et R2.
* L2R2 entre alors dans PC2 pour produire la clé K2 de 48 bits et ainsi de suite la génération de K16



DES

Sécurité :

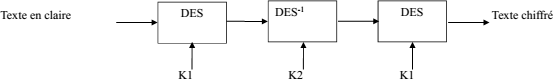
* Le chiffre le plus controversé et le plus attaqué.
* clé 56 bits trop courte.
* Le 17 juin 1997, le DES est cassé en trois semaines par une fédération de petites machines sur Internet.
* Maintenant, on estime (selon un rapport présenté au Sénat Américain) le temps nécessaire, à un Etat, pour décrypter un message chiffré avec le DES à seulement quelques secondes.



3DES

3DES, une alternative au DES

* Une solution à court terme consiste à chaîner trois chiffrement DES à l'aide de deux clés de 56 bits. Ce procède est appelé 3DES ou Triple DES (note TDES)



3DES

* 3DES offre un niveau acceptable de sécurité mais son application demande beaucoup de ressources système.
* C = Ek1(Dk2(Ek1(M))) et M = Dk1(Ek2(Dk1(C)))



3DES

* Le 3DES permet d'augmenter significativement la sécurité du DES, toutefois il a l'inconvénient majeur de demander également plus de ressources pour les chiffrement et le déchiffrement.
* On distingue habituellement plusieurs types de chiffrement triple DES :

▫ DES-EEE3 : 3 chiffrements DES avec 3 clés différentes

▫ DES-EDE3 : une clé différente pour chacune des 3 opérations DES (chiffrement, déchiffrement, chiffrement) ;

▫ DES-EEE2 et DES-EDE2 : une clé différente pour la seconde opération (déchiffrement).



AES (Advanced Encryption System)

Remplaçant du DES

* Appel d'offres (1997 ) NIST (National Institute of Standards and Technologies) pour élaborer l'AES
* Processus de sélection (1997-2000) :

▫ 15 candidats initiaux

▫ 5 retenus pour le second tour

🞄 Rijndael (Daemen-Rijmen, Belgique)

🞄 MARS (IBM, USA)

🞄 SERPENT (Biham-Knudsen-Anderson)

🞄 RC6 (RSA Labs)

🞄 Twofish (USA)

* Le 2 octobre 2000, le NIST donne sa réponse : c'est le Rijndael qui est choisi, un algorithme mis au point par deux belges, Joan Daemen et Vincent Rijmen



AES (Advanced Encryption System)

* Principe

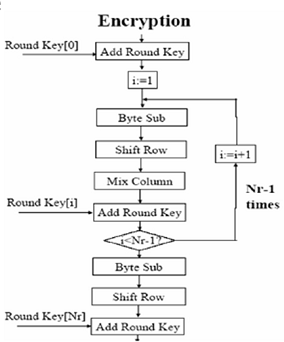
▫ Rijndael :

🞄 blocs de longueur variable 128, 192 ou 256 bits

🞄 clé : 128, 192 ou 256 bits

▫ Rijndael version ‘AES’ est restreint à une longueur de bloc fixée à 128 bits.

▫ 10, 12, 14 rondes suivant la clé 128, 192, 256 .



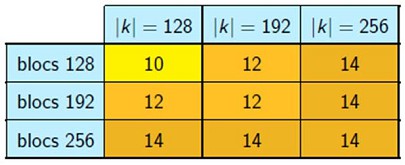
AES (Advanced Encryption System)

* Principe : Structure générale

▫ Ou exclusif avec la clé

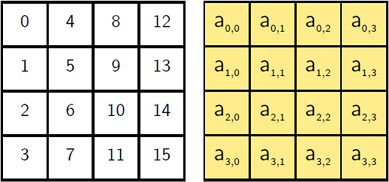
▫ Réalisation de Nr -1 rondes identiques (Nr : nombre de rondes)

▫ Ronde finale un peu différente



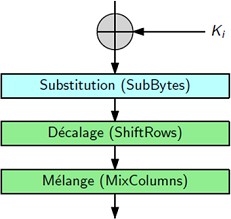
AES

* Le nombre de tours varie suivant la taille des blocs et la taille des clés



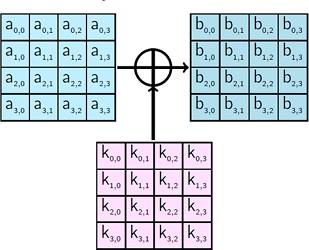
AES

* Les blocs de messages (128 bits) sont découpés en 16 octets placés dans une matrice de 4 x 4 = 16 cases



AES

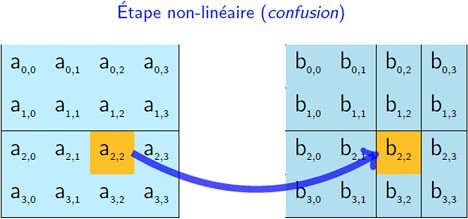
* Cette matrice subit ensuite 4 transformations par tour
  + AddRoundKey# Ou exclusif avec la sous clé
  + SubByte# Substitution octet par octet
  + ShiftRow# Décalage des lignes
  + MixColumn# Mélange sur les colonnes (sauf pour la dernière ronde)



* Remarque : C'est la seule partie du protocole qui utilise la clé.

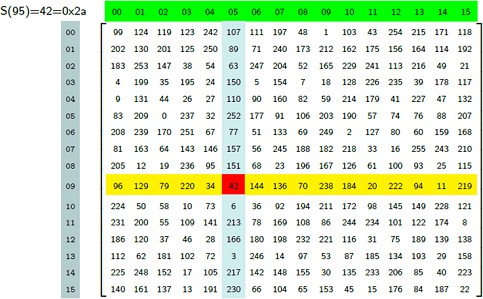
AES

* Étape AddRoundKey



AES

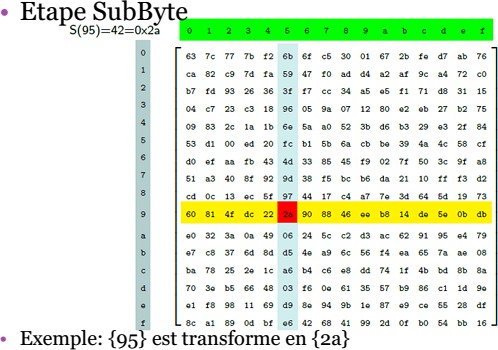
* Étape SubByte



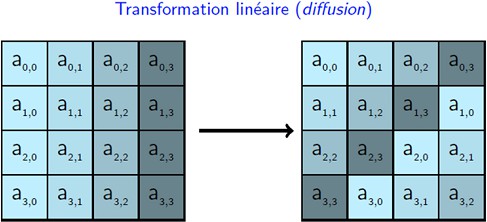
AES

* Etape SubByte

La permutation de SubBytes est donnée par la S-boîte suivante :



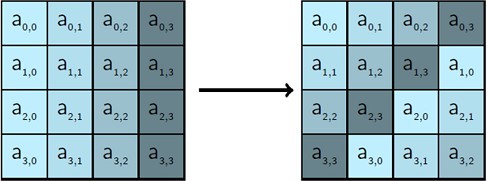
AES



AES

* Étape ShiftRows

Transformation : La première ligne est reste inchangée, la deuxième ligne est décalée d'une case vers la gauche, la troisième de deux cases vers la gauche et la quatrième de trois cases vers la gauche



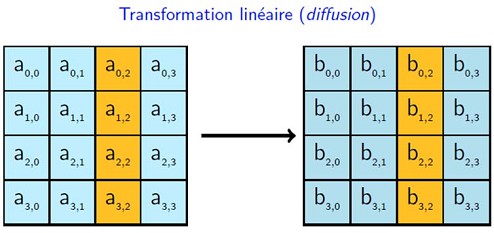
AES

* Exemple :

2A 64 D5 CA E4 4C AA ED 1F 35 5A 37 94 4E F0 84

devient

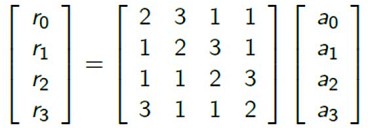
2A 4C 5A 84 E4 35 F0 CA1F 4E D5 ED 94 64 AA 37



* Remplacée dans le dernier tour par addRoundKey

AES

* Étape MixColumns



AES

* Étape MixColumns

▫ Pour Mixcolumns, on a:



AES : Génération des clés

* Les sous-clés de 128 bits, numérotées de 0 à 10, sont dérivées de la clé secrète de la manière suivante :

▫ la sous-clé numéro 0 correspond à la clé secrète ;

▫ ensuite, la sous-clé numéro i (utilisée à la ième itération) est obtenue à partir de la sous-clé numéro (i−1)

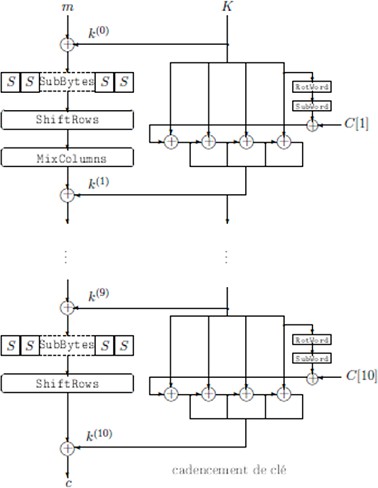
▫ On permute de manière circulaire, par la fonction RotWord, les quatre derniers octets de la clé numéro (i − 1), puis on leur applique la fonction SubWord composée de 4 permutations S.



AES : Génération des clés

▫ Après avoir ajouté une constante (dépendant de i) au premier octet (les trois autres octets de la constante C [i] sont nuls), on effectue une addition bit-à-bit entre les quatre octets ainsi obtenus et les quatre premiers octets de la sous-clé précédente.

▫ Les trois autres blocs de quatre octets de la clé numéro i sont ensuite simplement le résultat d'un OU EXCLUSIF entre le bloc correspondant de la sous-clé (i−1) et le bloc précédent de la sous-clé i.



AES



AES

Performance :

* La comparaison la plus pertinente est celle entre AES128 et TDES:

▫ le triple DES est 3 fois plus lent que DES

▫ l’AES est 2,7 fois plus rapide que TDES

▫ On a donc un gain de sécurité sans perte de performances



AES

Sécurité :

* L’AES a été conçu pour résister au mieux la cryptanalyse linéaire et différentielle
* La meilleure attaque reste à l’heure actuelle la recherche exhaustive de la clé



AES

* Il est impossible d’effectuer une recherche exhaustive sur 128 bits

▫ Comparaison DES/AES: en imaginant une machine cassant un chiffrement DES par seconde, il faudrait 149 mille milliards d’années pour retrouver une clé de 128 bits

▫ Pour casser un tel chiffrement, il faut trouver une autre méthode que la recherche exhaustive.



AES

* Résultats de cryptanalyse contre A.E.S.

▫ Depuis juin 2003, le gouvernement américain a annonce que AES peut être utilise pour crypter les informations classifiées SECRET:

« La conception et la résistance des clés A.E.S. de différentes tailles (128, 192 ou 256 bits) sont suffisantes pour protéger les informations classées au niveau SECRET. Les informations classées au niveau TOP SECRET doivent utiliser des clés de longueur 192 ou 256 bits »



AES

Résultats de cryptanalyse contre AES

▫ Recherche exhaustive : On essaie toutes les clés possibles.

🞄 Pour les plus courtes, il faut en tester 2128, soit environ 700 000 000 000 fois l'âge de l'univers pour

un test de 1 000 000 000 clés par seconde.

🞄 Pour les plus longues, il faut en tester 2256, soit environ 2.1050 fois l'âge de l'univers pour un test de 1 000 000 000 clés par seconde.

▫ Autres attaques : Toutes les attaques efficaces contre AES. sont des attaques contre de mauvaises implémentations logicielles ou matérielles.



Echange de clés

* Protocole de Diffie-Hellman

▫ Alice et Bob se mettent d'accord sur un grand entier *n* tel que *(n-1)/2* soit premier et sur un entier *g* primitif par rapport à *n*. Ces deux entiers sont publics.

▫ Alice choisit *a*, puis envoie à Bob *A=ga mod n.*

▫ Bob choisit *b*, puis envoie à Alice : *B=gb mod n*.

▫ Alice calcule *KAB=Ba mod n =gba mod n*.

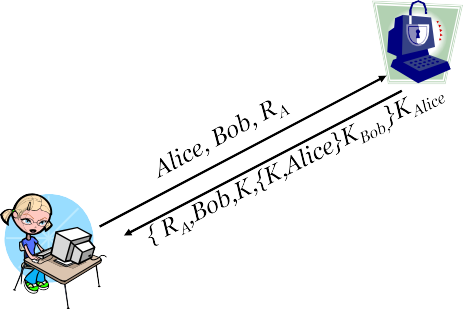
▫ Bob calcule *KBA=Ab mod n =gab mod n*.

▫ *KAB=KBA=gab mod n* est le secret partagé par Alice et Bob.



Distribution de clés

* Needham et Schroeder ont proposé un protocole de distribution de clés pour les cryptosystèmes symétriques.
* Leur protocole suppose un centre de distribution de clés KDC (Key distribution Center) responsable de générer et de distribuer toutes les clés de communication, appelées clés de sessions.
* Il suppose aussi que toute entité X du réseau partage une clé secrète Kx avec le KDC.
* Le KDC utilise cette clé pour protéger les messages qu’il envoie à cette entité.



Bob

Alice

*{K , Alice} K Bob*

*{RB} K*

*{RB - 1}K*

Authentification avec un tiers

* Needham Schroeder

KDC



Cryptosystème symétriques

* Caractéristiques de chiffrement à clé secrète

▫ Avantages

🞄 Rapide

🞄 Facile à mettre en œuvre sur une puce électronique.

▫ Inconvénients

🞄 Distribution des clés difficile.

🞄 Nécessite N(N-1)/2 clés (pour N correspondants).



Cryptosystème asymétrique

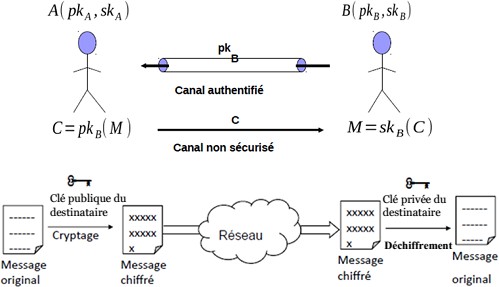
* En 1976, Diffie et Hellman introduisent le concept de cryptographie à clé publique

▫ Deux clés, une publique et une privée

▫ La clé publique peut être révélée sans créer de risques

▫ La clé privée ne doit être accessible que par son propriétaire

▫ Il n'est pas possible (infaisable) de dériver une des deux clés à partir de l'autre



Cryptosystème asymétrique

* Schéma général du chiffrement asymétriques



Arithmétique pour RSA

* Pour un entier *n*, sachant qu’il est le produit de deux nombres premiers, il est difficile de retrouver les facteurs *p* et *q* tels que n =pq
* Le principe du chiffrement RSA, chiffrement à clé publique, repose sur cette difficulté.



Arithmétique pour RSA

* Congruences

▫ On dit que deux entiers *a* et *b* sont congrus modulo un entier n strictement positif si *a-b* est un multiple de n. On écrit : *a ≡ b* mod (n)



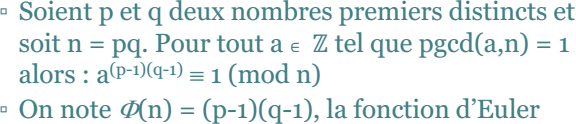
Arithmétique pour RSA

* Petit théorème de Fermat



Arithmétique pour RSA

* Corollaire



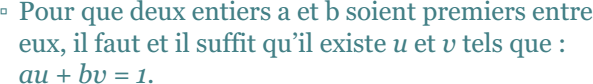
Arithmétique pour RSA

* Petit théorème de Fermat amélioré



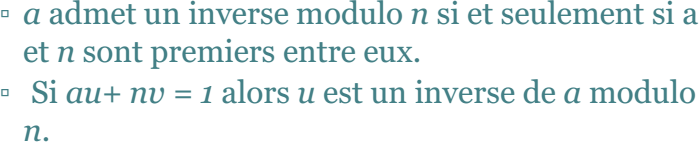
Arithmétique pour RSA

* Inverse modulo n



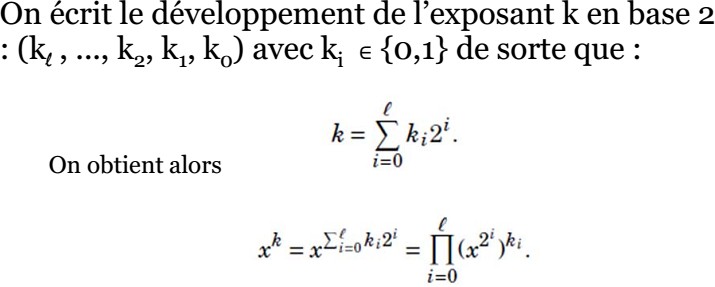
Arithmétique pour RSA

* Théorème de Bézout



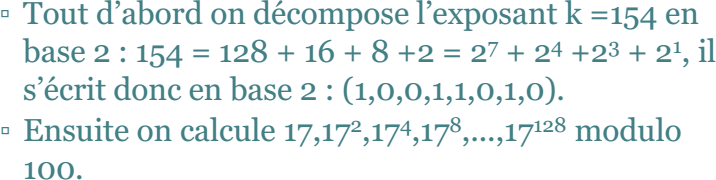
Arithmétique pour RSA

* Proposition



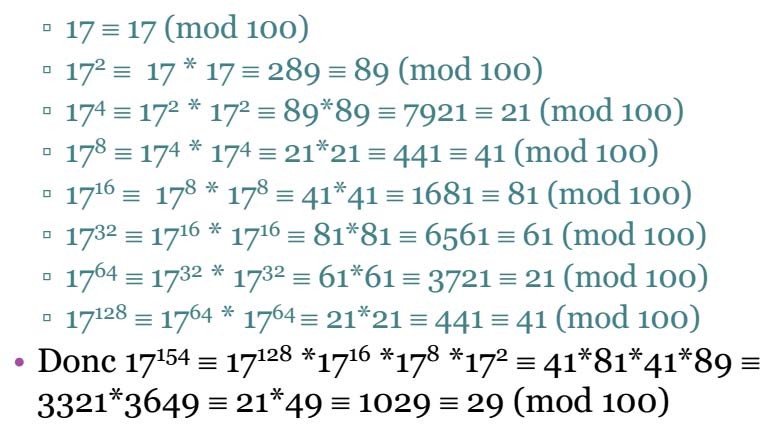
Arithmétique pour RSA

* Exponentiation rapide



Arithmétique pour RSA

* Exemple : calculons 17154 (mod 100).



Arithmétique pour RSA

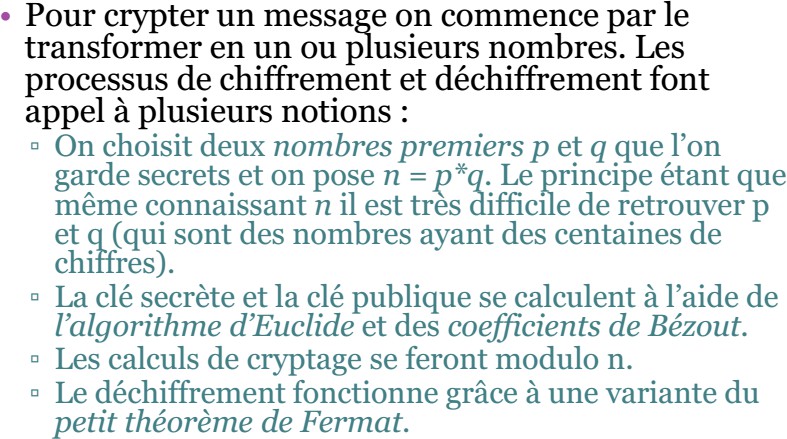


RSA

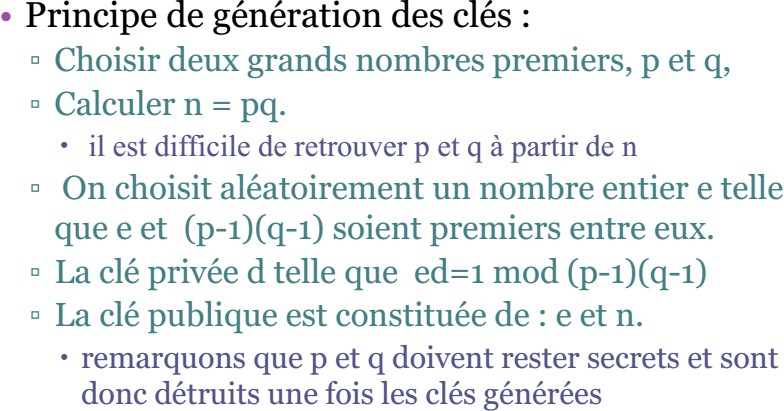
* Développé en 1977 par Ron Rivest, Adi Shamir et Len Adleman
* Basé sur la difficulté de factorisation des grands nombres.
* Procédures de Chiffrement / Déchiffrement à 2 clés :

▫ Une clé publique révélée à tous les correspondants

▫ Une clé privée, garder secrète par son détenteur.



RSA



RSA



RSA

* Chiffrement

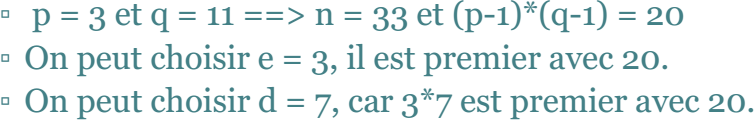
▫ Le message en clair : M < n

▫ Le message chiffré C est obtenu en calculant C=Me mod n

* Déchiffrement

▫ Le message chiffré : C

▫ Le message en clair M sera obtenu en calculant Cdmod n=Med mod n = M



Exemple d’utilisation de RSA

Considérons le cas suivant :

* choix des clés



Exemple d’utilisation de RSA

* Chiffrement : Par exemple, on veut envoyer un message clair « HI» (on découpe le message par exemple : H=>8 et I =>9. Remarquez

que ces valeurs sont inférieures à n=33, sinon l'algorithme RSA n'est pas applicable).

▫ 83mod 33= 17

▫ 93mod 33=3

▫ On transmet le message chiffré : 17, 3



Exemple d’utilisation de RSA

* Déchiffrement :

▫ On reçoit la valeur : 17, 3

▫ 177= 8 mod 33 =>H

▫ 37 = 9 mod 33 => I



RSA

* Performances :

▫ Problème principal, lenteur de la méthode

* Sécurité :

▫ Le chiffre est considéré comme sûr si l’on respecte les contraintes diverses d’utilisation.

🞄 Tant qu’on ne trouve pas de solution efficace au problème de factorisation.



Cryptosystème asymétriques

Caractéristiques de chiffrement à clé publique

* Avantages :

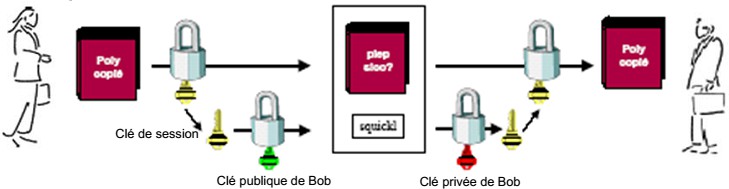
▫ Nécessite 2N clés (pour N correspondants).

▫ Signature numérique

▫ Distribution des clés relativement rapide

* Inconvénients

▫ Lent



Chiffrement hybride

* Pour profiter de l’efficacité de chiffrement à clé secrète et des avantages de chiffrement à clé publique

▫ Algorithme à clé publique pour chiffrer une clé de session secrète

▫ Algorithme à clé secrète pour chiffre le message avec la clé de session secrète



Fonction de hachage à sens unique

* Une fonction à sens unique est une fonction facile à calculer mais difficile à inverser.
* Une fonction de hachage est une fonction qui convertit une chaîne de longueur quelconque en une chaîne de taille inférieure et généralement fixe

▫ la chaîne résultante est appelée empreinte (digest en anglais) ou condensé de la chaîne initiale.



Fonction de hachage à sens unique

* Une fonction de hachage à sens unique = fonction de hachage + fonction à sens unique



Fonction de hachage à sens unique

* Propriétés d’une fonction de hachage à sens unique :

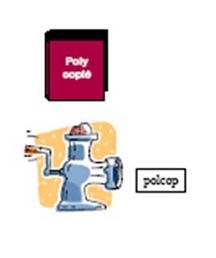
▫ Opère sur des messages M de taille quelconque

▫ Génère une empreinte h(M) de taille fixe

▫ Étant donne M, il est facile de calculer h(M)

▫ Étant donné h(M), il est pratiquement impossible de retrouver M

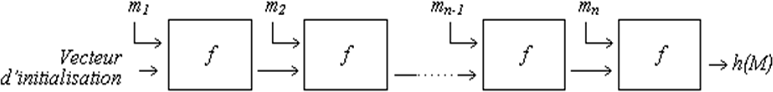
▫ Pour tout message M, il est impossible de trouver un autre message M’ tel que h(M) = h(M’) (résistance aux collisions)



Fonction de hachage à sens unique

* Rôle principal d’une fonction de hachage à sens unique :

▫ Fournir l’empreinte d’un fichier, d’un message ou d’un bloc de données.



Construction de MERKLE-DAMGARD

* Fonctionnement

▫ le message M est décomposé en n blocs m1 ,...,mn

▫ une fonction de compression f est appliquée à chaque bloc et au résultat de la compression du bloc précédent

▫ l'empreinte h(M) est le résultat de la dernière compression



Fonction de hachage à sens unique

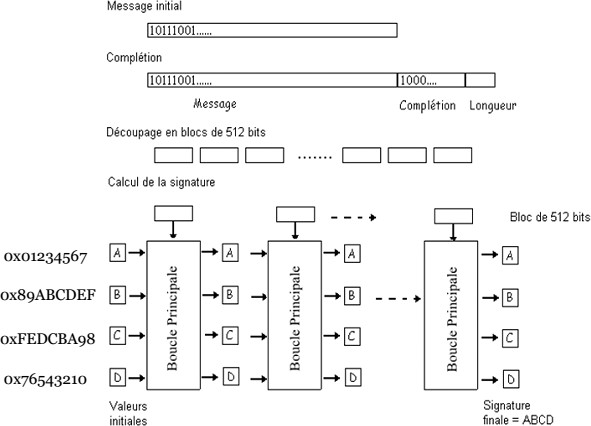
* MD5 (Message Digest)

▫ Développé par Ron Rivest

▫ MD5 est une version améliorer de MD4

▫ Produit une empreinte de 128 bits.

▫ Le message d’entrée est décomposé en blocs de 512 bits.



Fonction de hachage à sens unique

* MD5 (Message Digest)



Fonction de hachage à sens unique

* MD5 (Message Digest)

▫ Sécurité

🞄 En 1996, une attaque réussie: H. Dobbertin a montré qu’on pouvait obtenir des collisions sur MD5 en quelques heures de calcul.

▫ Depuis MD5 est considéré comme inutilisable pour des applications de haute sécurité.



Fonction de hachage à sens unique

* RIPE-MD

▫ Développée dans le cadre du projet RIPE (RACE Integrity Primitives Evaluation) de la communauté Européenne,

▫ Fournit une empreinte de 128 bits.

▫ RIPE-MD-160 est une version renforcée de RIPEMD qui fournit une empreinte de 160 bits.



Fonction de hachage à sens unique

* SHA 1 (Secure Hash Algorithm)

▫ SHA est une fonction de hachage conçue par le NIST (National Security Agency)

▫ Version SHA-0 : corrigée assez vite en SHA-1.

▫ produit une empreinte de 160 bits

▫ travaille sur des blocs de 512 bits.

▫ Il nécessite plus de ressources que MD5.



Fonction de hachage à sens unique

* SHA 2 (Secure Hash Algorithm)

▫ Extension de SHA1 proposée par le NIST en 2001

▫ produit des empreintes de 224, 256, 384 et 512 bits.



Fonction de hachage à sens unique

* Keccak

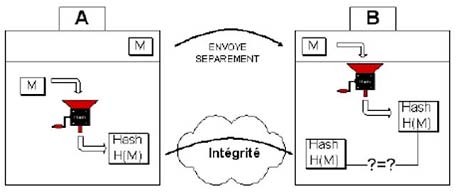
▫ Fonction développée par G. Bertoni, J. Daemen,

M. Peeters et G. Van Assche.

▫ Futur standard SHA3 choisi par le NIST (oct. 2012)

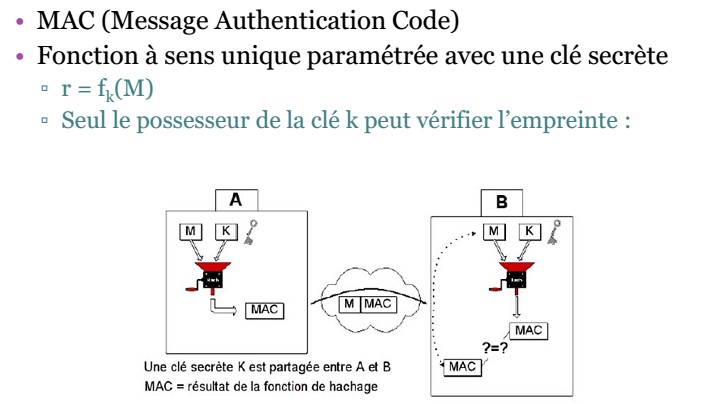
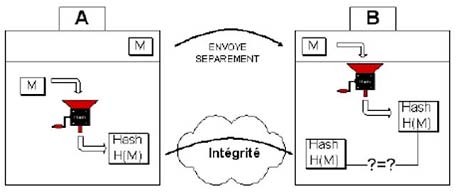
▫ produit des empreintes de tailles variables

🞄 224, 256, 384 et 512 bits



Fonction de hachage à sens unique

* Intégrité



Fonction de hachage à sens unique