UNIVERSITÉ DE LA MANOUBA
ÉCOLE NATIONALE DES SCIENCES DE L'INFORMATIQUE

Cours :

SÉCURITÉ INFORMATIQUE

Préparé par :

DR. MAROUA BAKRI

maroua.bakri@ensi-uma.tn

Niveau : II3

PLAN DU CHAPITRE 4 (1/2)

CRYPTOGRAPHIE

1. Introduction
2. La cryptographie
   - Confidentialité des messages
     - Chiffrement symétrique
     - Chiffrement asymétrique
     - Chiffrement mixte ou hybride
   - Authentification, Non-répudiation et Intégrité des messages
     - Fonctions de hachage
     - MAC – Message Authentication Code
     - Signature numérique
   - Gestion de clés
     - Distribution des clés
     - Protocole d'échange de clés de Diffie-Hellman
     - Certificat de clé publique
     - PKI – Public Key Infrastructure
3. La cryptanalyse
   - Attaque sur texte chiffré seul
   - Attaque à texte clair connu
   - Attaque à texte clair choisi
   - Attaque à texte chiffré choisi

Plan du chapitre 4 (2/2)

Cryptographie

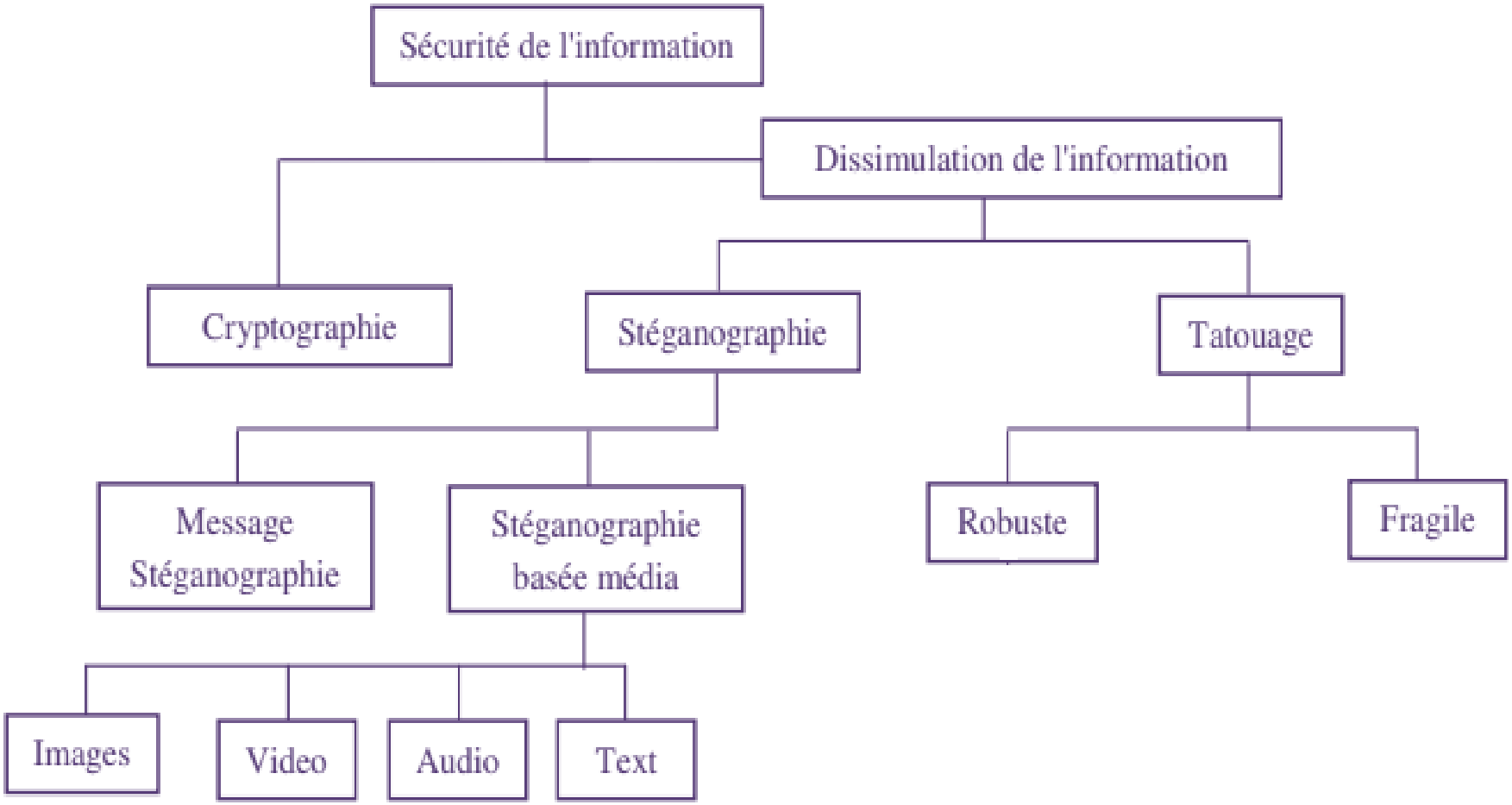
4. La stéganographie
   - Stéganographie linguistique
   - Stéganographie technique
   - Propriétés des systèmes de stéganographie
   - Techniques de stéganographie
5. La stéganalyse
   - Types d'attaques de stéganalyse
   - Méthodes de stéganalyse
6. Le tatouage numérique

Plan du chapitre 4 (1/2)

Cryptographie

1. Introduction
2. La cryptographie
   - Confidentialité des messages
     - Chiffrement symétrique
     - Chiffrement asymétrique
     - Chiffrement mixte ou hybride
   - Authentification, Non-répudiation et Intégrité des messages
     - Fonctions de hachage
     - MAC – Message Authentication Code
     - Signature numérique
   - Gestion de clés
     - Distribution des clés
     - Protocole d'échange de clés de Diffie-Hellman
     - Certificat de clé publique
     - PKI – Public Key Infrastructure
3. La cryptanalyse
   - Attaque sur texte chiffré seul
   - Attaque à texte clair connu
   - Attaque à texte clair choisi
   - Attaque à texte chiffré choisi

TECHNIQUES DE LA SÉCURITÉ DE L'INFORMATION



- ***La cryptographie*** est la science d'écriture d'un message en code secret afin de préserver sa sécurité et sa confidentialité. Le but est donc de brouiller un message afin de le rendre incompréhensible pour les personnes non autorisées.

- La discipline duale de la cryptographie est ***la cryptanalyse***. Elle consiste en l'étude des procédés cryptographiques dans le but de décrypter les messages chiffrés sans posséder la clé de déchiffrement. La cryptographie et la cryptanalyse constituent les deux branches principales de la science mathématique : cryptologie.

- ***La stéganographie*** est l'art de cacher un message secret au sein d'un autre message porteur (texte, image, son, vidéo...) de caractère anodin, de sorte que l'existence même du secret en soit dissimulée.

- De manière analogue à la cryptographie, la stéganographie a également comme discipline duale ***la stéganalyse***. L'objectif principal de la stéganalyse est de détecter la présence d'un message caché, et aussi dans la mesure du possible, d'avoir accès à son contenu.

- ***Le tatouage numérique*** (en anglais digital watermarking) est une technique permettant d'ajouter des informations de copyright ou d'autres messages de vérification à un document numérique de format textuel, graphique, audio ou vidéo.

Plan du chapitre 4 (1/2)

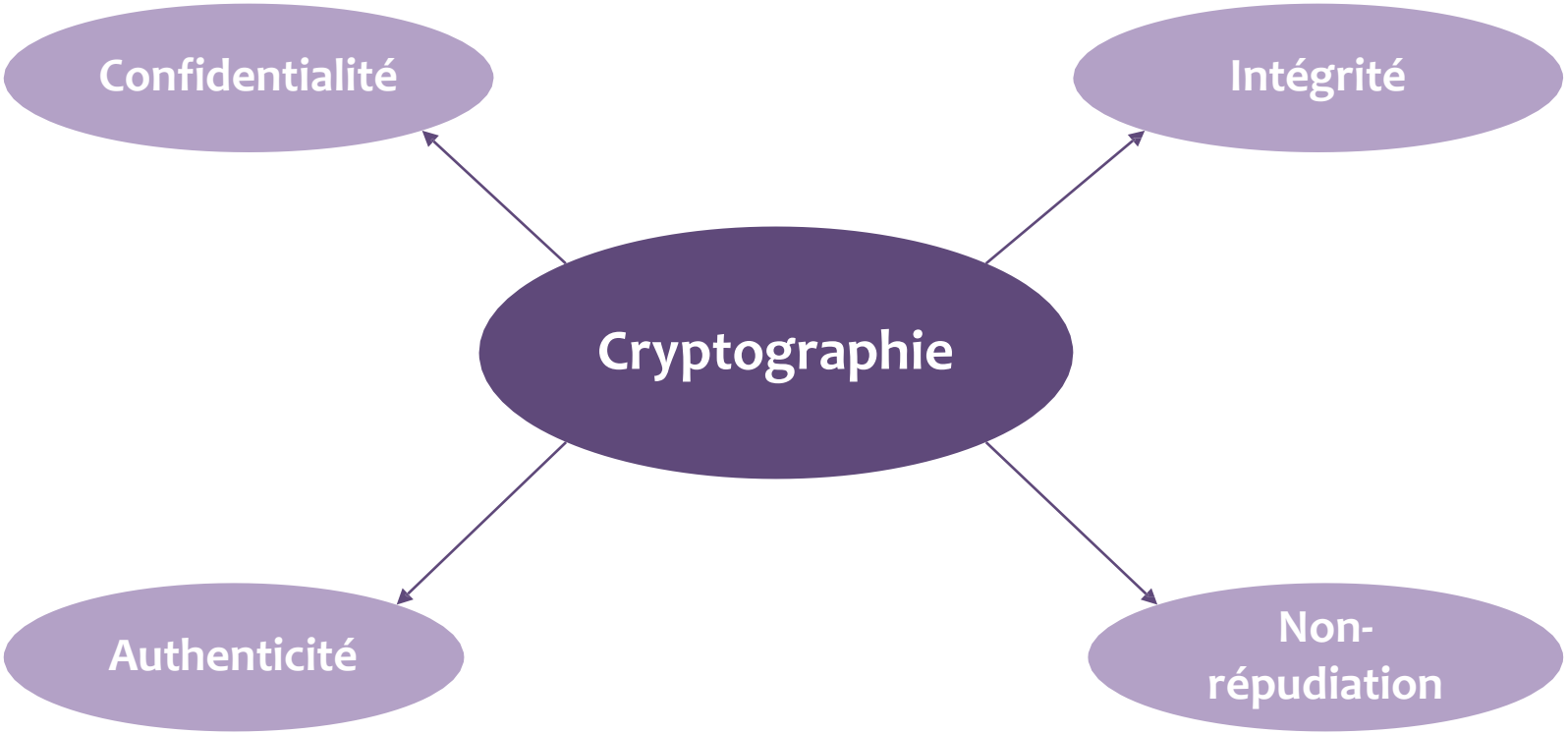
Cryptographie

1. Introduction
2. La cryptographie
   - Confidentialité des messages
     - Chiffrement symétrique
     - Chiffrement asymétrique
     - Chiffrement mixte ou hybride
   - Authentification, Non-répudiation et Intégrité des messages
     - Fonctions de hachage
     - MAC – Message Authentication Code
     - Signature numérique
   - Gestion de clés
     - Distribution des clés
     - Protocole d'échange de clés de Diffie-Hellman
     - Certificat de clé publique
     - PKI – Public Key Infrastructure
3. La cryptanalyse
   - Attaque sur texte chiffré seul
   - Attaque à texte clair connu
   - Attaque à texte clair choisi
   - Attaque à texte chiffré choisi

Plan du chapitre 4 (1/2)

Cryptographie

1. Introduction
2. La cryptographie
   - Confidentialité des messages
     - Chiffrement symétrique
     - Chiffrement asymétrique
     - Chiffrement mixte ou hybride
   - Authentification, Non-répudiation et Intégrité des messages
     - Fonctions de hachage
     - MAC – Message Authentication Code
     - Signature numérique
   - Gestion de clés
     - Distribution des clés
     - Protocole d'échange de clés de Diffie-Hellman
     - Certificat de clé publique
     - PKI – Public Key Infrastructure
3. La cryptanalyse
   - Attaque sur texte chiffré seul
   - Attaque à texte clair connu
   - Attaque à texte clair choisi
   - Attaque à texte chiffré choisi


Confidentialité
Intégrité
Cryptographie
Authenticité
Non-répudiation

- ***La confidentialité*** : Le fait de s'assurer que l'information n'est seulement accessible qu'à ceux dont l'accès est autorisé.
- ***L'intégrité*** : Le fait de s'assurer que l'information ne subisse aucune altération ou destruction volontaire ou accidentelle, et conserve le format initial .
- ***L'authenticité*** : Le fait de s'assurer que l'expéditeur est bien celui qu'il prétend être.
- ***La non-répudiation*** :
  - ***d'origine*** : L'émetteur ne peut nier avoir écrit le message et il peut prouver qu'il ne l'a pas fait si c'est le cas.
  - ***de réception*** : Le receveur ne peut nier avoir reçu le message et il peut prouver qu'il ne l'a pas reçu si c'est effectivement le cas.
  - ***de transmission*** : L'émetteur du message ne peut nier avoir envoyé le message et il peut prouver qu'il ne l'a pas fait si c'est le cas.

Plan du chapitre 4 (1/2)

Cryptographie

1. Introduction
2. La cryptographie
   - Confidentialité des messages
     - Chiffrement symétrique
     - Chiffrement asymétrique
     - Chiffrement mixte ou hybride
   - Authentification, Non-répudiation et Intégrité des messages
     - Fonctions de hachage
     - MAC – Message Authentication Code
     - Signature numérique
   - Gestion de clés
     - Distribution des clés
     - Protocole d'échange de clés de Diffie-Hellman
     - Certificat de clé publique
     - PKI – Public Key Infrastructure
3. La cryptanalyse
   - Attaque sur texte chiffré seul
   - Attaque à texte clair connu
   - Attaque à texte clair choisi
   - Attaque à texte chiffré choisi

- La fonction cryptographique assurant la confidentialité des messages est appelé ***chiffrement (cryptage, encryptage, codage, ou brouillage)***. Il s'agit d'une transformation qui rend le message à transmettre incompréhensible.

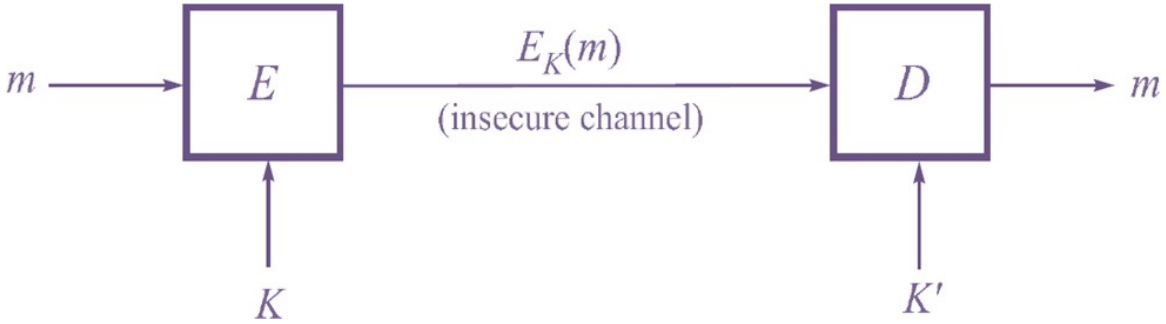
Texte en clair ---Chiffrement---> Texte chiffré (crypté, codé, brouillé, ou cryptogramme)

- La transformation inverse afin de reconstruire le texte en clair à partir du texte chiffré est appelée : **déchiffrement**

Texte chiffré ---Déchiffrement---> Texte en clair

- Le chiffrement classique repose sur deux méthodes principales :
  - ***La substitution (monoalphabétique/polyalphabétique)*** : chaque lettre/groupe de lettres est remplacé(e) par une autre lettre/un autre groupe de lettres.
  - ***La transposition*** : modifie l'ordre des symboles du texte en clair sans les déguiser.
- Pour le chiffrement moderne, les transformations sont des fonctions mathématiques appelées algorithmes cryptographiques qui dépendent d'un paramètre appelé clé.
- Le chiffrement moderne utilise les mêmes bases que le chiffrement traditionnel (substitution et transposition) mais de façon différente.

- Un crypto-système est l'ensemble des deux méthodes de chiffrement et de déchiffrement utilisable en sécurité.



  - Le nœud S envoie le message m au nœud R via un canal non sécurisé.
  - S chiffre m en utilisant un algorithme de chiffrement E et une clé de chiffrement K.
  - $E_K(m)$ est le texte chiffré (Cipher text)
  - m est le texte en clair (Plain text)
  - R décrypte le texte chiffré avec un algorithme de déchiffrement D et d'une clé de déchiffrement K'.

- Propriété de base pour un crypto-système : $D_{k'}(E_k(m)) = m$

Les principes de cryptographie moderne : Principes de Kerckhoffs

1. Le système doit être matériellement, sinon mathématiquement, indéchiffrable ;

   Le système doit offrir un niveau de sécurité suffisant au regard de celui voulu pour le traitement des informations.

2. Il faut qu'il n'exige pas le secret, et qu'il puisse sans inconvénient tomber entre les mains de l'ennemi ;

   La sécurité du système ne doit pas reposer sur le secret de sa conception. Ses spécifications et détails de fonctionnement doivent pouvoir être rendus publics.

3. La clef doit pouvoir en être communiquée et retenue sans le secours de notes écrites, et être changée ou modifiée au gré des correspondants ;

4. Il faut qu'il soit applicable à la correspondance télégraphique ;

   Certes, nous sommes aujourd'hui passés à d'autres moyens de communication. Mais, il faut faire en sorte que le système prenne en entrée et renvoie après traitement des données conformes aux formats et standards en vigueur.

Les principes de cryptographie moderne : Principes de Kerckhoffs

5. Il faut qu'il soit portatif, et que son maniement ou son fonctionnement n'exige pas le concours de plusieurs personnes ;

   Le système doit pouvoir être utilisé quel que soit l'endroit ou l'on se trouve (mobilité de profils).

6. Enfin, il est nécessaire, vu les circonstances qui en commandent l'application, que le système soit d'un usage facile, ne demandant ni tension d'esprit, ni la connaissance d'une longue série de règles à observer.

   Il s'agit de s'attacher, lors de la conception, l'implémentation et l'intégration d'un système de sécurité à lui donner une ergonomie qui soit la meilleure possible (un système qui sera simple d'accès).

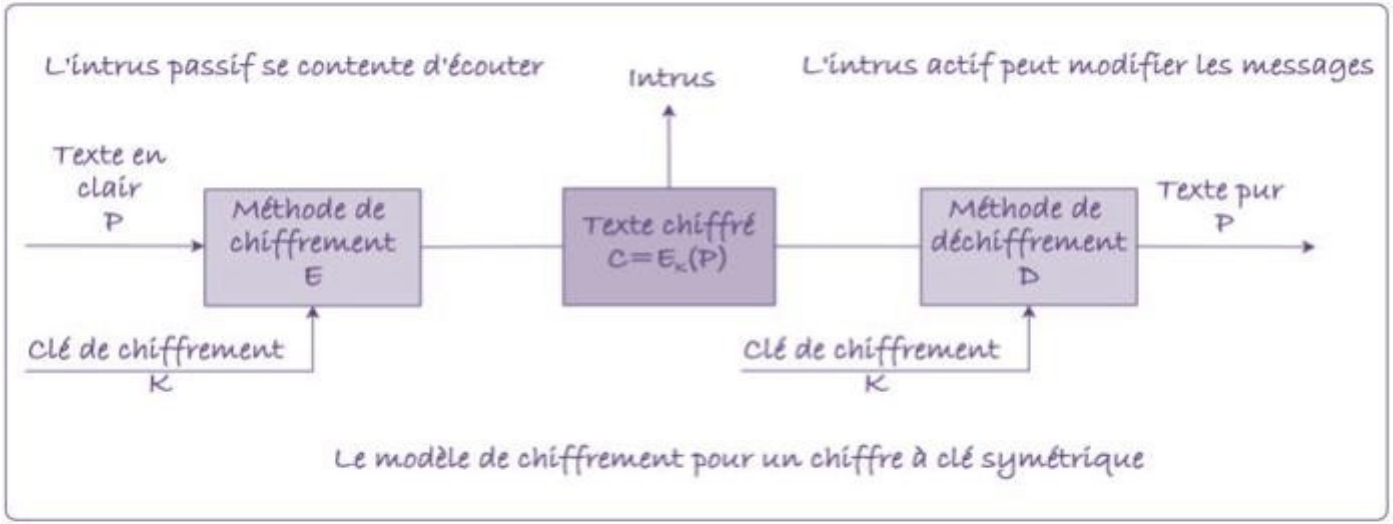
Plan du chapitre 4 (1/2)

Cryptographie

1. Introduction
2. La cryptographie
   - Confidentialité des messages
     - Chiffrement symétrique
     - Chiffrement asymétrique
     - Chiffrement mixte ou hybride
   - Authentification, Non-répudiation et Intégrité des messages
     - Fonctions de hachage
     - MAC – Message Authentication Code
     - Signature numérique
   - Gestion de clés
     - Distribution des clés
     - Protocole d'échange de clés de Diffie-Hellman
     - Certificat de clé publique
     - PKI – Public Key Infrastructure
3. La cryptanalyse
   - Attaque sur texte chiffré seul
   - Attaque à texte clair connu
   - Attaque à texte clair choisi
   - Attaque à texte chiffré choisi

Principe

- On parle de système de chiffrement symétrique lorsque les clés utilisées pour le chiffrement et le déchiffrement sont identiques.
- Le chiffrement à clé symétrique appelé aussi à clé secrète repose sur la no-divulgation des clés et la résistance des algorithmes aux attaques de cryptanalyse.
- Le chiffrement symétrique utilise des canaux sécurisés pour échanger la clé secrète (IPSec).



Le modèle de chiffrement pour un chiffre à clé symétrique

Avantages et inconvénients

❑ Avantages

- ✓ Le chiffrement/déchiffrement est très rapide : les algorithmes de chiffrement symétrique sont généralement beaucoup moins complexes que les algorithmes de chiffrement asymétrique.
- ✓ Utilise peu de ressources systèmes, toujours dans le même principe d'algorithme moins complexe.

Avantages et inconvénients

❑ Inconvénients

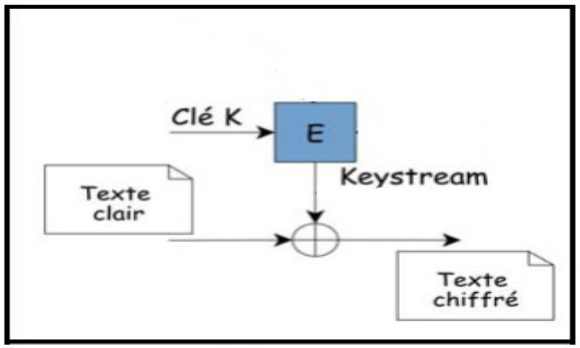
- ✓ Le chiffrement symétrique n'assure que la confidentialité des données, contrairement au chiffrement asymétrique qui permet d'assurer des principes de sécurité supplémentaires tq l'authenticité de l'émetteur et la non-répudiation de l'envoi.
- ✓ Une clé symétrique correspond à un échange entre 2 personnes, pour communiquer avec d'autres personnes il faudra une autre clé symétrique.
  - Soit un grand nombre de clé selon le nombre de personnes avec qui on communique. Pour n personnes, il faut $n(n-1)/2$ clés secrètes.
- ✓ L'utilisation d'une clé unique présente un problème :
  - Communiquer la clé de manière sûre à la personne avec laquelle on souhaite dialoguer.
  - Il est nécessaire de garantir la confidentialité de cette clé. Les échanges qui suivront reposent sur celle-ci. En d'autres termes, si une tierce personne accède à la clé, elle pourra lire, modifier, altérer tous les échanges qui s'effectueront entre les 2 protagonistes de départ.
  - Nécessité d'un canal sécurisé pour envoyer la clé secrète générée d'un côté du canal de communication vers l'autre côté.

Modes de chiffrement symétrique

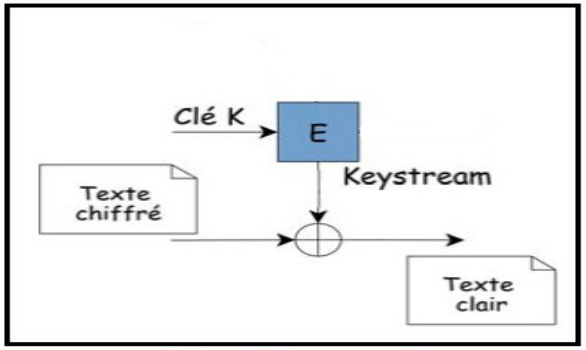
- Le chiffrement symétrique (ou à clé secrète) est très utilisé et se caractérise par une grande rapidité ce qui accélère nettement les débits et autorise son utilisation massive.
- Le chiffrement symétrique fonctionne habituellement suivant deux procédés différents :
  - ✓ Chiffrement par bloc (Block cipher)
  - ✓ Chiffrement par flux (Stream cipher)

Chiffrement par flux (Stream cipher) (1/2)

- Le chiffrement par flux (Stream cipher) est effectué bit-à-bit sans attendre la réception complète des données à crypter.
- Le chiffrement par flux repose sur un générateur pseudo-aléatoire (keystream) produisant une séquence de bits précise utilisée en tant que clé.
- Un algorithme de chiffrement par flux repose sur un générateur pseudo-aléatoire (GPA) qui étend une clé secrète K de k bits en une suite chiffrante pseudo-aléatoire S (keystream) de L bits (L > k).
  - ✓ **Chiffrement :** pour tout message en clair P de n bits (n ≤ L), le message chiffré C est obtenu par $\forall i \in [1, n]$, $C_i = P_i \oplus S_i$.
  - ✓ **Déchiffrement :** le message en clair P est obtenu par $\forall i \in [1, n]$, $P_i = C_i \oplus S_i$.



Chiffrement



Déchiffrement

Chiffrement par flux (Stream cipher) (2/2)

- **Propriétés :**
  - La suite chiffrante S (Keystream) ne dépend pas du message clair, mais uniquement de la clé ;
  - Il est possible de chiffrer des messages de tailles variables ;
  - Le chiffrement et le déchiffrement s'effectuent de la même manière, puisque le « Ou exclusif » est une opération involutive ;
  - L'impact de la modification d'une partie du message chiffré pendant la transmission du message est limité à cette partie du message déchiffré ;
  - Les algorithmes de chiffrement par flux sont de façon générale :
    - Très rapides (en matériel et en logiciel)
    - Implémentation matérielle avec peu de portes logiques
    - Adaptés aux applications temps réel (telles que le WI-FI (algorithme RC4))

Chiffrement par bloc (Block cipher) (1/9)

- Un algorithme de chiffrement par bloc (Block Cipher) transforme des blocs de données de taille fixe en bloc de données chiffrées de la même taille.
- La transformation reste la même pour chaque bloc.
- La longueur n des blocs et la taille l de la clé utilisée sont deux caractéristiques des systèmes de chiffrement par blocs.
- Si la longueur du message n'est pas un multiple de la longueur d'un bloc, on le complète avec une séquence de bourrage.
- **Exemple de technique de bourrage :**
  - ✓ Une façon de bourrer (RFC2040) consiste à compléter le dernier bloc par autant d'octets que nécessaire, chaque octet ayant pour valeur le nombre d'octets ajoutés.

    Exple : s'il manque 3 octets au message M, pour obtenir un bloc de 8 octets on ajoute 3 octets égaux à 3 :

o_1	o_2	o_3	o_4	o_5	$(03)_H$	$(03)_H$	$(03)_H$

  - ✓ S'il se trouve que la taille de la donnée à chiffrer est un multiple de la taille d'un bloc, on ajoute un bloc entier dont chaque octet a pour valeur la taille en octet d'un bloc.

    Exple : pour des blocs de 8 octets, on ajoute le bloc suivant :

$(08)_H$	$(08)_H$	$(08)_H$	$(08)_H$	$(08)_H$	$(08)_H$	$(08)_H$	$(08)_H$

Chiffrement par bloc (Block cipher) (2/9)

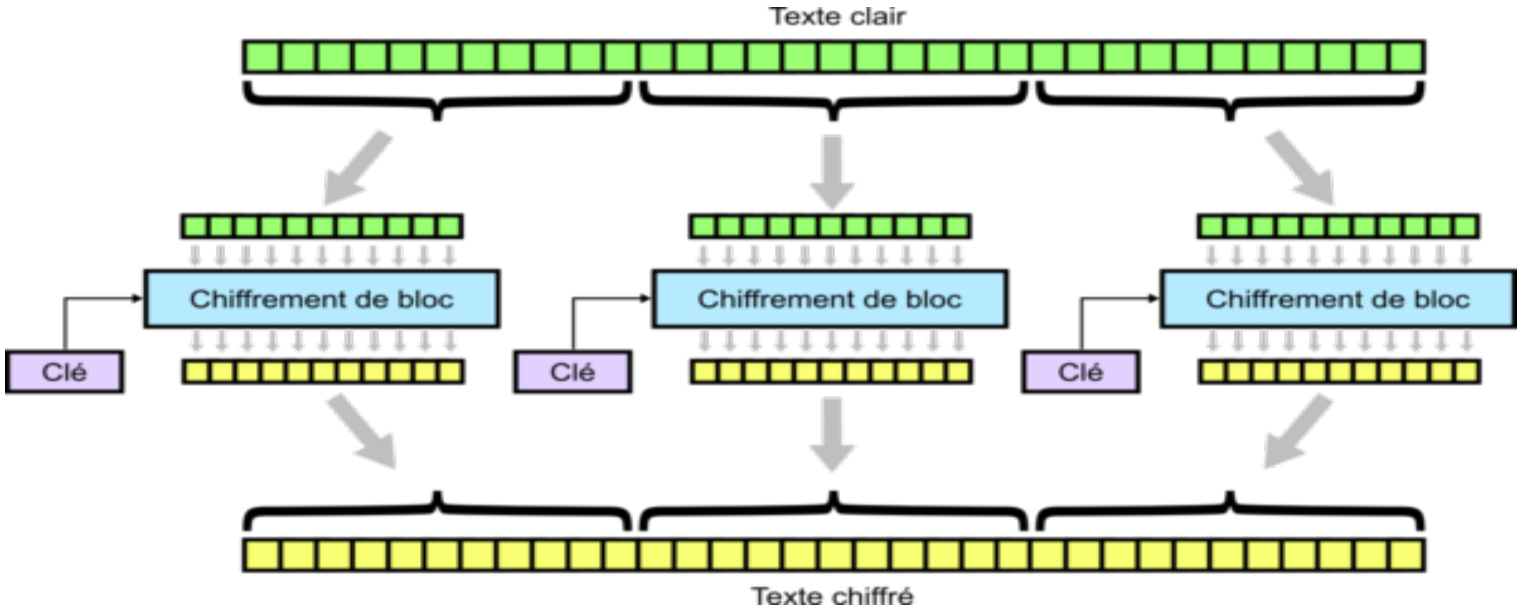
Il existe 4 modes de chiffrement par bloc :

- ✓ Electronic CodeBook (ECB);
- ✓ Cipher Block Chaining (CBC);
- ✓ Cipher FeedBack (CFB) ;
- ✓ Output FeedBack (OFB).

Chiffrement par bloc (Block cipher) (3/9)

Electronic CodeBook (ECB)

- Le mode Electronic CodeBook (ECB) est le plus simple des modes s'appliquant aux block ciphers.
- Le message à chiffrer est subdivisé en plusieurs blocs qui sont chiffrés séparément les uns après les autres.



Chiffrement par bloc (Block cipher) (4/9)

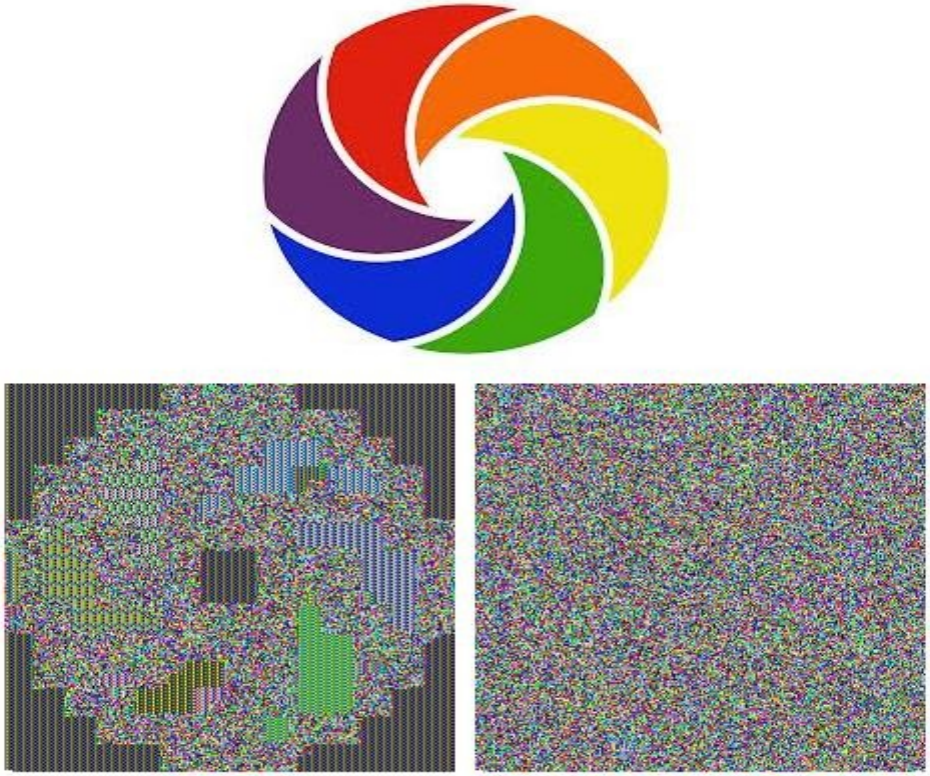
Electronic CodeBook (ECB)

- Ce mode est très vulnérable aux attaques :
  - Étant donné que deux blocs avec le même contenu seront chiffrés de la même manière, on peut donc tirer des informations à partir du texte chiffré en cherchant les séquences identiques. Ainsi, il est possible de recenser tous les cryptés possibles (code books) puis par recoupements et analyses statistiques recomposer une partie du message original sans avoir tenté de casser la clé de chiffrement.
  - Le mode ECB ne respecte pas l'intégrité des données. Un attaquant peut remplacer certains blocs chiffrés par d'autres blocs chiffrés du message, ou permuter deux blocs, sans que le destinataire s'en aperçoive. Imaginons que le message chiffré soit le montant d'une transaction électronique, et que l'attaquant arrive à permuter deux chiffres !

Pour ces raisons, l'utilisation du mode ECB n'est pas recommandée.

Chiffrement par bloc (Block cipher) (5/9)

Electronic CodeBook (ECB)

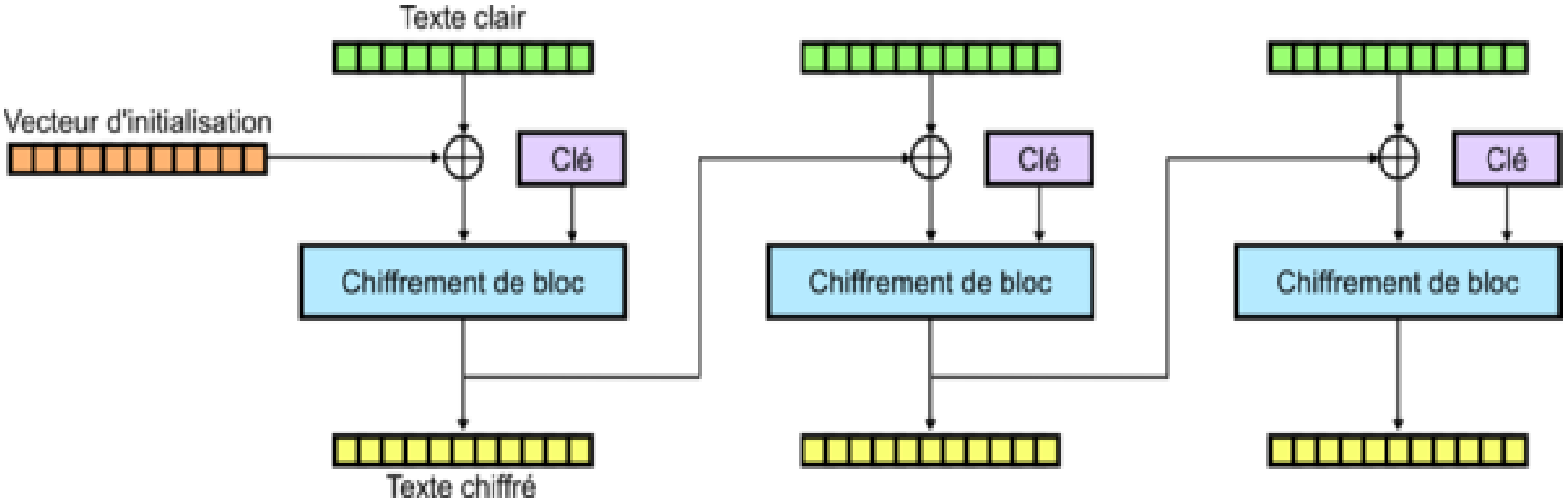


The bitmap image encrypted using DES and the same secret key. The ECB mode was used for the left image and the CBC mode was used for the right image.

Chiffrement par bloc (Block cipher) (6/9)

Cipher Block Chaining (CBC)

- Dans le mode Cipher Block Chaining (CBC), on applique sur chaque bloc un « OU exclusif » avec le chiffrement du bloc précédent avant qu'il soit lui-même chiffré. Le but est de rendre chaque message unique.
- Utilisation d'un vecteur d'initialisation (Initialization Vector, IV) qui change à chaque session et qui doit être transmis au destinataire.



Chiffrement par bloc (Block cipher) (7/9)

Cipher Block Chaining (CBC)

Ce mode a plusieurs avantages, et aussi un gros inconvénient.

- **Avantages :**

  Le mode CBC chiffre le même message clair différemment avec des blocs d'initialisation différents. De plus, le chiffrement d'un bloc dépend également des blocs précédents, et par conséquent, si l'ordre des blocs du cryptogramme est modifié, le déchiffrement est impossible et le destinataire se rend compte du problème.

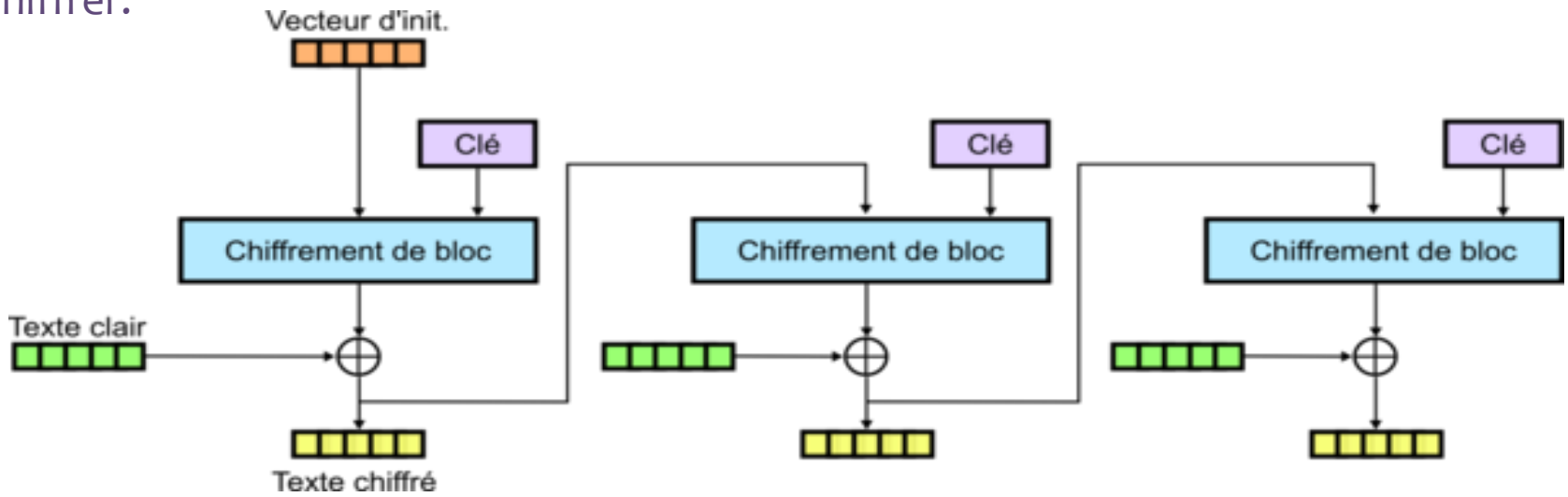
- **Inconvénient :**

  L'inconvénient principal de ce mode est sa lenteur. Les algorithmes de chiffrement et de déchiffrement sont assez longs à mettre en œuvre et ne sont pas adéquats pour les applications temps réel (par exemple : une communication téléphonique) .

Chiffrement par bloc (Block cipher) (8/9)

Cipher FeedBack (CFB)

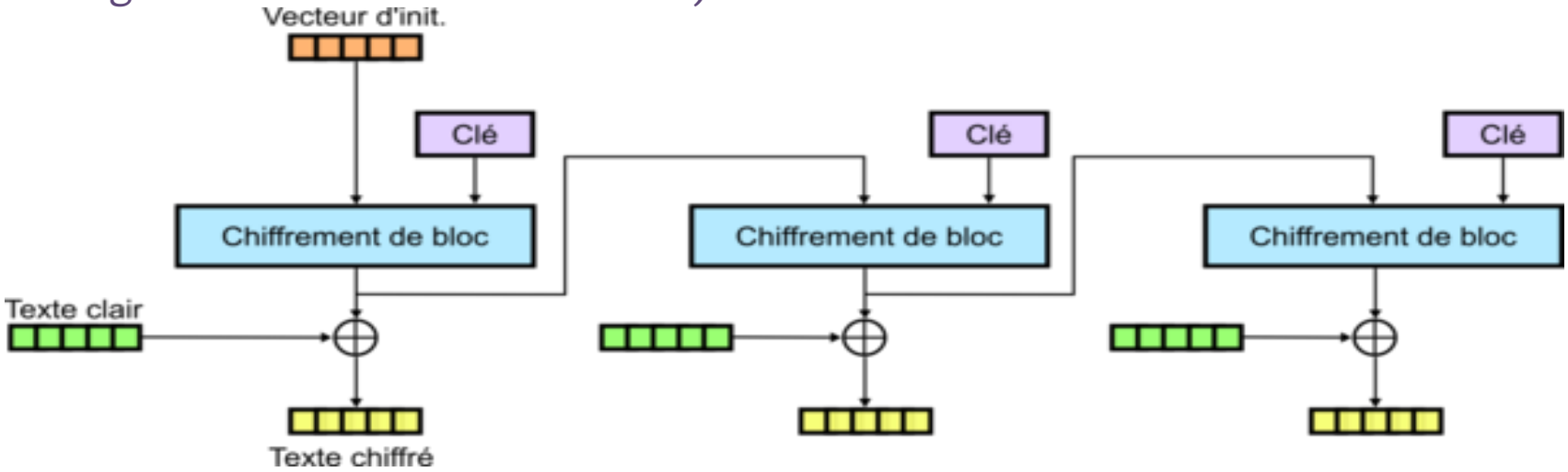
- Le mode Cipher FeedBack (CFB) est un mode destiné aux block ciphers dans le but d'en autoriser une utilisation plus souple, qui s'apparente plus à celle des algorithmes de chiffrement par flux.
- Comme dans les algorithmes de chiffrement par flux, un flux de clé (keystream) est généré pour être utilisé par la suite pour le cryptage de chaque bloc du message à chiffrer.



Chiffrement par bloc (Block cipher) (9/9)

Output FeedBack (OFB)

- Le mode Output FeedBack (OFB) est une variante de mode CFB. Dans ce mode, le flux de clé est obtenu en chiffrant le précédent flux de clé.
- Il présente beaucoup de problèmes de sécurité et il est peu conseillé (si un attaquant arrive à connaître le vecteur d'initialisation d'un message chiffré ainsi que le clair d'un autre message chiffré, il peut reconstituer aisément la chaîne ayant chiffré le premier message et donc déchiffrer ce dernier).



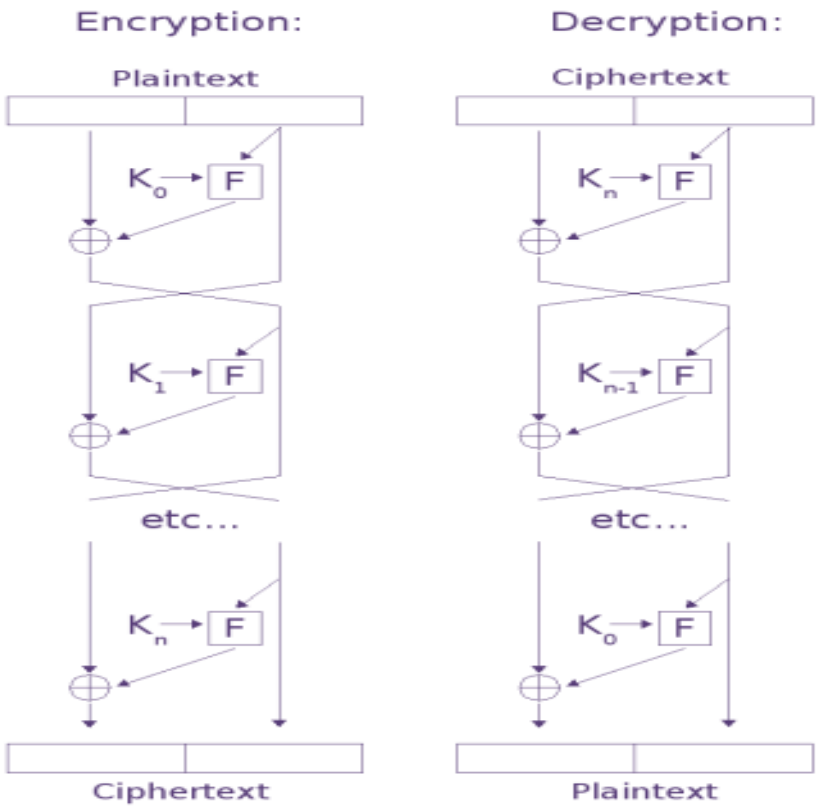
Algorithmes de chiffrement symétrique

La structure de Feistel

- La structure de Feistel fut décrite en 1973 par Feistel, employé chez IBM.
- La plupart des chiffrements de la fin du XXè siècle sont basés sur cette structure.
- Structure inversible ce qui permet de réutiliser le matériel de chiffrement pour déchiffrer un message. La seule modification s'opère dans la manière dont la clé est utilisée.
- Dans une structure de Feistel, le bloc d'entrée d'un round (itération) est séparé en deux parties. La fonction de chiffrement est appliquée sur la première partie du bloc et l'opération binaire ou-exclusive est appliquée sur la partie sortante de la fonction de chiffrement et la deuxième partie. Ensuite, les deux parties sont permutées et le prochain round commence.
- Les performances de la structure de Feistel dépendent des choix effectués pour les paramètres suivants :
  - Taille du bloc/de la clé : si elle augmente, la sécurité augmente également.
  - Nombre de rounds : plus il y en a, plus la sécurité est renforcée.
  - Algorithme de génération des clés : plus il est complexe, plus la sécurité est renforcée.

Algorithmes de chiffrement symétrique

La structure de Feistel



Feistel Cipher

Algorithmes de chiffrement symétrique

DES (Data Encryption Standard)

15 mai 1973 — Le NIST (National Institute of Standards and Technology) lance un appel d'offre pour la proposition d'un algorithme de chiffrement ayant les propriétés suivantes :

- Posséder un haut niveau de sécurité lié à une clé de petite taille servant au chiffrement et au déchiffrement
- Être compréhensible
- Ne pas dépendre de la confidentialité de l'algorithme
- Être adaptable et économique
- Être efficace et exportable

1974 — IBM propose "Lucifer", qui grâce à la NSA (National Security Agency) est modifié le 23 novembre 1976 pour donner le DES (Data Encryption Standard).

1978 — Le DES a été approuvé par le NIST.

Algorithmes de chiffrement symétrique

DES (Data Encryption Standard)

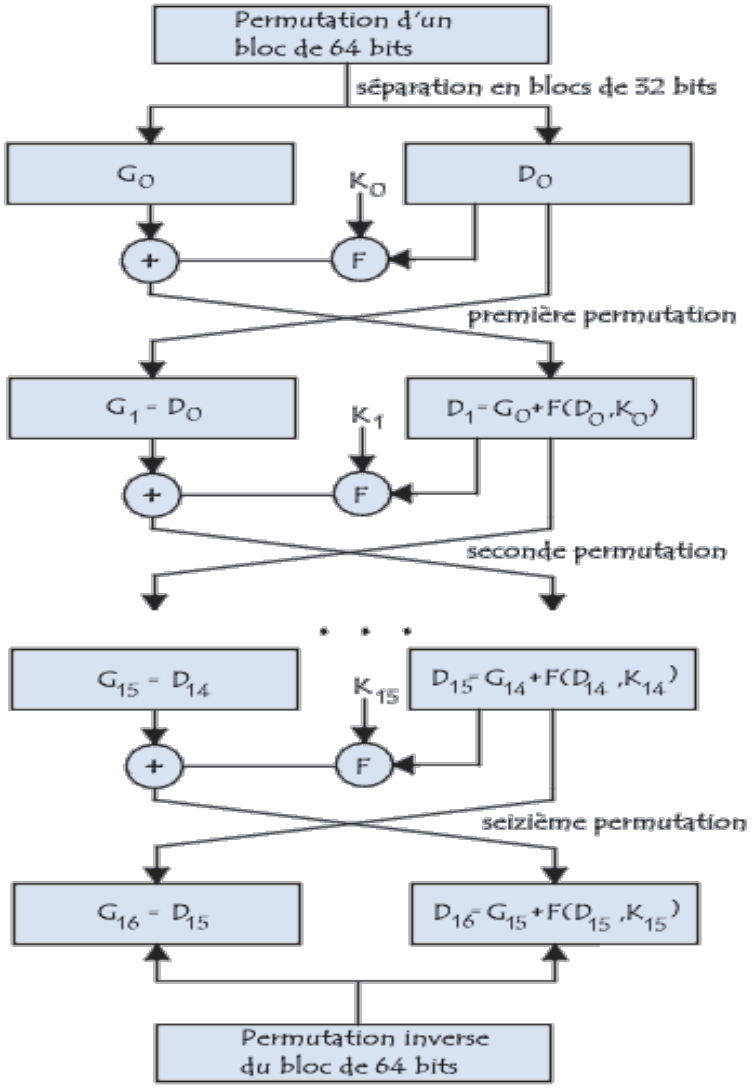
- DES est un algorithme de chiffrement symétrique par blocs de 64 bits.
- DES utilise une clé de chiffrement de longueur "utile" 56 bits.
  - ✓ La clé est complétée par des bits de parité (un octet) servant à vérifier l'intégrité de la clé.
  - ✓ Chaque bit de parité est ajusté de façon à avoir un nombre impair de '1' dans l'octet auquel il appartient.
- L'algorithme consiste à effectuer des combinaisons, des substitutions et des permutations entre le texte à chiffrer et la clé, en faisant en sorte que les opérations puissent se faire dans les deux sens (pour le chiffrement et le déchiffrement).
- La combinaison entre substitutions et permutations est appelée code produit.

Algorithmes de chiffrement symétrique

DES (Data Encryption Standard)

Les grandes lignes de l'algorithme sont les suivantes :

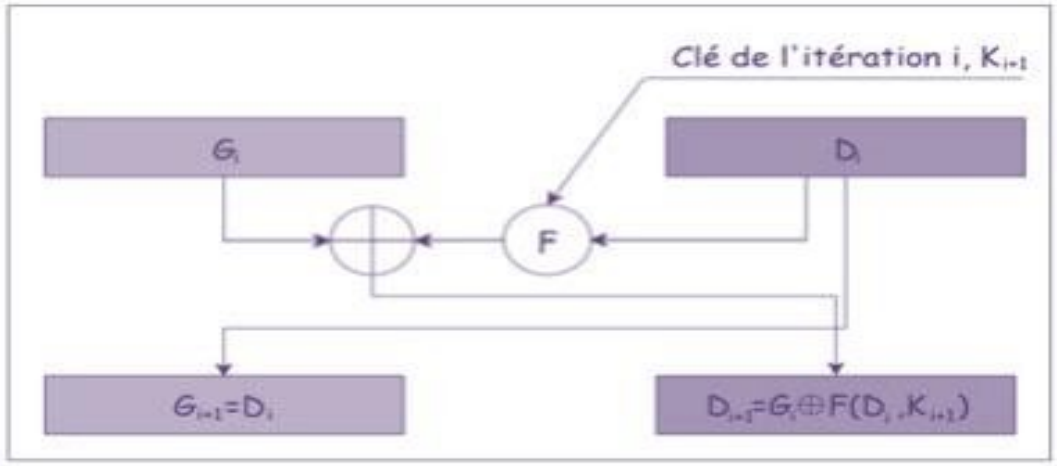
- Fractionnement du texte en blocs de 64 bits (8 octets) ;
- Permutation initiale des blocs ;
- Découpage des blocs en deux parties: gauche et droite, nommées G et D ;
- Étapes de permutation et de substitution répétées 16 fois (appelées rondes) ;
- Recollement des parties gauche et droite puis permutation initiale inverse.



Algorithmes de chiffrement symétrique

DES (Data Encryption Standard)

- Chaque étage reçoit deux entrées de 32 bits et restitue deux sorties de 32 bits.
- La partie gauche de la sortie est une simple recopie de la partie droite de l'entrée.
- La partie droite de la sortie est le résultat d'un OU exclusif bit à bit entre la partie gauche de l'entrée et une fonction de la partie droite de l'entrée et de la clé K_{i+1} relative à l'étage i considéré.



Algorithmes de chiffrement symétrique

DES (Data Encryption Standard)

Permutation initiale

- Chaque bit d'un bloc est soumis à la permutation initiale, pouvant être représentée par la matrice de permutation initiale (notée PI) suivante :

$$PI = \begin{pmatrix} 58 & 50 & 42 & 34 & 26 & 18 & 10 & 2 \\ 60 & 52 & 44 & 36 & 28 & 20 & 12 & 4 \\ 62 & 54 & 46 & 38 & 30 & 22 & 14 & 6 \\ 64 & 56 & 48 & 40 & 32 & 24 & 16 & 8 \\ 57 & 49 & 41 & 33 & 25 & 17 & 9 & 1 \\ 59 & 51 & 43 & 35 & 27 & 19 & 11 & 3 \\ 61 & 53 & 45 & 37 & 29 & 21 & 13 & 5 \\ 63 & 55 & 47 & 39 & 31 & 23 & 15 & 7 \end{pmatrix}$$

- Le 58ème bit du bloc de texte de 64 bits se retrouve en première position, le 50ème en seconde position et ainsi de suite.

Algorithmes de chiffrement symétrique

DES (Data Encryption Standard)

Scindement en blocs de 32 bits

- Une fois la permutation initiale réalisée, le bloc de 64 bits est scindé en deux blocs de 32 bits, notés respectivement G et D. Soient Go et Do l'état initial de ces deux blocs :

$$G_0 = \begin{pmatrix} 58 & 50 & 42 & 34 & 26 & 18 & 10 & 2 \\ 60 & 52 & 44 & 36 & 28 & 20 & 12 & 4 \\ 62 & 54 & 46 & 38 & 30 & 22 & 14 & 6 \\ 64 & 56 & 48 & 40 & 32 & 24 & 16 & 8 \end{pmatrix} \quad D_0 = \begin{pmatrix} 57 & 49 & 41 & 33 & 25 & 17 & 9 & 1 \\ 59 & 51 & 43 & 35 & 27 & 19 & 11 & 3 \\ 61 & 53 & 45 & 37 & 29 & 21 & 13 & 5 \\ 63 & 55 & 47 & 39 & 31 & 23 & 15 & 7 \end{pmatrix}$$

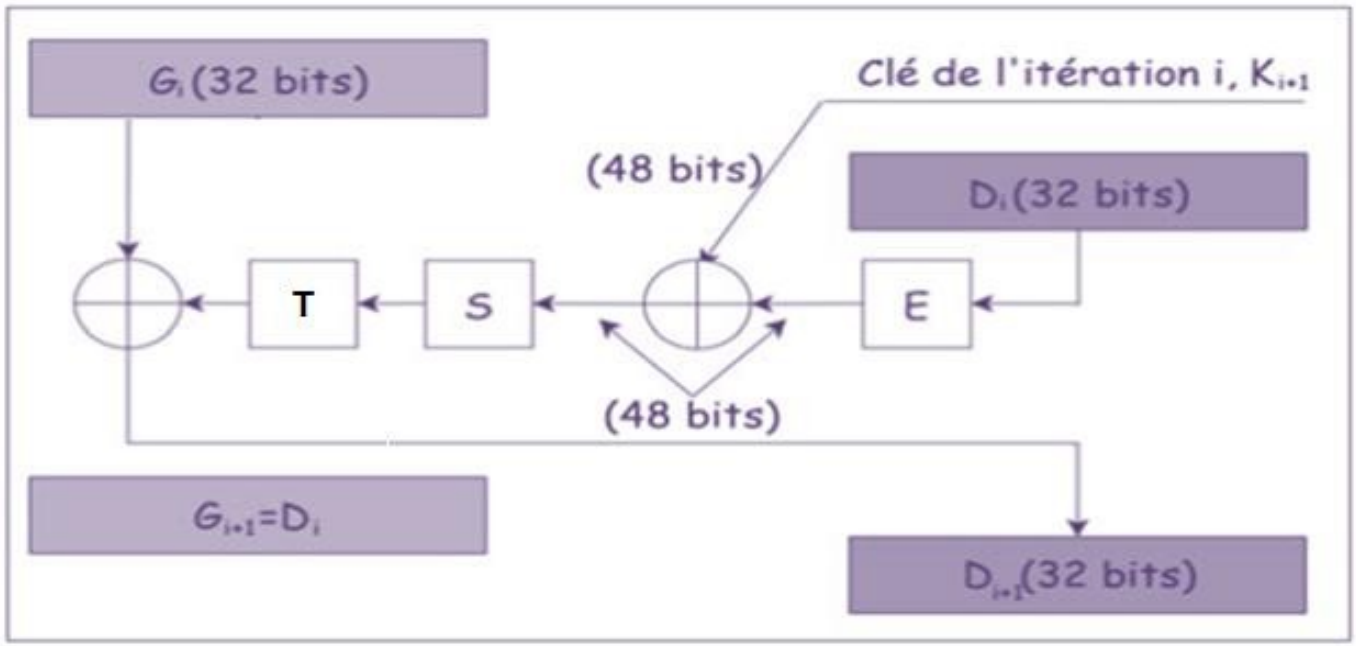
- Il est à noter que G_0 contient tous les bits possédant une position paire dans le message initial, tandis que D_0 contient les bits de position impaire.

Algorithmes de chiffrement symétrique

DES (Data Encryption Standard)

Rondes

- Les blocs G_i et D_i sont soumis à un ensemble de transformations itératives appelées rondes.



Algorithmes de chiffrement symétrique

DES (Data Encryption Standard)

Fonction d'expansion

- Les 32 bits du bloc D_0 sont étendus à 48 bits grâce à une table (matrice) appelée table d'expansion (notée E), dans laquelle les 48 bits sont mélangés et 16 d'entre eux sont dupliqués.

$$D_0 = \begin{pmatrix} 57 & 49 & 41 & 33 & 25 & 17 & 9 & 1 \\ 59 & 51 & 43 & 35 & 27 & 19 & 11 & 3 \\ 61 & 53 & 45 & 37 & 29 & 21 & 13 & 5 \\ 63 & 55 & 47 & 39 & 31 & 23 & 15 & 7 \end{pmatrix} \qquad E = \begin{pmatrix} 32 & 1 & 2 & 3 & 4 & 5 \\ 4 & 5 & 6 & 7 & 8 & 9 \\ 8 & 9 & 10 & 11 & 12 & 13 \\ 12 & 13 & 14 & 15 & 16 & 17 \\ 16 & 17 & 18 & 19 & 20 & 21 \\ 20 & 21 & 22 & 23 & 24 & 25 \\ 24 & 25 & 26 & 27 & 28 & 29 \\ 28 & 29 & 30 & 31 & 32 & 1 \end{pmatrix}$$

- Ainsi, le dernier bit de D_0 (càd le $7^{ème}$ 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 D_0 (respectivement 57, 33, 25, l, 59, 35, 27, 3, 6l, 37, 29, 5, 63, 39, 31 et 7 du bloc d'origine) sont dupliqués et disséminés dans la matrice.
- La matrice résultante de 48 bits est notée $E[D_0]$. L'algorithme DES procède ensuite à un OU exclusif entre la première clé K_1 et $E[D_0]$. Le résultat de ce OU exclusif est une matrice de 48 bits que nous appellerons D_0 par commodité (il ne s'agit pas du D_0 de départ!).

Algorithmes de chiffrement symétrique

DES (Data Encryption Standard)

Fonction de substitution

- D_0 est scindé en 8 blocs de 6 bits, noté D_{0i}. Chacun de ces blocs traverse une boîte de substitution notée généralement S_i .
- Les premiers et derniers bits de chaque D_{0i} détermine (en binaire) la ligne de la fonction de sélection, les autres bits (respectivement 2, 3, 4 et 5) déterminent la colonne.
  - ✓ La sélection de la ligne se faisant sur deux bits, il y a 4 possibilités (0,1,2,3) ;
  - ✓ La sélection de la colonne se faisant sur 4 bits, il y a 16 possibilités (0 à 15) ;
  - ✓ La boîte de substitution ”sélectionne” une valeur codée sur 4 bits ;
- Ci-après la première fonction de substitution, représentée par une matrice de 4 par 16 :

$$S_1 = \begin{pmatrix} 14 & 4 & 13 & 1 & 2 & 15 & 11 & 8 & 3 & 10 & 6 & 12 & 5 & 9 & 0 & 7 \\ 0 & 15 & 7 & 4 & 14 & 2 & 13 & 1 & 10 & 6 & 12 & 11 & 9 & 5 & 3 & 8 \\ 4 & 1 & 14 & 8 & 13 & 6 & 2 & 11 & 15 & 12 & 9 & 7 & 3 & 10 & 5 & 0 \\ 15 & 12 & 8 & 2 & 4 & 9 & 1 & 7 & 5 & 11 & 3 & 14 & 10 & 0 & 6 & 13 \end{pmatrix}$$

Algorithmes de chiffrement symétrique

DES (Data Encryption Standard)

Fonction de substitution

- Soit D_{01} égal à 101110 :
  - Le premier et le dernier bit (10) donnent $(2)_{10}$ => ligne 2
  - Les bits 2, 3, 4 et 5 (0111) donnent $(7)_{10}$ => colonne 7
  - Le résultat de la boite de substitution est donc la valeur $(11)_{10}$ soit en binaire 1011 ;
- Chacun des 8 blocs de 6 bits est passé dans la boite de substitution correspondante, ce qui donne en sortie 8 valeurs de 4 bits chacune.

Algorithmes de chiffrement symétrique

DES (Data Encryption Standard)

Permutation sur 32 bits

- Le bloc de 32 bits obtenu est soumis à une permutation T dont la table est la suivante :

$$T = \begin{pmatrix} 16 & 7 & 20 & 21 & 29 & 12 & 28 & 17 \\ 1 & 15 & 23 & 26 & 5 & 18 & 31 & 10 \\ 2 & 8 & 24 & 14 & 32 & 27 & 3 & 9 \\ 19 & 13 & 30 & 6 & 22 & 11 & 4 & 25 \end{pmatrix}$$

- L'ensemble de ces résultats en sortie de T est soumis à un OU Exclusif avec le G_0 de départ pour donner D_1, tandis que le D_0 initial donne G_1.

Algorithmes de chiffrement symétrique

DES (Data Encryption Standard)

Permutation initiale inverse

- À la fin des 16 itérations les deux blocs G_{16} et D_{16} sont recollés puis soumis à la permutation initiale inverse.

  N.B : $PI^{-1}(PI(X)) = X$

$$PI^{-1} = \begin{pmatrix} 40 & 8 & 48 & 16 & 56 & 24 & 64 & 32 \\ 39 & 7 & 47 & 15 & 55 & 23 & 63 & 31 \\ 38 & 6 & 46 & 14 & 54 & 22 & 62 & 30 \\ 37 & 5 & 45 & 13 & 53 & 21 & 61 & 29 \\ 36 & 4 & 44 & 12 & 52 & 20 & 60 & 28 \\ 35 & 3 & 43 & 11 & 51 & 19 & 59 & 27 \\ 34 & 2 & 42 & 10 & 50 & 18 & 58 & 26 \\ 33 & 1 & 41 & 9 & 49 & 17 & 57 & 25 \end{pmatrix}$$

- Le résultat en sortie est un texte codé sur 64 bits.

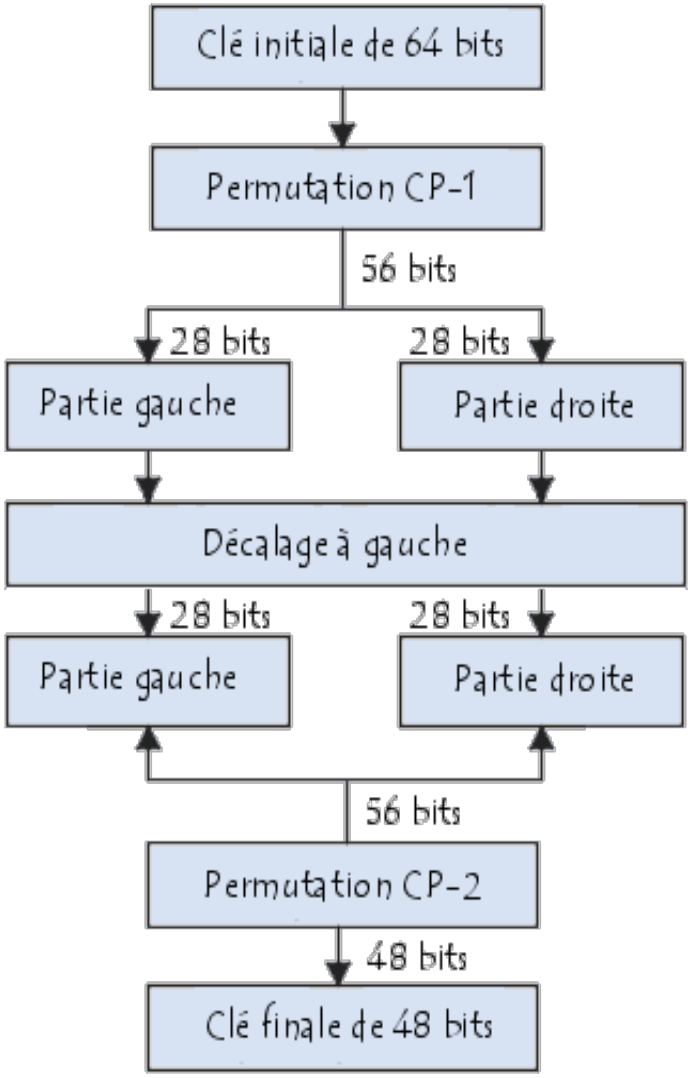
Algorithmes de chiffrement symétrique

DES (Data Encryption Standard)

Génération des clés

- Les bits de parité de la clé sont éliminés afin d'obtenir une clé d'une longueur utile de 56 bits.
- Les 56 bits de la clé subissent par la suite une permutation notée CP^{-1} dont la matrice est la suivante :

$$CP-1 = \begin{pmatrix} 57 & 49 & 41 & 33 & 25 & 17 & 9 & 1 & 58 & 50 & 42 & 34 & 26 & 18 \\ 10 & 2 & 59 & 51 & 43 & 35 & 27 & 19 & 11 & 3 & 60 & 52 & 44 & 36 \\ 63 & 55 & 47 & 39 & 31 & 23 & 15 & 7 & 62 & 54 & 46 & 38 & 30 & 22 \\ 14 & 6 & 61 & 53 & 45 & 37 & 29 & 21 & 13 & 5 & 28 & 20 & 12 & 4 \end{pmatrix}$$



Algorithmes de chiffrement symétrique

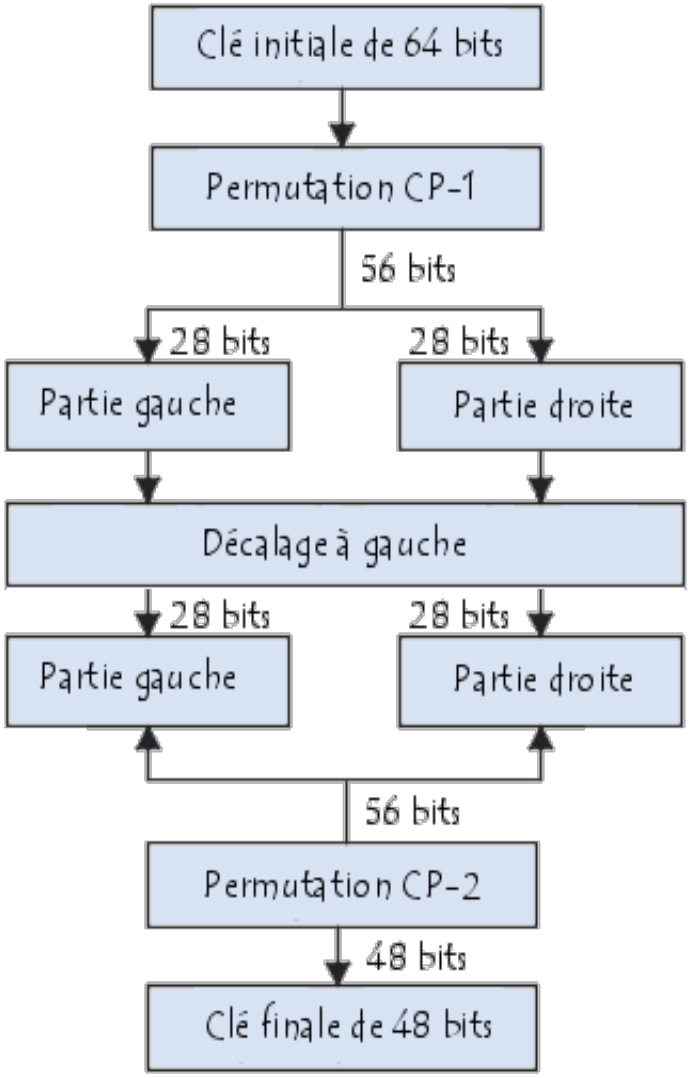
DES (Data Encryption Standard)

Génération des clés

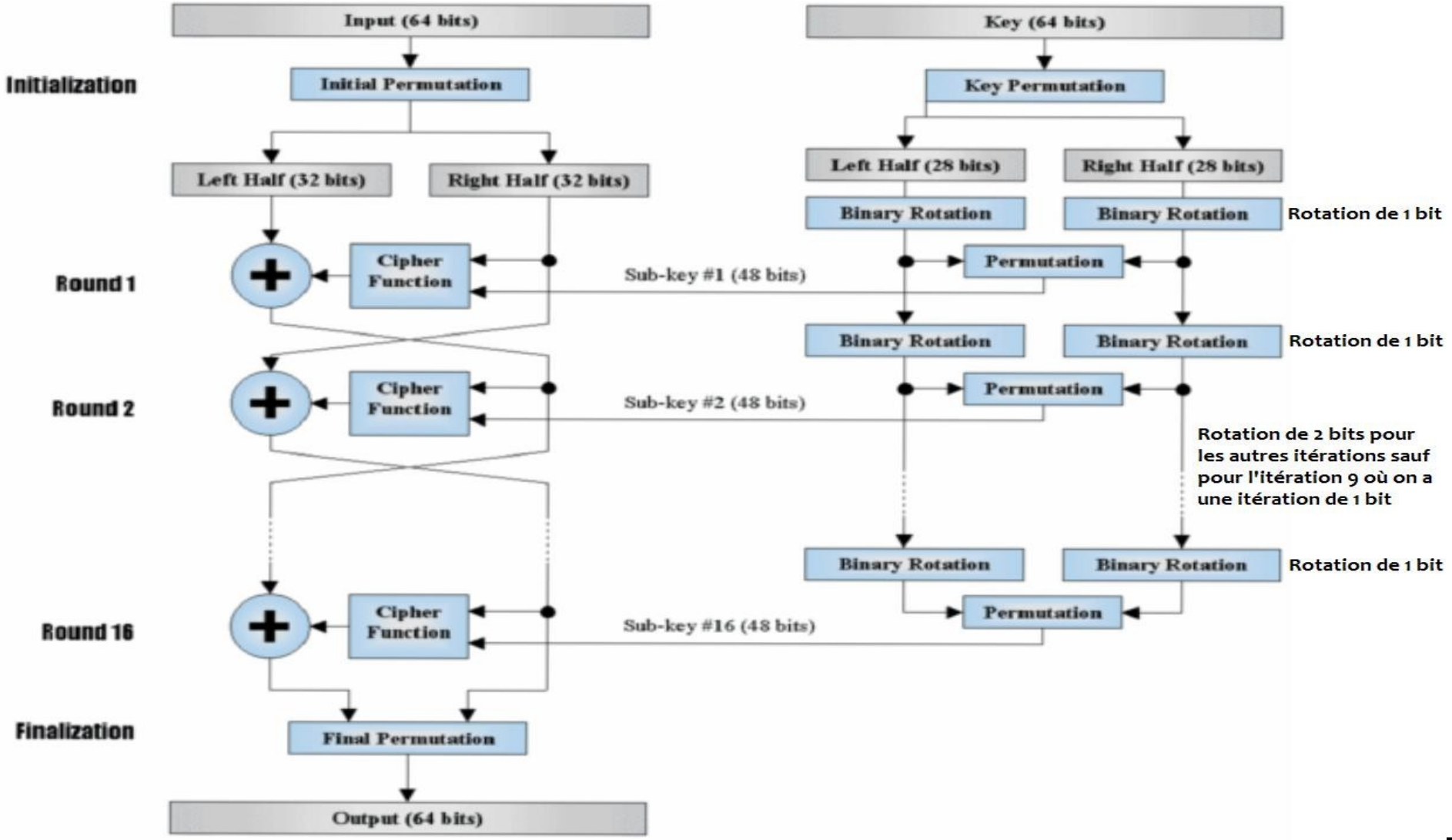
- Cette matrice peut s'écrire sous la forme de deux matrice G_i et D_i (pour gauche et droite) composées chacune de 28 bits :

$$G_i = \begin{pmatrix} 57 & 49 & 41 & 33 & 25 & 17 & 9 \\ 1 & 58 & 50 & 42 & 34 & 26 & 18 \\ 10 & 2 & 59 & 51 & 43 & 35 & 27 \\ 19 & 11 & 3 & 60 & 52 & 44 & 36 \end{pmatrix} \quad D_i = \begin{pmatrix} 63 & 55 & 47 & 39 & 31 & 23 & 15 \\ 7 & 62 & 54 & 46 & 38 & 30 & 22 \\ 14 & 6 & 61 & 53 & 45 & 37 & 29 \\ 21 & 13 & 5 & 28 & 20 & 12 & 4 \end{pmatrix}$$

- Ces deux blocs subissent ensuite une rotation à gauche. Le nombre de bits de décalage dépend du numéro du cycle, c'est pour cela que la clé est à chaque fois différente. Les décalages effectués sont, dans l'ordre des rondes : 1, 2, 4, 6, 8, 10, 12, 14, 15, 17, 19, 21, 23, 25, 27, 28.



Confidentialité des messages
- Chiffrement symétrique
- Chiffrement asymétrique
- Chiffrement mixte ou hybride



Algorithmes de chiffrement symétrique

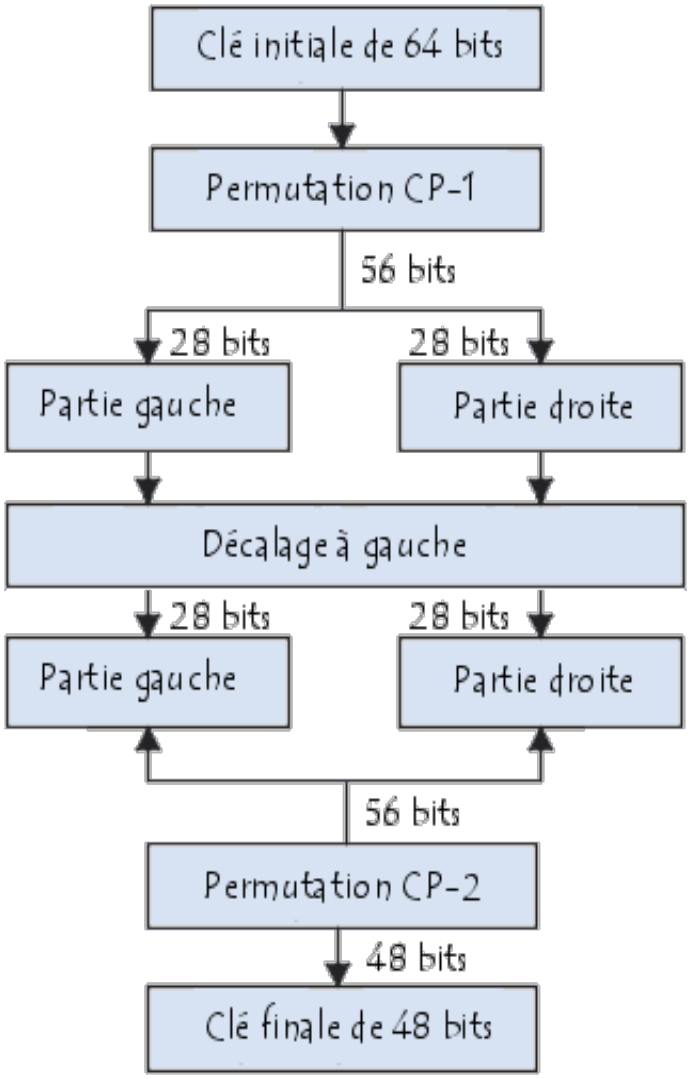
DES (Data Encryption Standard)

Génération des clés

- Les 2 blocs de 28 bits sont ensuite regroupés en un bloc de 56 bits. Celui-ci passe par une permutation, notée CP^{-2}, fournissant en sortie un bloc de 48 bits, représentant la clé K_i.

$$CP-2 = \begin{pmatrix} 14 & 17 & 11 & 24 & 1 & 5 & 3 & 28 & 15 & 6 & 21 & 10 \\ 23 & 19 & 12 & 4 & 26 & 8 & 16 & 7 & 27 & 20 & 13 & 2 \\ 41 & 52 & 31 & 37 & 47 & 55 & 30 & 40 & 51 & 45 & 33 & 48 \\ 44 & 49 & 39 & 56 & 34 & 53 & 46 & 42 & 50 & 36 & 29 & 32 \end{pmatrix}$$

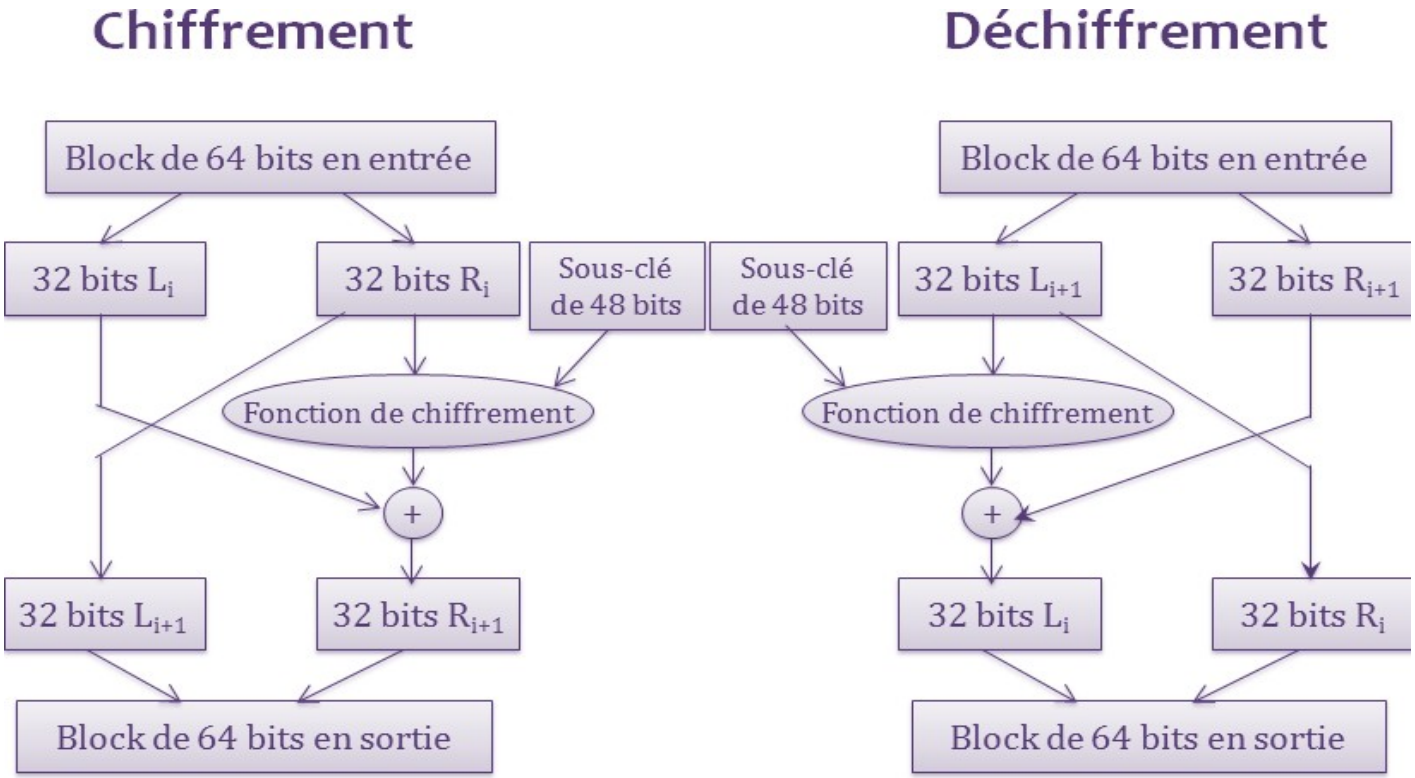
- Des itérations de l'algorithme permettent de donner les 16 clés K_1 à K_{16} utilisées dans l'algorithme du DES.



Algorithmes de chiffrement symétrique

DES (Data Encryption Standard)

Déchiffrement DES



Algorithmes de chiffrement symétrique

DES (Data Encryption Standard)

Problèmes du DES

- Taille de la clé (recherche exhaustive en 256 est réaliste) ;
- Taille du bloc (attaques avec 232 messages) ;
- Cryptanalyse différentielle et linéaire ;
- D'autres attaques (DaviesMurphy, bilinear ...)

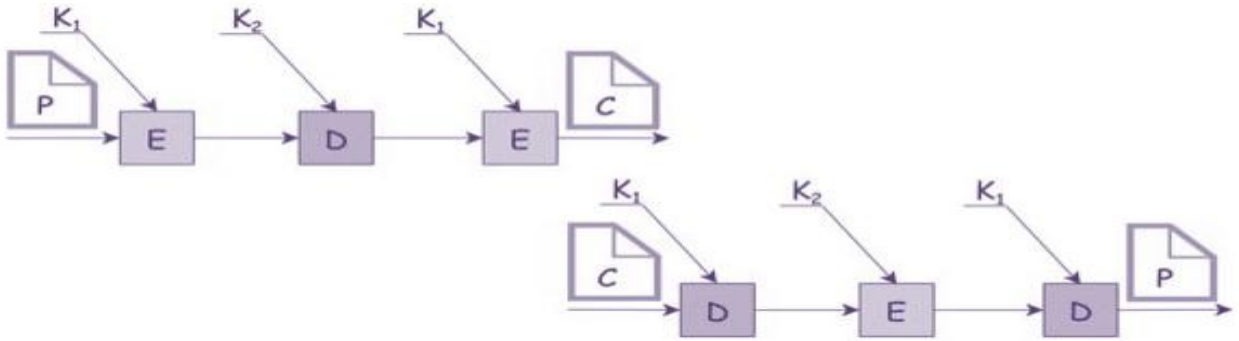
=> Passage au Triple-DES (TDES)

N.B : Malgré tout, le DES est un algorithme très bien conçu : il a plutôt bien résisté à 30 ans de cryptanalyse.

Algorithmes de chiffrement symétrique

TDES (Triple Data Encryption Standard)

- Deux clés et trois étages :
  - ✓ Dans un premier étage, le texte en clair est codé avec DES avec la clé K_1 ;
  - ✓ Dans le deuxième étage, DES est exécuté en mode déchiffrement avec la clé K_2 ;
  - ✓ Dans le dernier étage, DES est exécuté en mode chiffrement en utilisant la clé K_1 ;



- Le TDES permet d'augmenter significativement la sécurité du DES.
- Toutefois, il a l'inconvénient majeur de demander également plus de ressources pour le chiffrement et le déchiffrement.

Algorithmes de chiffrement symétrique

TDES (Triple Data Encryption Standard)

- 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).

- Problèmes du TDES :
  - ✓ Le TDES permet d'éviter les problèmes liés à la taille de clé trop courte du DES, mais le problème de la taille du bloc subsiste.
  - ✓ Le TDES n'est pas très rapide.

Algorithmes de chiffrement symétrique

AES (Advanced Encryption Standard)

Septembre 1997	Appel d'offre lancé par le NIST (National Institute of Standards and Technology) pour un nouveau standard qui serait appelé AES (Advanced Encryption Standard) – L'algorithme devait être un chiffre symétrique par blocs – La totalité de sa conception devait être publique – Il devait être possible d'y utiliser des clés de 128, 192 et 256 bits – Il devait permettre des implémentations matérielles et logicielles – L'algorithme devait être public (utilisation devait être autorisée de façon non discriminatoire)
Août 1998	15 candidats retenus lors de la première conférence AES
Mars 1999	Seconde conférence AES
Avril 1999	Le NIST annonce les 5 finalistes : – Rijndael (Joan Daemen et Vincent Rijmen) – Serpent (Ross Andersen, Eli Biham et Lars Knudsen) – Twofish (équipe dirigée par Bruce Schneier) – RC6 (RSA Laboratories) – MARS (IBM)
Avril 2000	Troisième conférence AES
Octobre 2000	Le vainqueur est l'algorithme Rijndael (2 octobre 2000)
Mai 2002	L'AES remplace le DES

Algorithmes de chiffrement symétrique

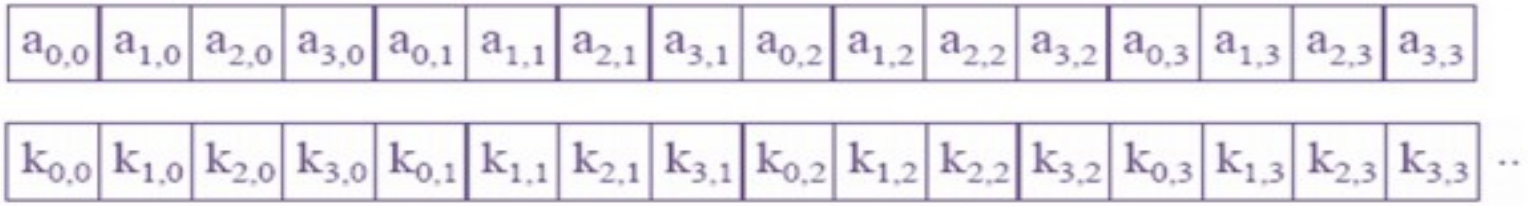
AES (Advanced Encryption Standard)

- Le texte à chiffrer est subdivisé en blocs de 128 bits (16 octets).
- Les clés secrètes ont au choix suivant la version du crypto-système : 128 bits (16 octets), 192 bits (24 octets), ou 256 bits (32 octets).
- Les données et les clés sont découpées en octets et sont placées dans deux tables d'état :
  - ✓ Table d'état du texte : ayant 4 lignes et 4 colonnes.
  - ✓ Table d'état de la clé : ayant 4 lignes et N_k colonnes ($N_k = 4$ (clé de 128 bits), $N_k = 6$ (clé de 192 bits), ou $N_k = 8$ (clé de 256 bits)).

Algorithmes de chiffrement symétrique

AES (Advanced Encryption Standard)

- Une colonne de la table d'état du texte ou de la clé correspond à un mot de 32 bits. Ainsi, chaque petit bloc représente 1 octet. L'input et l'output sont donc gérés comme des séquences linéaires d'octets.



- L'algorithme AES effectue plusieurs tours (rondes) d'une même composition de transformations.
- À chaque ronde, quatre transformations sont appliquées :
  1. Substitution d'octets dans la table d'état (SubBytes)
  2. Décalage de rangées dans la table d'état (ShiftRows)
  3. Déplacement de colonnes dans la table d'état sauf à la dernière ronde (MixColumns)
  4. Addition d'une clé de ronde qui varie à chaque ronde (AddRoundKey)

Algorithmes de chiffrement symétrique

AES (Advanced Encryption Standard)

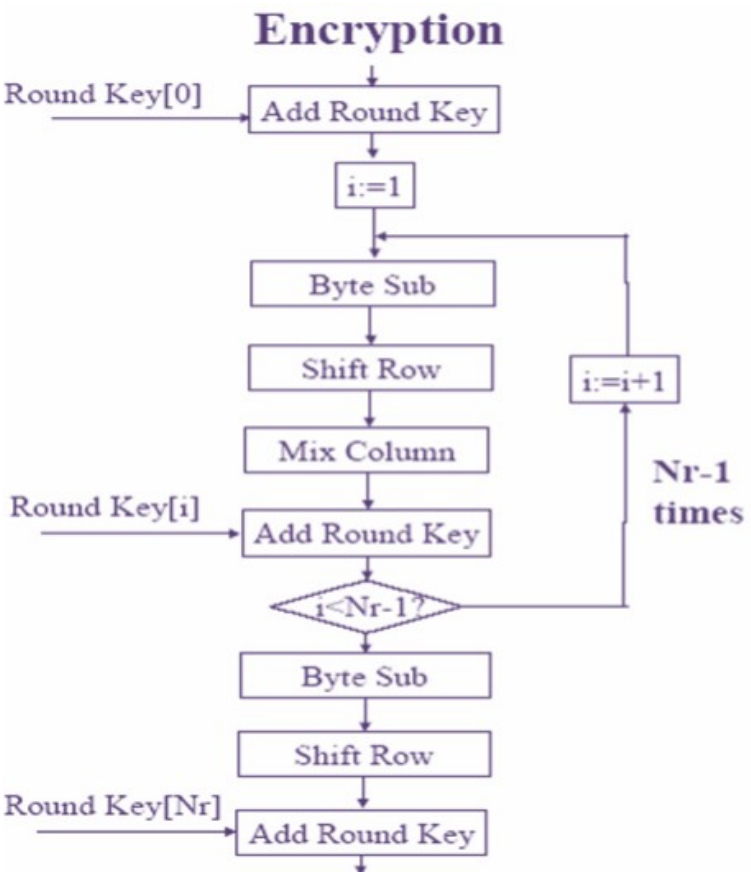
- Suivant la version (la taille de la clé), le nombre de tours n_r à effectuer est différent.

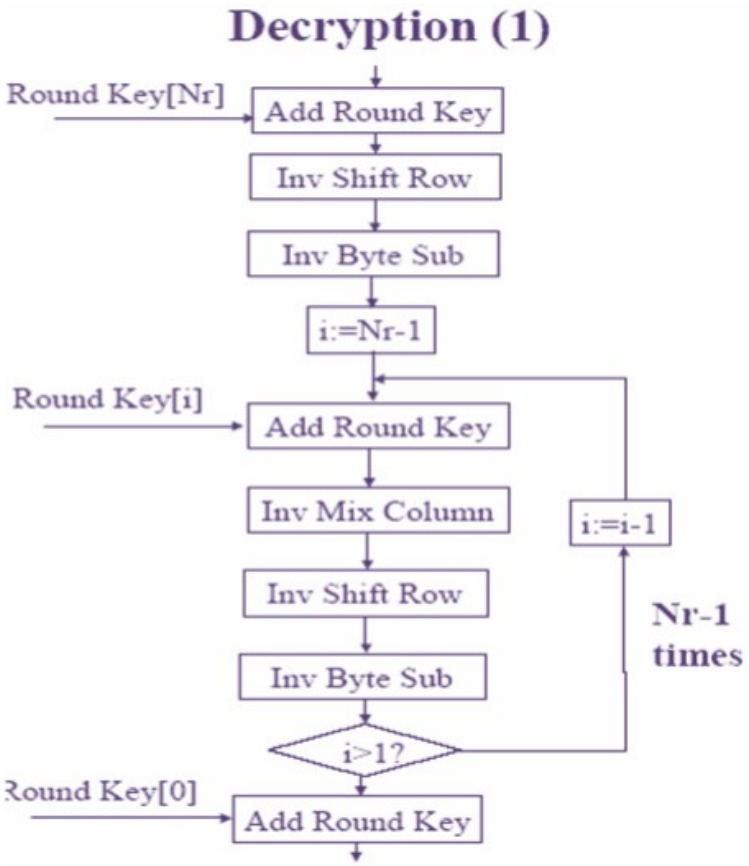
N_k	4	6	8
n_r	10	12	14

- À partir de la clé initiale K, le système crée (n_r +1) clés de tours (rondes) ayant chacune 16 octets.

Algorithmes de chiffrement symétrique

AES (Advanced Encryption Standard)



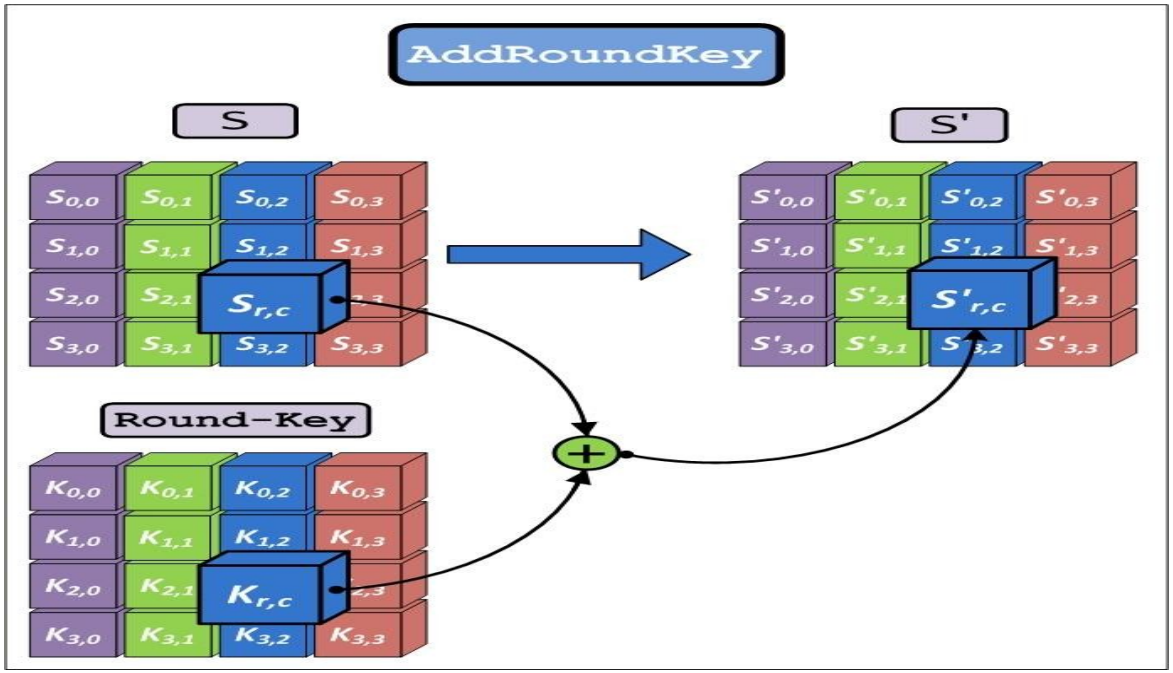


Algorithmes de chiffrement symétrique

AES (Advanced Encryption Standard)

AddRoundKey

La procédure AddRoundKey consiste à faire un ou exclusif (XOR) entre les 128 bits des données à chiffrer et les 128 bits de la clé de tour T_r.



Algorithmes de chiffrement symétrique

AES (Advanced Encryption Standard)

SubBytes

- Il s'agit d'une étape de substitution d'octets (bytes) des données selon une matrice S-Box prédéfinie.
- Par exemple, si S_{11}={53}, il sera substitué par la case dans la ligne 5 et la colonne 3, donc par {ed}.

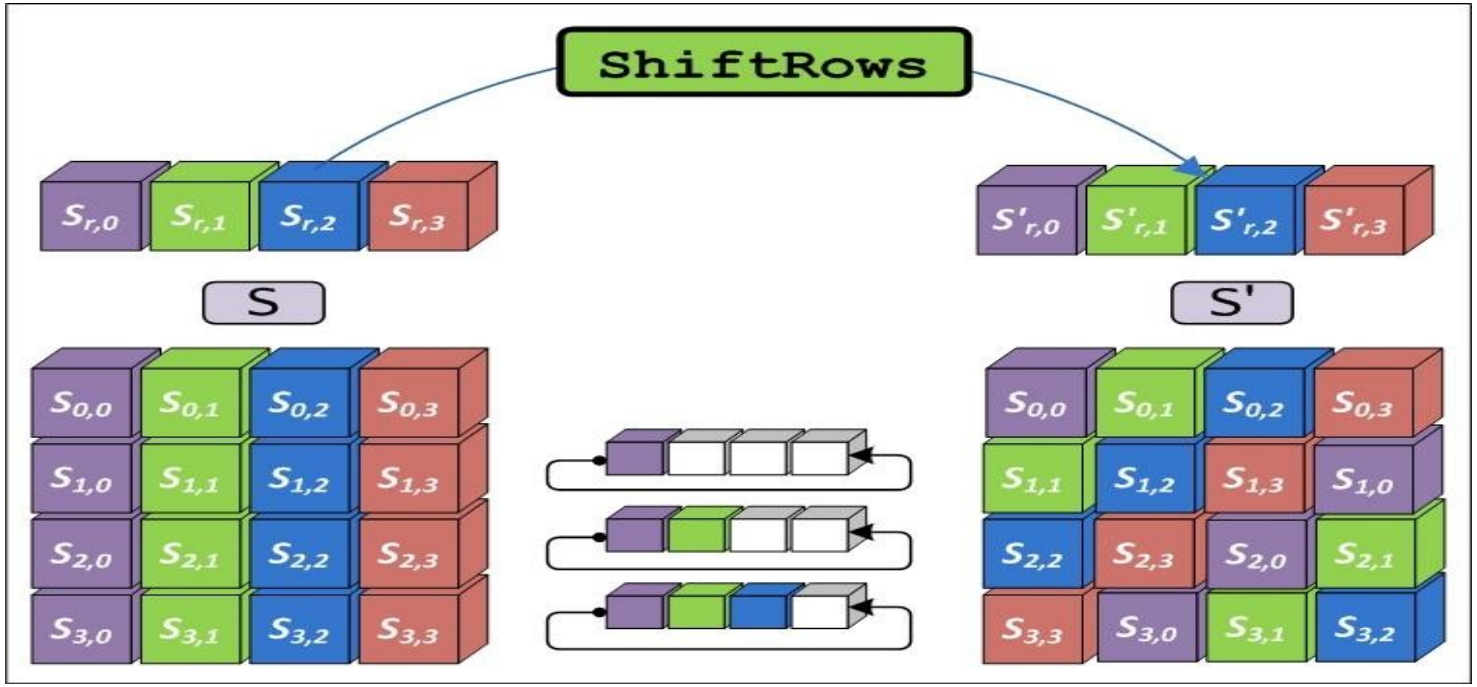
x \ y	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
5	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
8	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
a	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
b	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
d	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
f	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

Algorithmes de chiffrement symétrique

AES (Advanced Encryption Standard)

ShiftRows

La procédure ShiftRows consiste à opérer une rotation à gauche sur chaque ligne du tableau d'entrée. Le nombre de cases dont on décale la ligne i ($0 \leq i \leq 3$) est de i.



Algorithmes de chiffrement symétrique

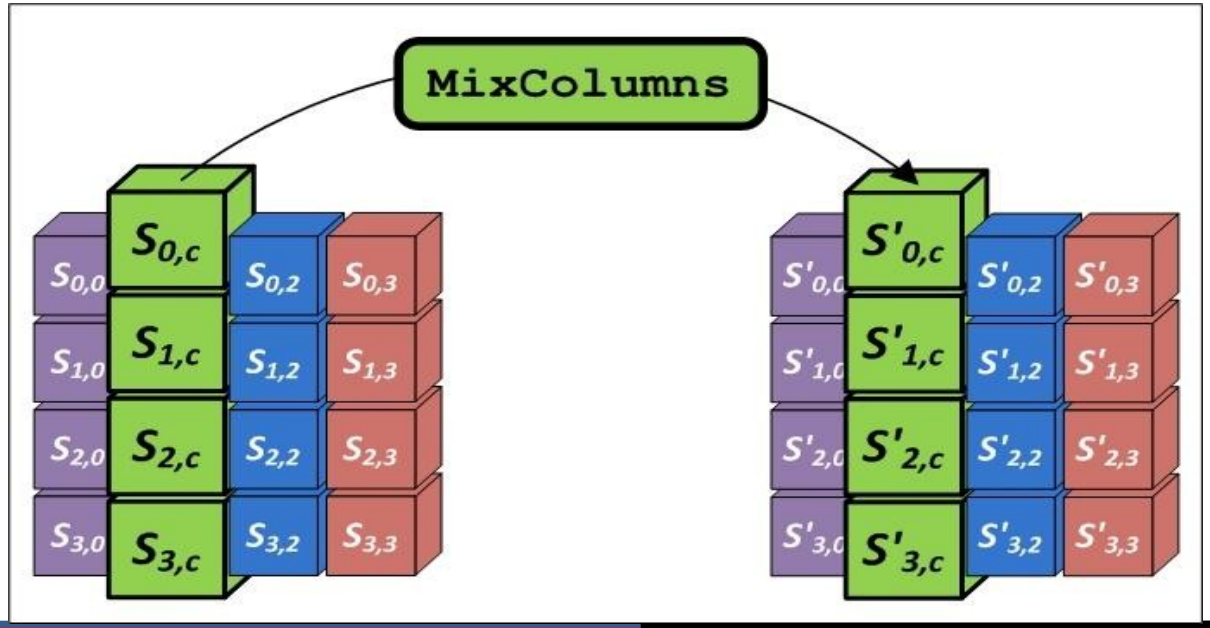
AES (Advanced Encryption Standard)

MixColumns

La transformation MixColums consiste à appliquer à chaque colonne du tableau des données une même transformation linéaire donnée par sa matrice dont les coefficients sont dans le corps fini F_{256} et sont écrits comme des octets en hexadécimal.

$$\begin{bmatrix} s'_{0,c} \\ s'_{1,c} \\ s'_{2,c} \\ s'_{3,c} \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} s_{0,c} \\ s_{1,c} \\ s_{2,c} \\ s_{3,c} \end{bmatrix}$$

pour $0 \le c \le 3$



Algorithmes de chiffrement symétrique

AES (Advanced Encryption Standard)

Génération des clés de tours

- La clé secrète initiale K subit une extension (Key Expansion) => clé étendue et représentée sous forme de table ayant 4 lignes et $4(n_r + 1)$ colonnes.
- La clé étendue sera par la suite découpée en sous-clés appelées clés de rondes ou clés de tours. La clé de la i^{ème} ronde est (une table ayant 4 lignes et 4 colonnes) constituée des 4 colonnes [4i , 4i + 1 , 4i + 2 , 4i + 3] de la table de la clé étendue.

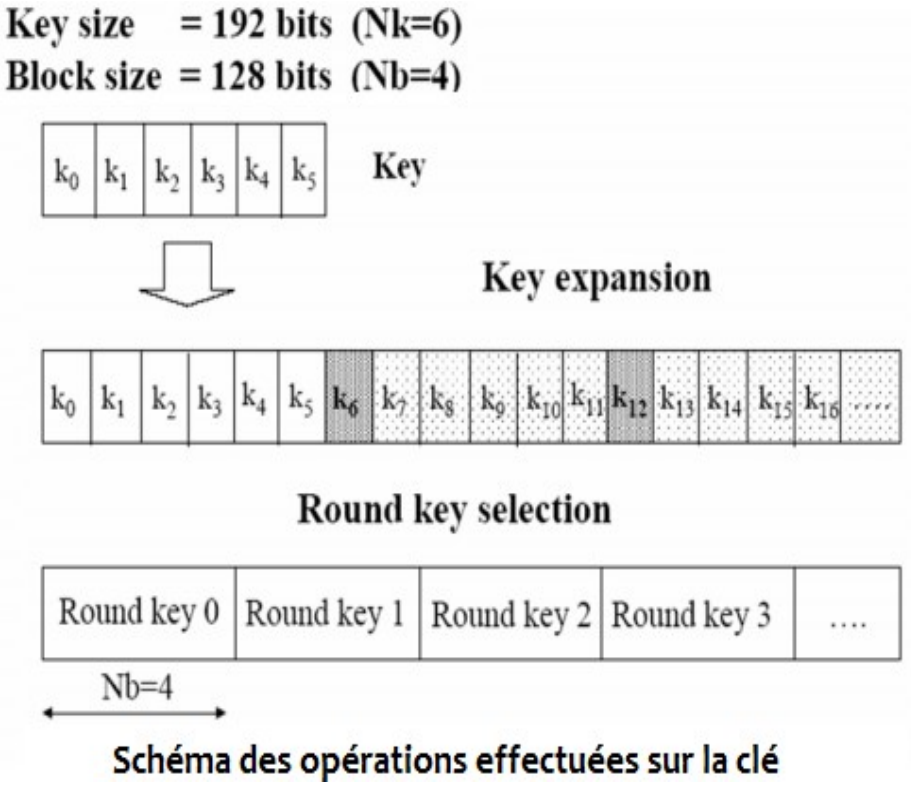


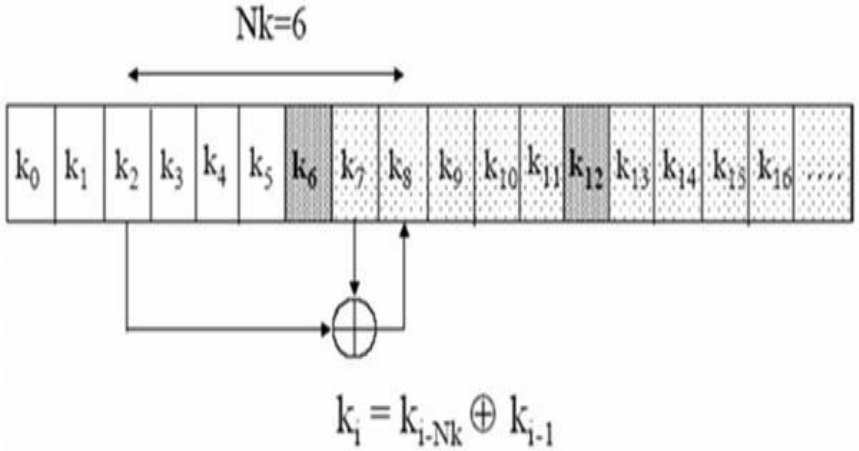
Schéma des opérations effectuées sur la clé

Algorithmes de chiffrement symétrique

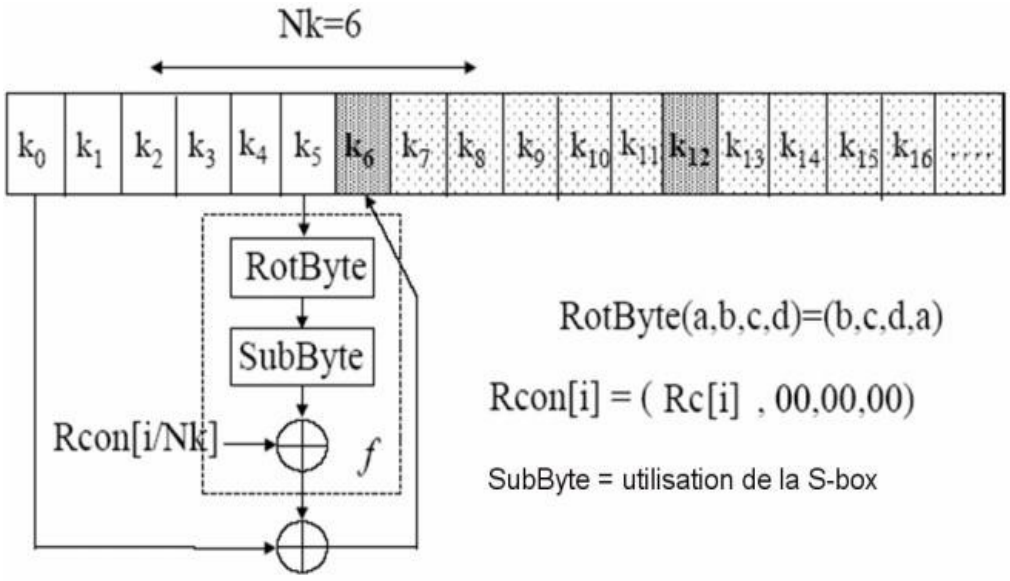
AES (Advanced Encryption Standard)

Extension de la clé

Le calcul de l'expansion de la clé se fait de deux manières distinctes selon le sous-bloc de la clé concernée.



Expansion de la clé avec bloc "commun"



Expansion de la clé avec les blocs "multiples de Nk"

Algorithmes de chiffrement symétrique

AES (Advanced Encryption Standard)

Extension de la clé

- L'ajout de "Rcon[x]" donne comme résultat un $\oplus$ sur les bits les plus significatifs. La table utilisée pour donner les valeurs de Rcon[] est donnée par le tableau suivant :

j	1	2	3	4	5	6	7	8	9	...
RC[j]	01	02	04	08	10	20	40	80	1B	...

Table de correspondance des Rcon[]

- La règle de construction de cette table est : $RC[1] = 1$

$RC[j] = 2.RC[j-1]$

Algorithmes de chiffrement symétrique

AES (Advanced Encryption Standard)

Avantages et limites de l'algorithme AES

- Les principaux avantages sont :
  - Des performances très élevées ;
  - La possibilité de réalisation en "Smart Card" avec peu de code ;
  - Il ne comprend pas d'opérations arithmétiques complexes : ce sont uniquement des décalages et des XOR => rapidité d'exécution ;
  - Il n'utilise pas de composants d'autres crypto-systèmes ;
  - Il n'est pas fondé sur des relations obscures entres opérations ;
  - Le nombre de rondes peut facilement être augmenté si c'est requis ;
  - Il ne possède pas de clés faibles ;
  - Il est résistant à la cryptanalyse différentielle et linéaire.

Algorithmes de chiffrement symétrique

AES (Advanced Encryption Standard)

Avantages et limites de l'algorithme AES

- Il possède pourtant quelques inconvénients et limites :
  - Le code et les tables sont différents pour le chiffrement et le déchiffrement ;
  - Le déchiffrement est plus difficile à implanter en "Smart Card" ;
  - Dans une réalisation matérielle, il y a peu de réutilisation des circuits de chiffrement pour effectuer le déchiffrement.

- **En date d'aujourd'hui, il n'existe pas d'attaque possible et pratique contre l'AES. Ainsi, l'AES demeure le standard de cryptage préféré pour les gouvernements, les banques et les systèmes de haute sécurité autour du monde.**

Plan du chapitre 4 (1/2)

Cryptographie

1. Introduction
2. La cryptographie
   - Confidentialité des messages
     - Chiffrement symétrique
     - Chiffrement asymétrique
     - Chiffrement mixte ou hybride
   - Authentification, Non-répudiation et Intégrité des messages
     - Fonctions de hachage
     - MAC – Message Authentication Code
     - Signature numérique
   - Gestion de clés
     - Distribution des clés
     - Protocole d'échange de clés de Diffie-Hellman
     - Certificat de clé publique
     - PKI – Public Key Infrastructure
3. La cryptanalyse
   - Attaque sur texte chiffré seul
   - Attaque à texte clair connu
   - Attaque à texte clair choisi
   - Attaque à texte chiffré choisi

Principe

- Contrairement au chiffrement symétrique qui utilise une même clé secrète pour le chiffrement et le déchiffrement des messages, le chiffrement asymétrique repose sur un système de pair de clés :
  - La clé publique, librement diffusée 
  - La clé privée, connue de son seul propriétaire 
- Il est impossible de déduire la clé privée à partir de la clé publique.
- Les deux clés s'annulent mutuellement.

Scénario d'utilisation : Confidentialité

Pour être sûr que seul Bob puisse lire le message :

Scénario d'utilisation : Authenticité de l'émetteur et non répudiation d'envoi

Pour être sûr que c'est bien Alice qui a envoyé le message :

Scénario d'utilisation : Confidentialité + Authenticité de l'émetteur

Double chiffrement puis double déchiffrement :

Avantages et inconvénients

Avantages

- ✓ L'élimination de la problématique de la transmission de clé ;
- ✓ Aucun canal sécurisé n'est nécessaire pour pratiquer l'échange de la clé publique ;
- ✓ L'impossibilité de décrypter le message dans le cas de son interception par une personne non autorisée ;
- ✓ Le chiffrement asymétrique crée aussi moins de problèmes de gestion de clés que le chiffrement symétrique : 2n clés seulement sont nécessaires pour que n entités communiquent en toute sécurité entre elles.

Inconvénients

- ✓ L'inconvénient majeur du chiffrement asymétrique par rapport au chiffrement symétrique est qu'il tend à être environ « 1000 fois plus lents » ;
- ✓ Le danger des attaques par substitution des clés (d'où la nécessité de valider les émetteurs des clés).

Algorithmes de chiffrement asymétrique

- Exemples :
  - RSA
  - ElGamal
  - ECC
  - …

- Les principales catégories d'algorithmes de chiffrement à clé publique :
  - Ceux qui se fondent sur la difficulté de factoriser de grands nombres ;
  - Ceux qui calculent des logarithmes discrets modulo un grand nombre premier ;
  - Ceux qui reposent sur des courbes elliptiques ;

- Tous ces problèmes sont considérés comme nativement NP-difficiles.

Algorithmes de chiffrement asymétrique

RSA (Rivest, Shamir, Adelman)

- Le premier algorithme de chiffrement à clé publique (chiffrement asymétrique) a été développé par R. Merckle et M. Hellman en 1977 ;
- Il fut vite rendu obsolète grâce aux travaux de Shamir, Zippel et Herlestman, de célèbres cryptanalistes ;
- L'algorithme à clé publique de Rivest, Shamir, et Adelman (d'où son nom RSA) apparaît en 1978 ;
- RSA servait encore en 2002 à protéger les codes nucléaires de l'armée américaine et russe ;
- Le RSA, proposé en 1978, utilisant des clés de chiffrement de 128 bits, n'a été cassé qu'en 1996 en faisant travailler de nombreux ordinateurs en parallèle ;
- RSA est considéré toujours comme sûr avec la technologie actuelle pour des clés suffisamment longues (1024, 2048 voire 4096 bits) ⇒ demande un temps de calcul assez important ;
- C'est le système le plus largement répandu (carte bancaire, protocole SSL, . . .);

Algorithmes de chiffrement asymétrique

RSA (Rivest, Shamir, Adelman)

- L'algorithme RSA repose sur les principes de la théorie des nombres ;
- Bob désire envoyer un message crypté à Alice en utilisant la méthode RSA :

1ère étape : Constitution des clés

1. Bob choisit deux grands nombres premiers (distincts et de taille à peu près égale) p et q ;
2. Bob calcule :
   - Le module de chiffrement : n = pq ; (La longueur de n devrait être entre 1024 et 2048 bits)
   - L'indicatrice d'Euler : $\varphi(n) = (p-1)(q-1)$;
3. Bob choisit l'exposant de chiffrement e, un entier premier avec $\varphi(n)$ (càd pgcd(e, $\varphi(n)$)=1).
4. Bob calcule l'exposant de déchiffrement d défini comme étant l'inverse mod $\varphi(n)$ de e. En d'autres termes, $ed \equiv 1 \bmod \varphi(n)$ (théorème de Bachet-Bézout).
5. Le couple (e,n) constitue la clé publique => utilisée pour le chiffrement.
   Le couple (d,n) constitue la clé privée => utilisée pour le déchiffrement.
6. Bob diffuse donc n et e, garde secret d et oublie $\varphi(n)$.

Algorithmes de chiffrement asymétrique

RSA (Rivest, Shamir, Adelman)

2ème étape : Chiffrement et déchiffrement du message

- Le texte en clair à chiffrer (considéré comme une chaîne de bits) est divisé en blocs de m bits chacun ; m étant le plus grand entier pour lequel la condition $0 < m < n$ est vérifiée ;
  - ✓ Pour chiffrer un message P, on calcule $C \equiv P^e [n]$;
  - ✓ Pour déchiffrer C, il s'agit de calculer $P \equiv C^d [n]$;
- On montre que pour tout P appartenant à l'intervalle $0 \leq P < n$, les fonctions de chiffrement et de déchiffrement sont inverses ;

Algorithmes de chiffrement asymétrique

RSA (Rivest, Shamir, Adelman)

Exemple

- Prenons 2 nombres premiers au hasard: **p = 29, q = 37**
- On calcule **n = pq = 29 * 37 = 1073**
- On doit choisir **e** au hasard tel que **e** soit premier avec **(p-1)(q-1) = (29-1)(37-1) = 1008**
- On prend **e = 71**
- On choisit **d** tel que **71*d mod 1008 = 1**
- On trouve **d = 71**
- On a maintenant nos clés :
  - ✓ La clé publique est **(e,n) = (71,1073)** (= clé de chiffrement)
  - ✓ La clé privée est **(d,n) = (71,1073)** (= clé de déchiffrement)

Algorithmes de chiffrement asymétrique

RSA (Rivest, Shamir, Adelman)

Exemple 1

- On va chiffrer le message '**HELLO**'. On va prendre le code ASCII de chaque caractère et on les met bout à bout:

P = 7269767679

- Ensuite, il faut découper le message en blocs qui comportent moins de chiffres que **n**. **n** comporte 4 chiffres, on va donc découper notre message en blocs de 3 chiffres:

726 976 767 900 (on complète avec des zéros)

- Ensuite on chiffre chacun de ces blocs en utilisant la méthode « **modulo exponentiation** »:

$C=726^{71} \bmod 1073$
$=726*(726^2)^{35} \bmod 1073$
$=726*(233 \bmod 1073)^{35} \bmod 1073$
$=726*233*(233^2)^{17} \bmod 1073$
$=726*233*639^{17} \bmod 1073$
$=726*233*639*(639^2)^8 \bmod 1073$
$=726*233*639*581^8 \bmod 1073$
$=726*233*639*(581^2)^4 \bmod 1073$
$=726*233*639*(639^2)^2 \bmod 1073$
$=726*233*639*581^2 \bmod 1073$
$=726*233*639*639 \bmod 1073 = 436 \bmod 1073$

Algorithmes de chiffrement asymétrique

RSA (Rivest, Shamir, Adelman)

Exemple 1

Ainsi, on obtient :

726^71 mod 1073 = 436
976^71 mod 1073 = 822
767^71 mod 1073 = 825
900^71 mod 1073 = 552

- Le message chiffré est **436 822 825 552.** On peut le décrypter avec d :

436^71 mod 1073 = 726
822^71 mod 1073 = 976
825^71 mod 1073 = 767
552^71 mod 1073 = 900

- C'est à dire la suite de chiffre **726976767900.** (code ASCII : 0 -> 127)
  On retrouve notre message en clair **72 69 76 76 79 : 'HELLO'.**

Algorithmes de chiffrement asymétrique

RSA (Rivest, Shamir, Adelman)

Exemple 2

- Saddam souhaiterait envoyer le message suivant à George : « Kisses from Iraq ». Malheureusement, Vladimir les espionne, et pourrait intercepter ce message. Nos deux compères vont donc chiffrer leurs échanges avec la méthode RSA.

- **Étape 1**

George a choisi **p = 37** et **q = 43**. Il en déduit **n = 37 × 43 = 1591**, et **φ(n) = 36 × 42 =1512**.

Il choisit ensuite e = 19, qui est premier avec 1512. L'inverse de 19 modulo 1512 est **d=955**.

George peut donc maintenant publier sa clé publique **(19,1591)** par exemple sur son site internet, et garde secrète sa clé privée **(955,1591)**.

- **Étape 2**

Saddam va utiliser la clé pour chiffrer son message. Comme Saddam veut envoyer le message sous forme d'un fichier informatique, le mieux est d'utiliser le code ASCII.

Algorithmes de chiffrement asymétrique

RSA (Rivest, Shamir, Adelman)

Exemple 2

K	i	s	s	e	s		f	r	o	m		l	r	a	q
75	105	115	115	101	115	32	102	114	111	109	32	43	114	97	113

- **Étape 3**

Il suffit à Saddam de coder chaque nombre comme expliqué en utilisant la méthode «**modulo exponentiation** » . Par exemple : **75^{19} mod 1591 = 371.**

$C=75^{19} \bmod 1591$
$=75*75^{18} \bmod 1591$
$=75*(75^2)^9 \bmod 1591$
$=75*(5625)^9 \bmod 1591$

$=75*(852 \bmod 1591)^9 \bmod 1591$
$=75*(852)^9 \bmod 1591$
$=75*852*(859^2)^4 \bmod 1591$
$=75*852 \bmod 1591 = 371$

75	105	115	115	101	115	32	102	114	111	109	32	43	114	97	113
371	1338	1410	1410	1174	1410	930	1397	632	703	483	930	1405	632	532	1441

Algorithmes de chiffrement asymétrique

RSA (Rivest, Shamir, Adelman)

Exemple 2

- **Étape 4**

Saddam envoie cette suite de nombres à George, qui va la déchiffrer avec sa clé d. Il va pouvoir retrouver le message original : **$371^{955} \bmod 1591 = 75$**

371	1338	1410	1410	1174	1410	930	1397	632	703	483	930	1405	632	532	1441
75	105	115	115	101	115	32	102	114	111	109	32	43	114	97	113
K	i	s	s	e	s		f	r	o	m		l	r	a	q

Algorithmes de chiffrement asymétrique

RSA (Rivest, Shamir, Adelman)

Sécurité du RSA

- La sécurité de RSA repose sur la difficulté de factoriser des grands nombres ;
- Si le cryptanalyste pouvait factoriser n qui est public, il pouvait retrouver p et q puis $\varphi(n)$; Connaissant $\varphi(n)$ et e, il est ainsi possible de déterminer d au moyen de l'algorithme d'Euclide ;
- Par bonheur, les mathématiciens essayent, depuis des années, de factoriser des grands nombres mais ils n'y arrivent pas => le problème est excessivement difficile ;
- Selon Rivest et ses collègues, factoriser un nombre de cinq cents bits demande 10^{25} années au moyen d'une méthode d'essais exhaustifs ;
- Il s'écoulera des siècles avant qu'il ne devienne possible de factoriser un nombre de cinq cents bits => à ce moment, il suffira de choisir des valeurs plus grandes de p et q;

Algorithmes de chiffrement asymétrique

RSA (Rivest, Shamir, Adelman)

Limites du RSA

- p et q doivent être très grands. On estime qu'il faut moins d'une seconde pour casser un code à base de nombre de 32 bits ;
- Il faut utiliser des clés suffisamment longues (1024, 2048 voire 4096 bits) ;
- Cas d'un modulo commun ;

Cas d'un modulo commun

- **Le modulo doit être spécifique à chaque personne.**
- Supposons qu'Alice et Bob utilisent le même modulo.
- *Si un même message est chiffré à l'intention d'Alice et de Bob, il devient possible pour tout le monde de le déchiffrer !*
- Le message P est chiffré avec les exposants e_A et e_B, supposés premiers entre eux : $ue_A + ve_B = 1$ (théorème de Bezout)
- Soient $C_A = P^{eA}$ et $C_B = P^{eB}$ les chiffrés
- N'importe qui peut ainsi calculer P : $C_A^{\,u} * C_B^{\,v} = P^{eA*u} * P^{eB*v} = P$

Conclusion : Ne jamais partager un modulo RSA à plusieurs

Algorithmes de chiffrement asymétrique

RSA (Rivest, Shamir, Adelman)

Conseils d'utilisation du RSA

- Ne jamais utiliser de trop petites valeurs pour n ;
- N'utiliser que des clés fortes (p-1 et q-1 ont un grand facteur premier) ;
- Ne pas chiffrer des blocs trop courts ;
- Ne pas utiliser de n communs à plusieurs clés (cas du modulo commun) ;
- Si (d,n) est compromise ne plus utiliser n.

Algorithmes de chiffrement asymétrique

ELGamal

- Un algorithme de chiffrement à clé publique inventé par Taher ElGamal en 1984 ;
- Il est basé sur la difficulté de calculer des logarithmes discrets ;
- ElGamel est 2 fois plus lent que le RSA ;
- Son inconvénient majeur reste la taille des données chiffrées qui représente 2 fois celle des données en clair ;
- La recherche de la clé privée à partir de la clé publique est équivalente au problème du logarithme discret qui est un problème NP-difficile. Ainsi, si le problème du logarithme discret est résolu polynomialement alors ELGamal sera cassé.
- Cependant, rien ne prouve qu'il n'est pas cassable par un autre moyen.

Algorithmes de chiffrement asymétrique

ELGamal

Principe du chiffrement

- Soit un entier premier **p** très grand et **(p-1)** doit avoir un grand facteur premier. On produit:
  - Une clé secrète **s** telle que **$s \in [1, p-2]$** ;
  - Une clé publique reposant sur l'entier **p**, un entier **a** premier avec **p** et l'entier **P** tel que :

$$P = a^s \bmod p$$

Le nombre **a** est pris tel que **$a \in [0, p-1]$** et **$\nexists\, k \in [1, p-2] : a^k \equiv 1 \bmod p$**

- Soit un message **M**, avec **M < p**. On détermine un nombre aléatoire **k** qui n'est connu que de celui qui chiffre et qui est différent à chaque message. On calcule alors :

$$C_1 = a^k \bmod p \quad \text{et} \quad C_2 = M.P^k \bmod p$$

- Le message chiffré est alors **$C = (C_1, C_2)$**. Il est deux fois plus long que le message original.

Algorithmes de chiffrement asymétrique

ELGamal

Principe du déchiffrement

- A la réception, on calcule :

$$R1 = (C1)^s \bmod p = a^{sk} \bmod p = P^k \bmod p$$

- Le destinataire possède la clé privée **(s)**. Ayant P^k, on divise **C2** par cette valeur :

$$D_k(C) = C2/R1 = M.P^k \bmod p \,/\, P^k \bmod p = M$$

- P^k est donc considéré comme un masque appliqué sous forme multiplicative à **M**.

- Pour décrypter le message, il faudra soit trouver directement un masque jetable, soit trouver la clé privée **s**, solution de $P = a^s \bmod p$ (et donc trouver le logarithme discret).

Algorithmes de chiffrement asymétrique

ELGamal

Exemple

Soient $p = 2579, a = 2, s = 765$. Il vient

- Clé privée $S_k = (765)$
- Clé publique $P_k = (2579, 2, 949)$ car $2^{765} \bmod 2579 = 949$

Pour chiffrer M = 1299, on choisit k = 853. Il vient

$$C_1 = 2^{853} \bmod 2579 = 435$$

$$C_2 = 1299 * 949^{853} \bmod 2579 = 2396$$

On peut effectivement vérifier que $2396/(435^{765}) \bmod 2579 = 1299$.

Algorithmes de chiffrement asymétrique

ECC

- La cryptographie sur les courbes elliptiques (Elliptic Curve Cryptography, ECC) est un concept qui a été proposé en 1985 par deux chercheurs Miller et Klobitz, de façon totalement indépendante.
- Ce type de cryptographie, toujours basé sur le modèle asymétrique permet aussi bien de chiffrer que de signer.
- La théorie sous-jacente, ainsi que l'implémentation sont très complexes, ce qui explique le fait que cette technologie soit moins répandue.
- De par la nécessité de traiter plus rapidement l'information, de gérer des quantités de données importantes tout en consommant moins de ressources et de miniaturiser au maximum, les avantages de cette technique attirent récemment de plus en plus l'attention des chercheurs.
- Les clés utilisées sont plus courtes pour une sécurité égale ou supérieure. Ce qui implique une consommation de mémoire et d'énergie moins importante et un calcul plus rapide (adéquation au domaine des systèmes embarqués)

Algorithmes de chiffrement asymétrique

ECC

Symmetric Key Size (bits)	RSA and Diffie-Hellman Key Size (bits)	Elliptic Curve Key Size (bits)
80	1024	160
112	2048	224
128	3072	256
192	7680	384
256	15360	521

Table 1: NIST Recommended Key Sizes

Algorithmes de chiffrement asymétrique

ECC

Les Courbes Elliptiques (Elliptic Curves, EC)

- Sur ***K***, les courbes elliptiques représentent l'ensemble des couples (x,y) tels que :

$$E = \{(x, y) \mid y^2 = x^3 + ax + b\}$$

avec :

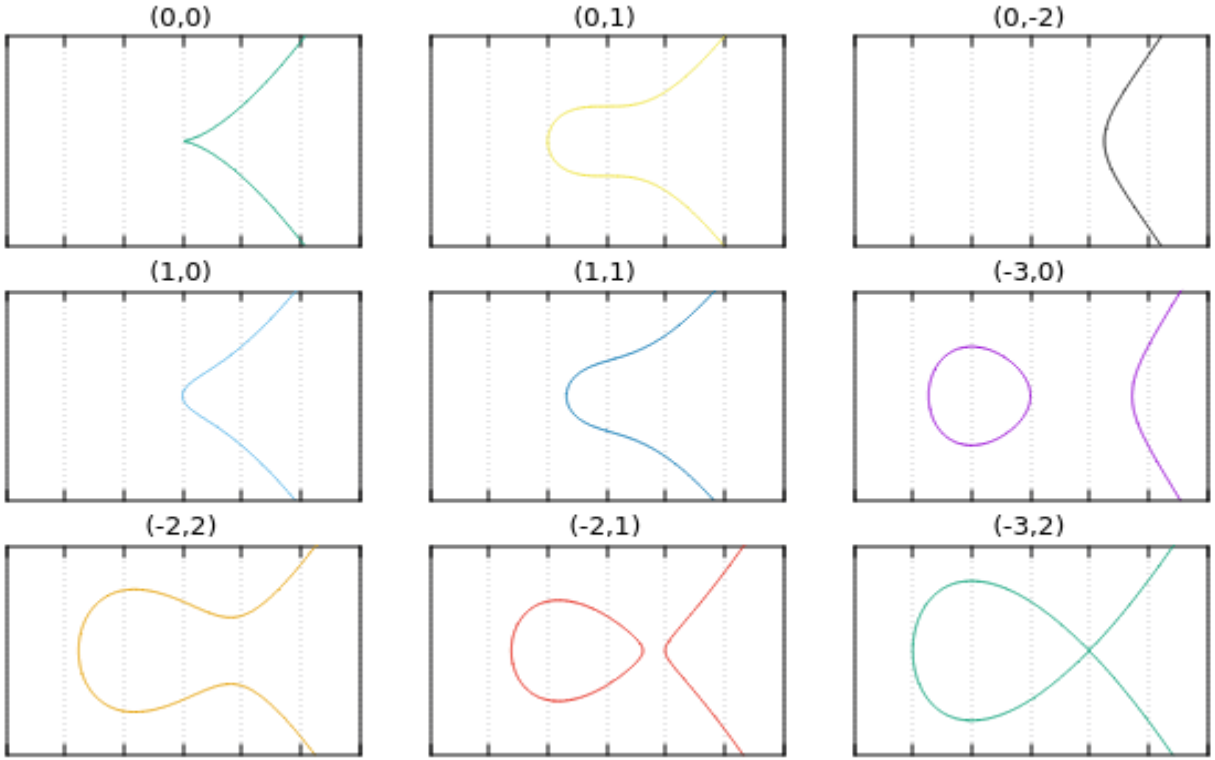
- a , b ∈ ***K***
- Son discriminant est non nul : $4a^3 + 27b^2 \neq 0$
- 3 points alignés sur une EC, leur somme vaut ***O*** (point à l'infini).

- Exemples de corps ***K*** sur lesquels on peut définir des EC : ***R , Q , C , Z/nZ***

Algorithmes de chiffrement asymétrique

ECC

Exemples de courbes elliptiques sur Z/nZ



Algorithmes de chiffrement asymétrique

ECC

Les Courbes Elliptiques (Elliptic Curves, EC)

+ ADDITION

point at infinity

$$E = \{(x, y) \mid y^2 = x^3 + ax + b\} \cup \{\mathcal{O}\}$$

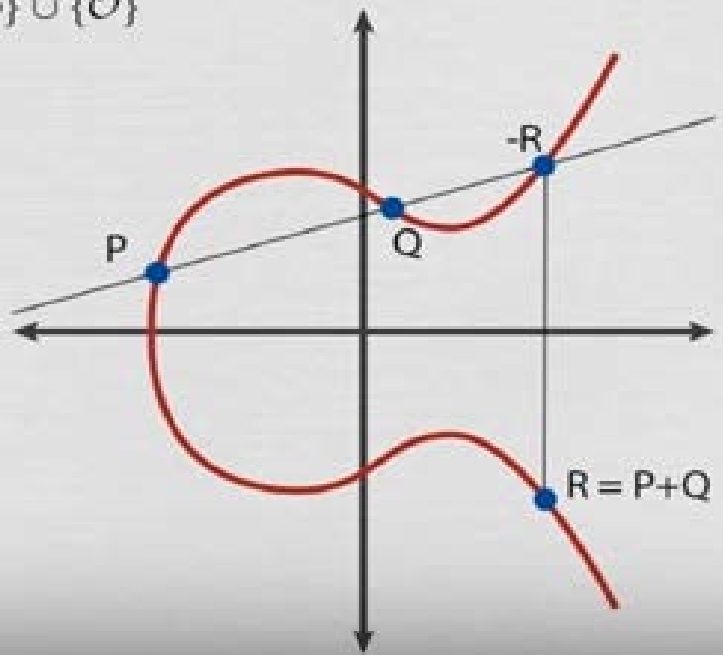
P+Q=?

Algebraically

$$s = \frac{y_P - y_Q}{x_P - x_Q}$$

$$x_R = s^2 - (x_P + x_Q)$$

$$y_R = s(x_P - x_R) - y_P$$



Algorithmes de chiffrement asymétrique

ECC

Les Courbes Elliptiques (Elliptic Curves, EC)

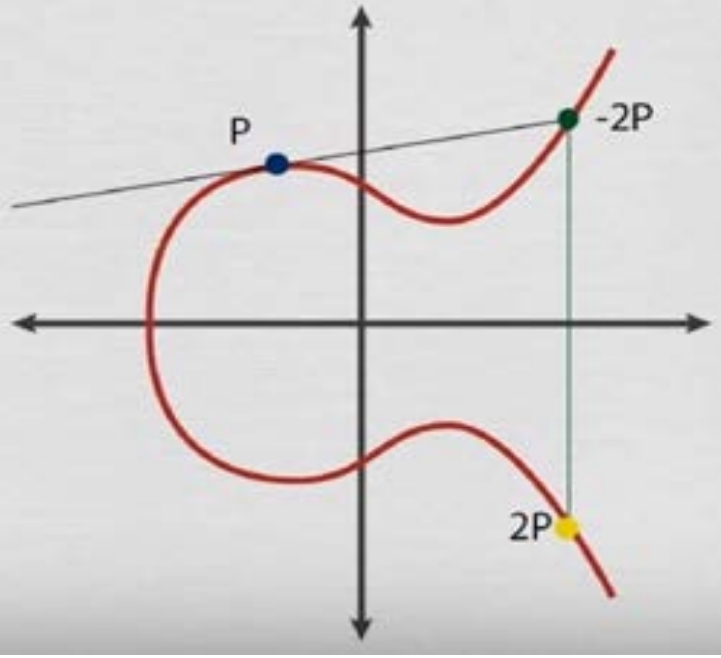
Point Doubling $P + P = R = 2P$

Algebraically

$$s = \frac{3x_P^2 + a}{2y_P}$$

$$x_R = s^2 - 2x_P$$

$$y_R = s(x_P - x_R) - y_P$$



Algorithmes de chiffrement asymétrique

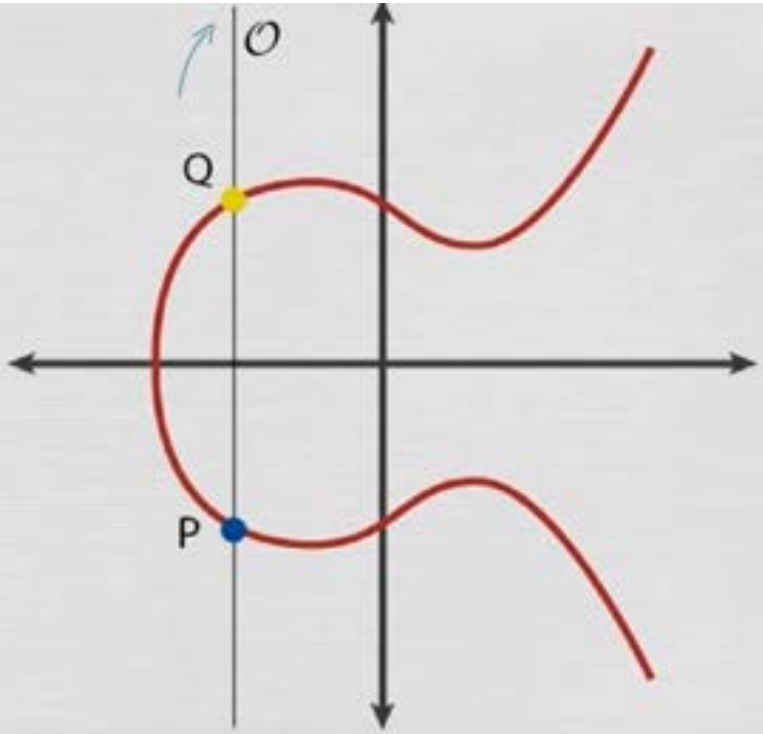
ECC

Les Courbes Elliptiques (Elliptic Curves, EC)

Adding Vertical Points

$P + Q = \mathcal{O}$ if $x_P = x_Q$

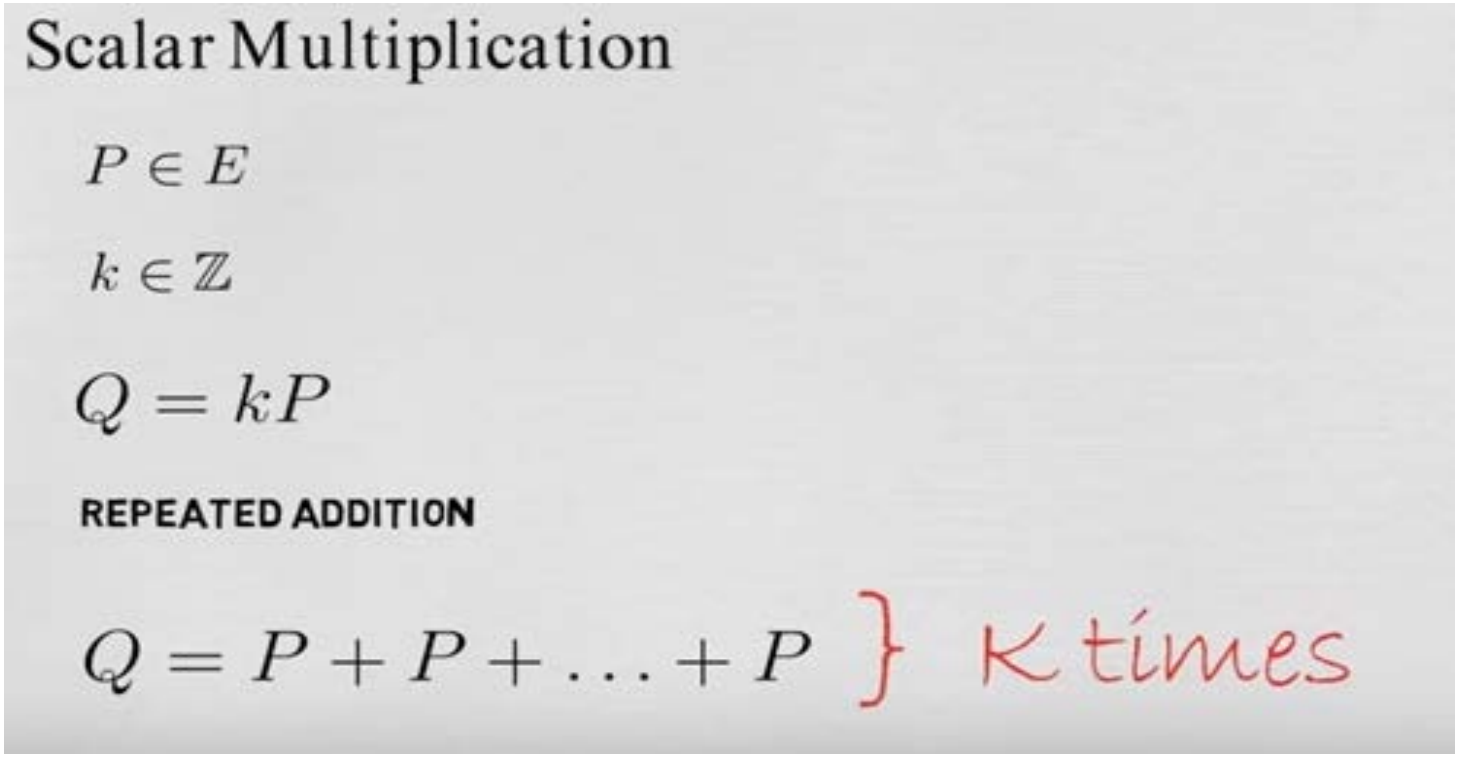
$P + P = \mathcal{O}$ if $x_P = 0$



Algorithmes de chiffrement asymétrique

ECC

Les Courbes Elliptiques (Elliptic Curves, EC)



Algorithmes de chiffrement asymétrique

ECC

La cryptographie sur les Courbes Elliptiques

- Pour utiliser les courbes elliptiques en cryptographie, il faut trouver un problème difficile (tel que la factorisation d'un produit en ses facteurs premiers dans le cas du RSA)
- Considérons l'équation : $Q = k P$ Où $Q, P \in E_p(a,b)$ (une EC sur Z/pZ) et $k < p$
- Il est facile de calculer **Q** connaissant **k** et **P**, mais il est difficile de déterminer **k** si on connait **Q** et **P**. Il s'agit du problème du logarithme discret pour les courbes elliptiques : $\log_P(Q)$.
- Dans une utilisation réelle, le **k** est très grand, rendant l'attaque par force brute inutilisable (rappelons qu'a priori, l'attaque par force brute est toujours possible. . .)

Algorithmes de chiffrement asymétrique

ECC

La cryptographie sur les Courbes Elliptiques

The Base Point (Generator)

$G \in E(\mathbb{Z}/p\mathbb{Z})$ GENERATES A CYCLIC GROUP

$ord(G) = n$ size of subgroup smallest positive integer s.t. $kG = \mathcal{O}$

Cofactor: $h = \frac{|E(\mathbb{Z}/p\mathbb{Z})|}{n}$ ← number of points on the curve

IDEALLY: $h = 1$

Algorithmes de chiffrement asymétrique

ECC

La cryptographie sur les Courbes Elliptiques

Paramètres utilisés

$$\{p, a, b, G, n, h\}$$

p : field (modulo p)

a, b : curve parameters

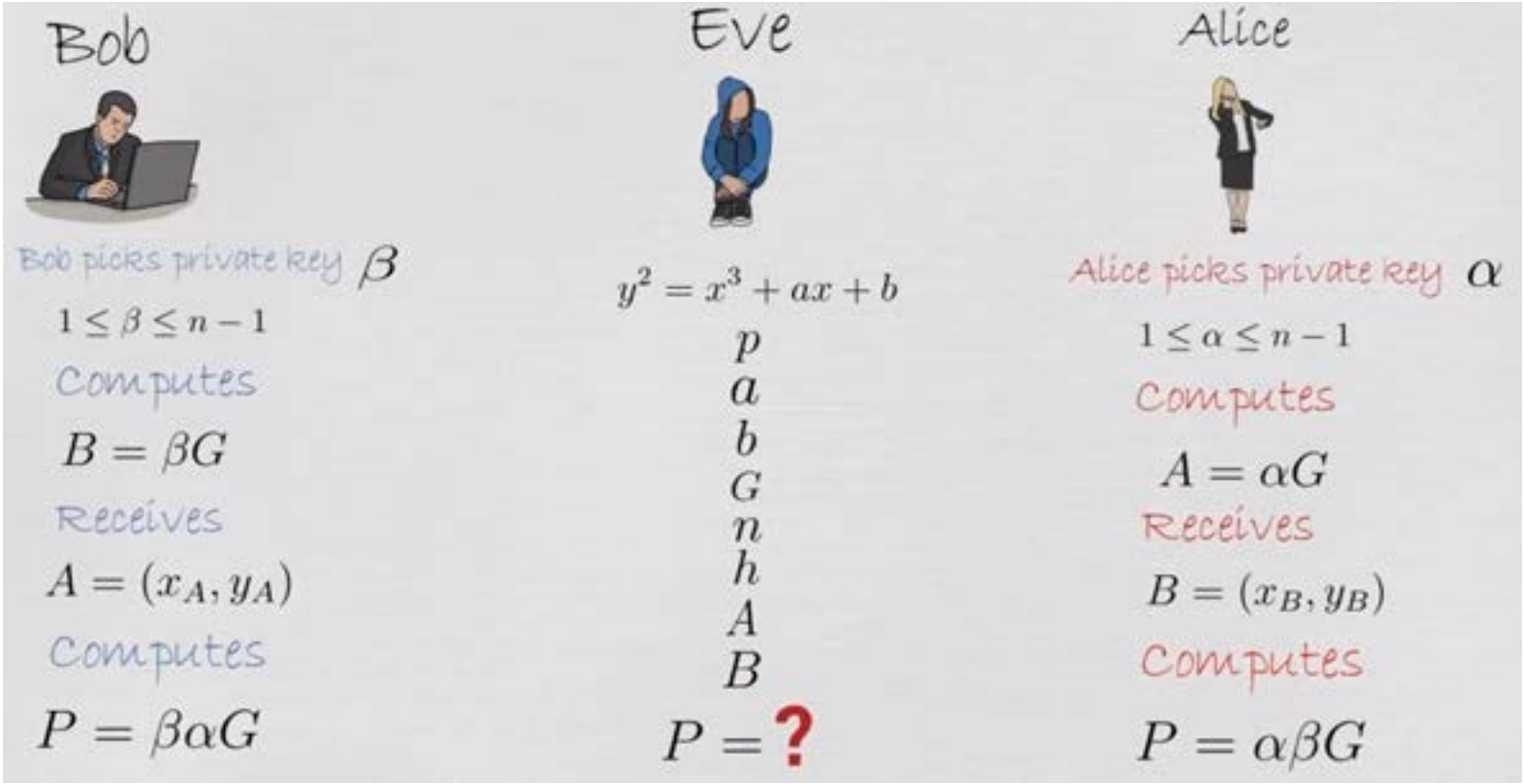
G : Generator Point

n : ord(G)

Algorithmes de chiffrement asymétrique

ECC

ECC pour l'échange de clés



Algorithmes de chiffrement asymétrique

ECC

ECC pour l'échange de clés (Exemple)

$$E : \quad y^2 \equiv x^3 + 2x + 2 \pmod{17}$$

- p = 17
- a = 2
- b = 2
- G (5 , 1)
- n = ?

Algorithmes de chiffrement asymétrique

ECC

ECC pour l'échange de clés (Exemple)

COMPUTE $2G = G + G$

$$s = \frac{3x_G^2 + a}{2y_G} \qquad s \equiv \frac{3(5^2)+2}{2(1)} \equiv 77 \cdot 2^{-1} \equiv 9 \cdot 9 \equiv 13 \pmod{17}$$

$$x_{2G} = s^2 - 2x_G \qquad x_{2G} \equiv 13^2 - 2(5) \equiv 16 - 10 \equiv 6 \pmod{17}$$

$$y_{2G} = s(x_G - x_{2G}) - y_G \qquad y_{2G} \equiv 13(5-6) - 1 \equiv -13 - 1 \equiv -14 \equiv 3 \pmod{17}$$

$$2G = (6, 3)$$

Algorithmes de chiffrement asymétrique

ECC

ECC pour l'échange de clés (Exemple)

$$
\begin{aligned}
G &= (5,1) & 11G &= (13,10)\\
2G &= (6,3) & 12G &= (0,11)\\
3G &= (10,6) & 13G &= (16,4)\\
4G &= (3,1) & 14G &= (9,1)\\
5G &= (9,16) & 15G &= (3,16)\\
6G &= (16,13) & 16G &= (10,11)\\
7G &= (0,6) & 17G &= (6,14)\\
8G &= (13,7) & 18G &= (5,16)\\
9G &= (7,6) & 19G &= \mathcal{O}\\
10G &= (7,11)
\end{aligned}
$$

$n = 19$

$h = 1$

Algorithmes de chiffrement asymétrique

ECC

ECC pour l'échange de clés (Exemple)

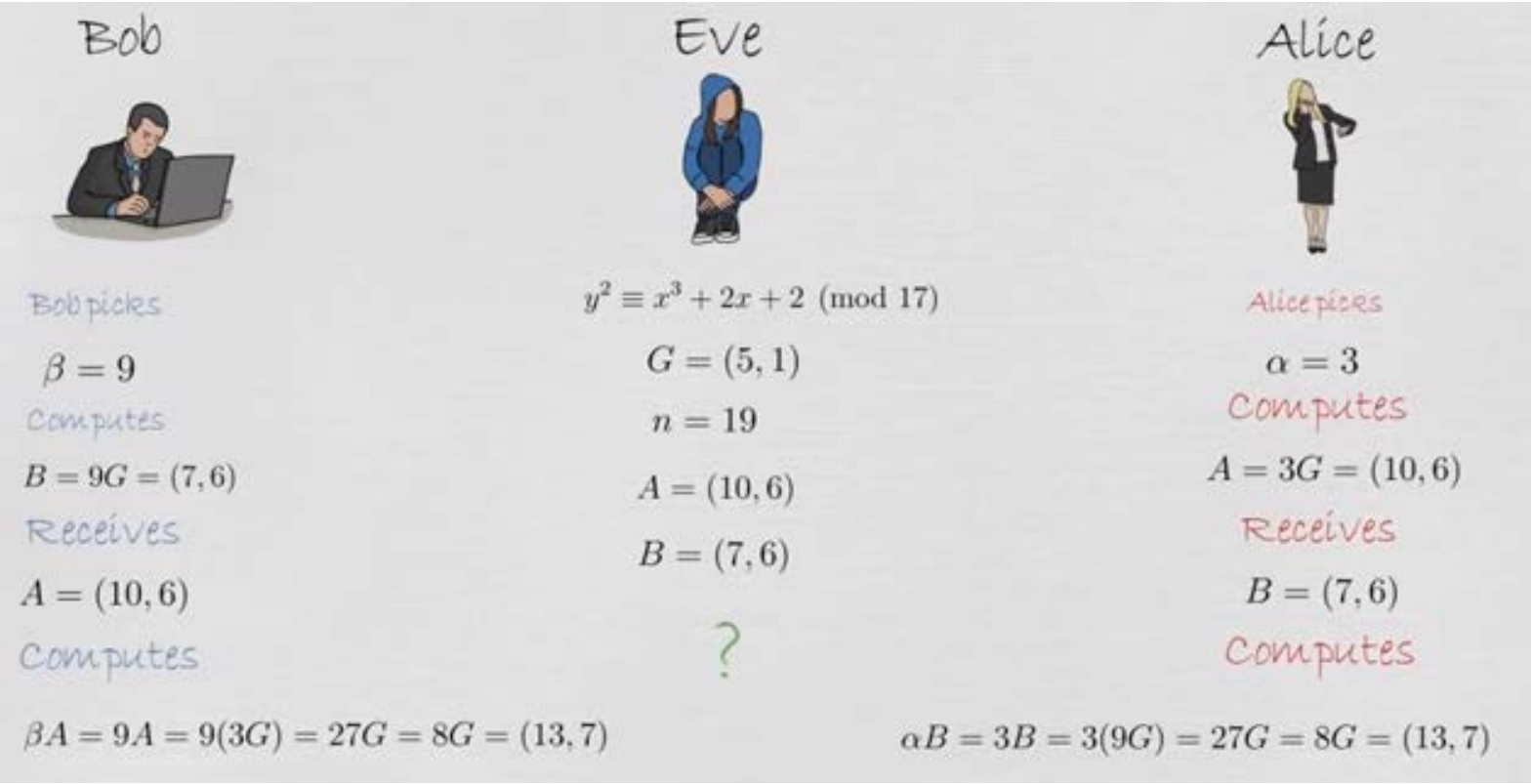
$$E : \quad y^2 \equiv x^3 + 2x + 2 \pmod{17}$$

- p = 17
- a = 2
- b = 2
- G (5 , 1)
- n = **19**

Algorithmes de chiffrement asymétrique

ECC

ECC pour l'échange de clés (Exemple)



Algorithmes de chiffrement asymétrique

ECC

A Real World Example

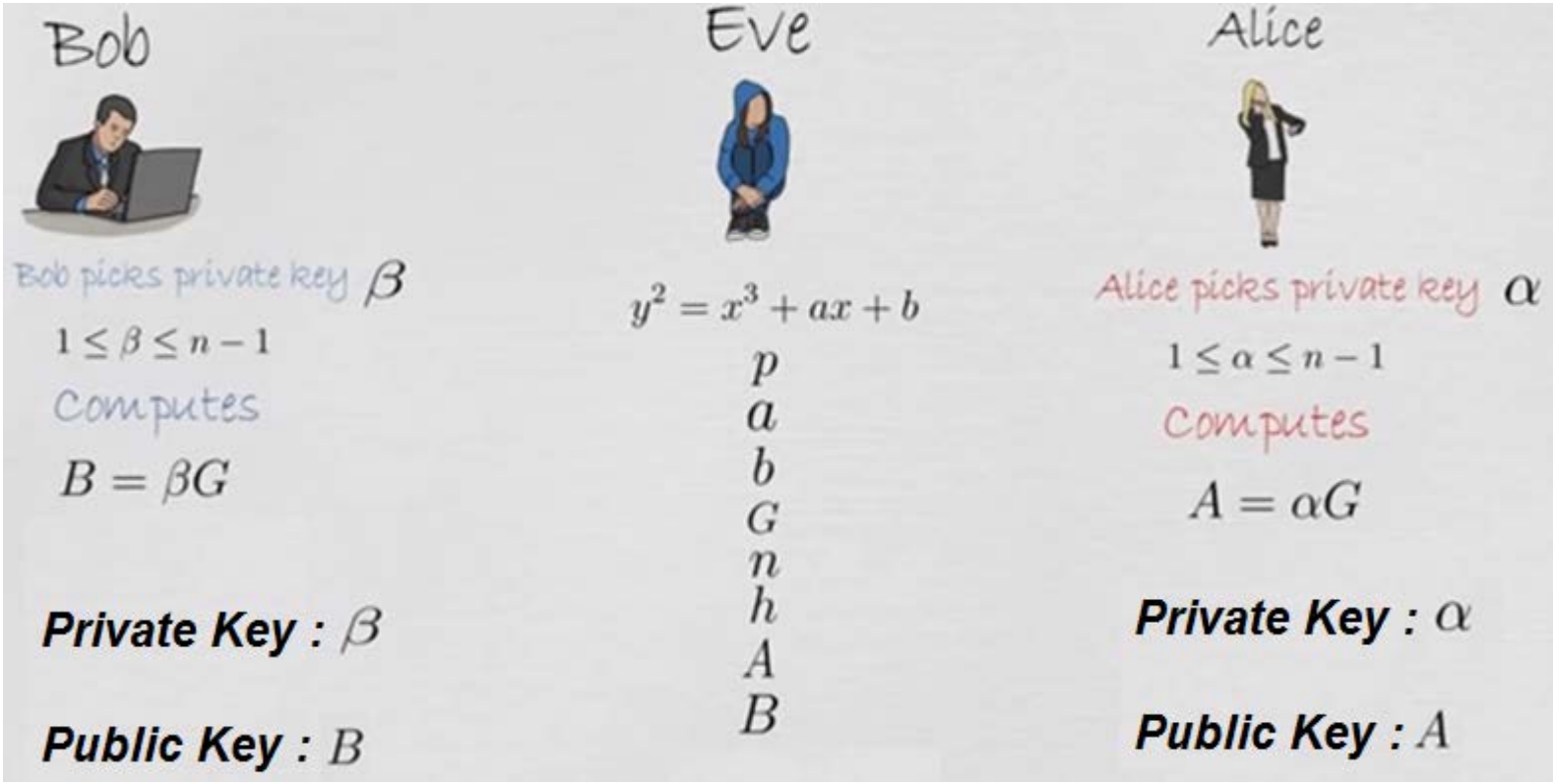
A Microsoft Digital Rights Management Curve

$$p = 7859631023794288223766947894468973962074985689 51$$
$$a = 3176890812513255034763174764138276932727469559 27$$
$$b = 7905289660787875871812057202571853543210065193 4$$
$$G_x = 7715072162626498261706482685655798899076925417 6$$
$$G_y = 3901575102465566285252794592665149955625331966 55$$

Algorithmes de chiffrement asymétrique

ECC

ECC pour chiffrer les données



Algorithmes de chiffrement asymétrique

ECC

ECC pour chiffrer les données

1. Une fois les clés privées et publiques sont générées, il faut encoder le texte clair **M** comme un point P_M de coordonnées **x** et **y**. C'est ce point qui sera chiffré.

2. Pour chiffrer le message, Alice détermine aléatoirement un nombre entier positif **k** et produit C_M comme un couple de points tel que : $C_M = (kG , P_M + kB)$

3. Pour déchiffrer, Bob devra multiplier le premier point par sa clé privée, et soustraire le résultat au second point reçu : $P_M + kB - \beta(kG) = P_M + k(\beta G) - \beta(kG) = P_M$

Algorithmes de chiffrement asymétrique

ECC

	Asymétrique(RSA)	Asymétrique(ECC)
Avantages	- Cryptosystème largement répandu - Nombreuses études au sujet de sa sécurité	- Taille de clé inférieure pour une sécurité égale - La taille des clés croit moins vite que le RSA si on souhaite une meilleure sécurité - Utilisation pour systèmes embarqués - Calculs moins lourds que l'exponentiation - Utilisation mémoire moindre - Cryptanalyse par algorithme exponentiel
Inconvénients	- Opérations de dé/chiffrement très inégales en termes de temps de calcul - Cryptanalyse par algorithme sous-exponentiel	- Complexe - Peu de développement sur des systèmes à grande échelle (mais tend à changer) - Travaux d'optimisation essentiellement destinés aux systèmes mobiles

Algorithmes de chiffrement asymétrique

ECC

	Symétrique	Asymétrique
Avantages	- Rapidité (jusqu'à 1000 fois plus rapide) - Facilité d'implantation sur hardware - Taille de clé : 128 bits ($\Rightarrow$ 16 caractères : mémorisable)	- Distributions des clés facilitées : pas d'authentification - Permet de signer des messages facilement - Nombre de clés à distribuer est réduit par rapport aux clés symétriques
Inconvénients	- Nombre de clés à gérer - Distribution des clés (authentification, confidentialité) - Certaines propriétés (p.ex. signatures) sont difficiles à réaliser	- Taille des clés - Vitesse de chiffrement

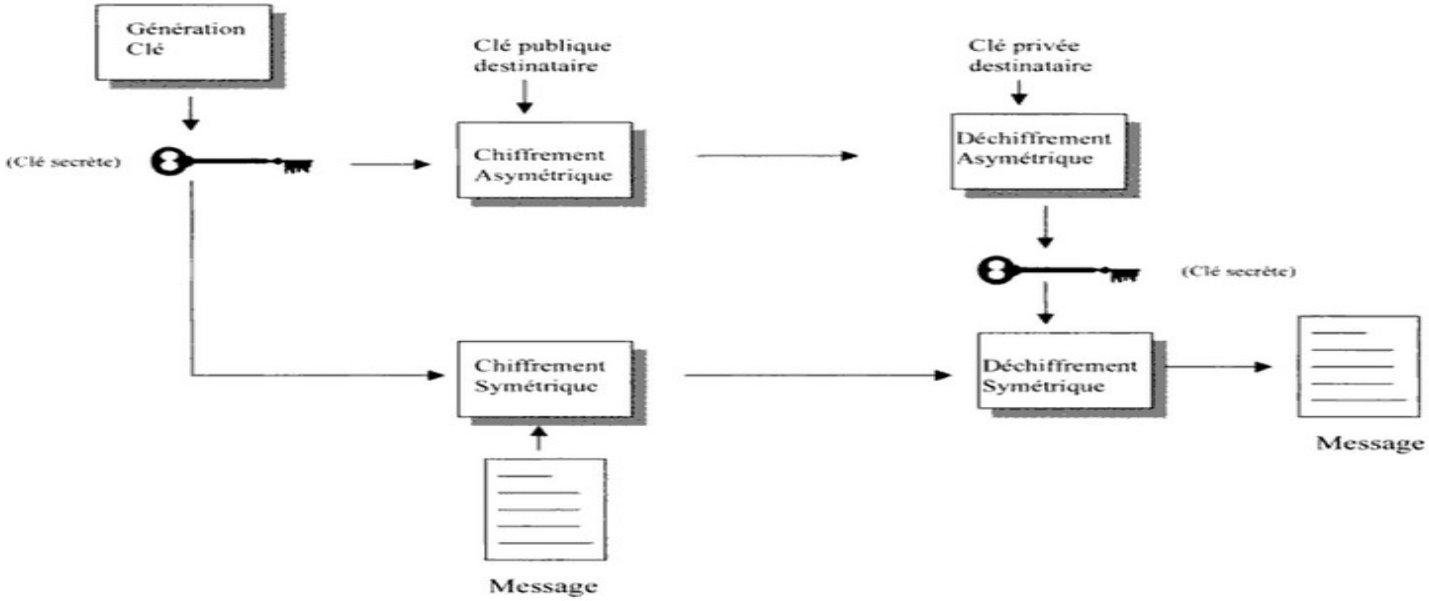
Plan du chapitre 4 (1/2)

Cryptographie

1. Introduction
2. La cryptographie
   - Confidentialité des messages
     - Chiffrement symétrique
     - Chiffrement asymétrique
     - Chiffrement mixte ou hybride
   - Authentification, Non-répudiation et Intégrité des messages
     - Fonctions de hachage
     - MAC – Message Authentication Code
     - Signature numérique
   - Gestion de clés
     - Distribution des clés
     - Protocole d'échange de clés de Diffie-Hellman
     - Certificat de clé publique
     - PKI – Public Key Infrastructure
3. La cryptanalyse
   - Attaque sur texte chiffré seul
   - Attaque à texte clair connu
   - Attaque à texte clair choisi
   - Attaque à texte chiffré choisi

- Le chiffrement asymétrique est intrinsèquement lent à cause des calculs complexes qui y sont associés, alors que le chiffrement symétrique brille par sa rapidité.
- D'autre part, le chiffrement symétrique souffre d'une grave lacune : Nécessité d'une transmission sécurisée de la clé secrète.
- Le chiffrement mixte (ou hybride) combine les deux systèmes cryptographiques : la chiffrement symétrique et le chiffrement asymétrique.
- L'objectif du chiffrement mixte est d'associer le meilleur des deux systèmes, à savoir la sécurité et le confort du chiffrement asymétrique et la rapidité du chiffrement symétrique.
- Le système cryptographique symétrique est utilisé pour crypter le contenu du message à transmettre, alors que le système cryptographique asymétrique est utilisé pour la distribution de la clé secrète.

- Pour communiquer de manière sécurisée, l'émetteur génère aléatoirement une clé de taille raisonnable appelée clé de session. Cette clé sera utilisée pour chiffrer toute la communication.
- Pour assurer la transmission sécurisée de la clé ainsi générée, l'émetteur utilise un système de chiffrement asymétrique (la clé est chiffrée à l'aide de la clé publique du destinataire avant d'être transmise).
- Une fois transmise, la clé sera déchiffrée à l'aide de la clé privée du destinataire et ensuite utilisée pour déchiffrer la communication.



Plan du chapitre 4 (1/2)

Cryptographie

1. Introduction
2. La cryptographie
   - Confidentialité des messages
     - Chiffrement symétrique
     - Chiffrement asymétrique
     - Chiffrement mixte ou hybride
   - **Authentification, Non-répudiation et Intégrité des messages**
     - Fonctions de hachage
     - MAC – Message Authentication Code
     - Signature numérique
   - Gestion de clés
     - Distribution des clés
     - Protocole d'échange de clés de Diffie-Hellman
     - Certificat de clé publique
     - PKI – Public Key Infrastructure
3. La cryptanalyse
   - Attaque sur texte chiffré seul
   - Attaque à texte clair connu
   - Attaque à texte clair choisi
   - Attaque à texte chiffré choisi

- Le chiffrement est une primitive cryptographique qui sert principalement pour assurer la confidentialité d'une communication. Cependant, d'autres facteurs peuvent entrer en ligne de compte pour garantir une communication sûre et sécurisée, à savoir : l'authenticité, la non-répudiation et l'intégrité de message.

- Pour assurer cette demande, d'autres primitives cryptographiques ont été conçues :
  - Les fonctions de hachage
  - Les codes d'authentification de message (MAC - Message authentication code)
  - Les signatures numériques

PLAN DU CHAPITRE 4 (1/2)

CRYPTOGRAPHIE

1. Introduction
2. La cryptographie
   - Confidentialité des messages
     - Chiffrement symétrique
     - Chiffrement asymétrique
     - Chiffrement mixte ou hybride
   - **Authentification, Non-répudiation et Intégrité des messages**
     - **Fonctions de hachage**
     - MAC – Message Authentication Code
     - Signature numérique
   - Gestion de clés
     - Distribution des clés
     - Protocole d'échange de clés de Diffie-Hellman
     - Certificat de clé publique
     - PKI – Public Key Infrastructure
3. La cryptanalyse
   - Attaque sur texte chiffré seul
   - Attaque à texte clair connu
   - Attaque à texte clair choisi
   - Attaque à texte chiffré choisi

- Une fonction de hachage est une méthode permettant de caractériser une information, une donnée. En faisant subir une suite de traitements reproductibles à une entrée, elle génère une empreinte servant à identifier la donnée initiale.
- Une fonction de hachage prend donc en entrée un message de taille quelconque, applique une série de transformations et réduit ces données. On obtient à la sortie une chaîne de caractères hexadécimaux, le condensé, qui résume en quelque sorte le fichier. Cette sortie a une taille fixe qui varie selon les algorithmes (128 bits pour MD5 et 160 bits pour SHA-1). Les fonctions de hachage sont très utilisées en informatique et en cryptographie. Leur qualité principale est que les algorithmes correspondants sont publics.
- Le résultat d'une fonction de hachage peut être appelé selon le contexte : somme de contrôle, empreinte, empreinte numérique, hash, hash code, digest, résumé de message, condensé, condensat, signature ou encore empreinte cryptographique lorsque l'on utilise une fonction de hachage cryptographique.
- Exemples d'usage :
  - ✓ Contrôle d'intégrité
  - ✓ Stockage de mots de passe
  - ✓ Signature numérique
  - ✓ MAC – Message Authentication Code

Contrôle d'intégrité

Par exemple en navigant sur le Web : les auteurs de logiciels proposent souvent des empreintes sur les pages dédiées aux téléchargements (des fichiers portant l'extension md5 ou sha1, qui contiennent la valeur hachée dudit programme). En comparant l'empreinte de la version téléchargée avec l'empreinte disponible sur le site, l'utilisateur peut s'assurer que sa version n'a pas été corrompue (erreurs de transmission, virus, etc.)

Stockage de mots de passe

Les serveurs où on se connecte avec un mot de passe :

- Si les mots de passe sont stockés en clair sur le serveur, toute attaque sur le serveur implique que tous les mots de passe sont sniffés.
- **Alternative :** Stocker le haché H(login || mot de passe). Ainsi, lorsque l'utilisateur se connectera, le serveur ne va pas vérifier si le couple (login || mot de passe) est identique, mais il va vérifier que la signature du (login || mot de passe) saisi est bien la même que la signature du (login || mot de passe) enregistré.

On demande à une fonction de hachage cryptographique de remplir les conditions suivantes :

- Pouvoir s'appliquer à n'importe quelle longueur de message M ;
- Produire un résultat de longueur constante ;
- Facilité de calculer h = H(M) pour n'importe quel message M ;
- Résistance au calcul de pré-image :

  Pour un h donné, il est impossible de trouver x tel que H(x) = h. On parle de propriété à sens unique ;
- Résistance au calcul de seconde pré-image :

  Pour un x donné, il est impossible de trouver y ≠ x tel que H(y) = H(x) ;
- Résistance aux collisions :

  Il est impossible de trouver x, y tels que H(y) = H(x) ;

Quelques fonctions de hachage célèbres :

- **MD5**

  Cette fonction de hachage est toujours très utilisée bien qu'au niveau de la sécurité, il est recommandé de passer à des versions plus robustes car des suites de collisions ont été trouvées. Elle renvoie une empreinte de 128 bits.

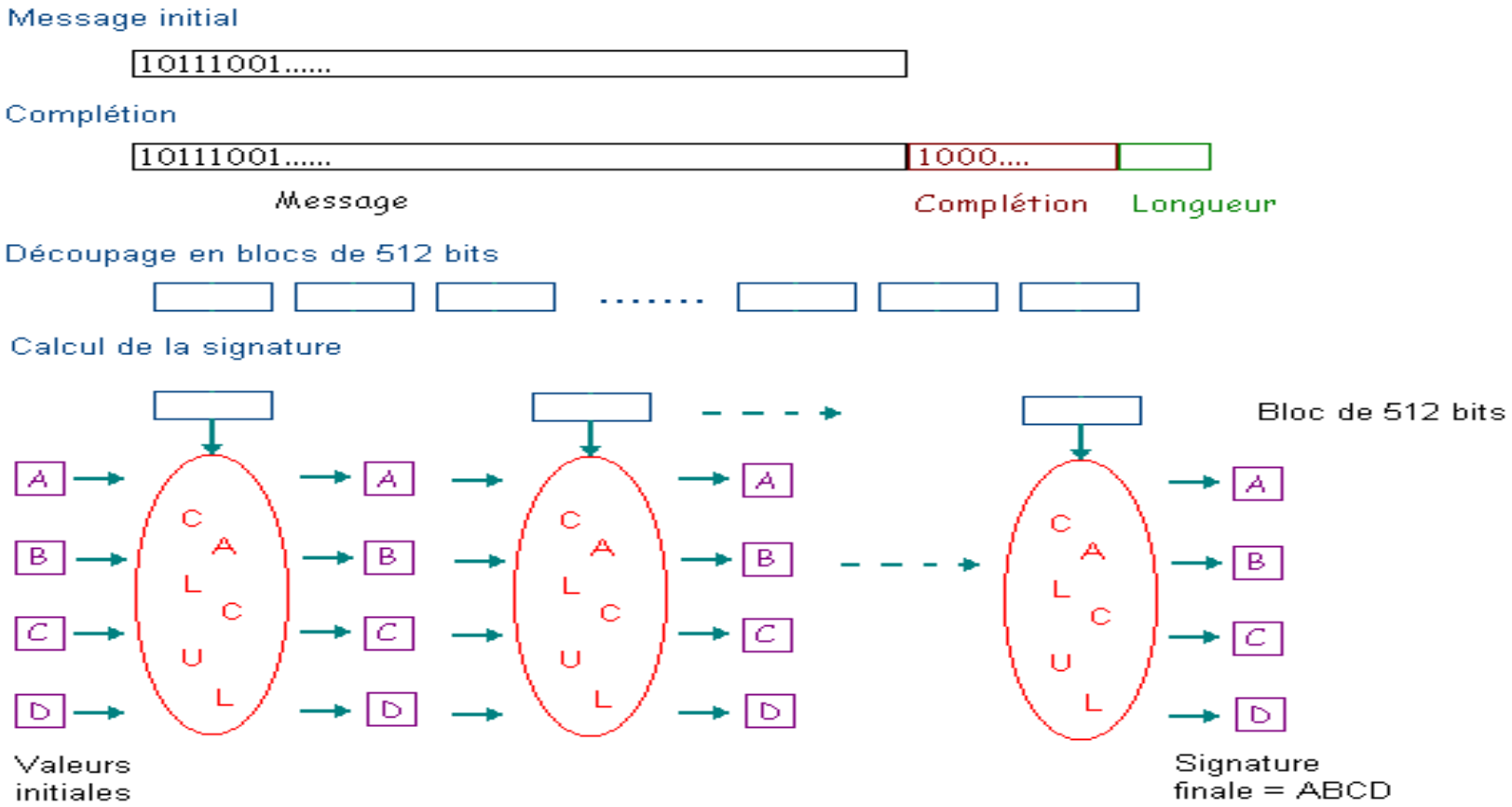
- **SHA1**

  Était la fonction remplaçante de MD5 car elle produisait des empreintes 160 bits et avec impossibilité de trouver des collisions … jusqu'en 2004-2005, date à laquelle des attaques ont prouvé des possibilités de générer des collisions. Depuis, cette date il n'est plus conseillé d'utiliser la fonction SHA1. Mais, elle est encore très utilisée.

- **SHA2 …**

  SHA256 et SHA512 sont 2 des grands standards utilisés actuellement, car il n'y a pas à ce jour d'attaques ayant trouvé des failles de sécurité sur ces fonctions de hachage. Elles produisent des signatures de respectivement 256 et 512 bits.

Exemple de fonction de hachage : MD5



Description du fonctionnement du MD5

Exemple de fonction de hachage : MD5

Étape 1 : Complétion

- MD5 manipule des blocs de 512 bits. Il complète la longueur du message en entrée telle que la longueur soit congrue à 448 modulo 512, en rajoutant un 1 suivi d'autant de 0 que nécessaire à la fin du message (opération de bourrage ou padding).
- Ensuite, la longueur initiale (avant le padding) du message est rajoutée aux 448 bits, sous forme de 64 bits ce qui amène à une taille multiple de 512 bits.
- Les blocs de 512 bits sont traités itérativement. Chaque bloc est décomposé en 16 mots de 32 bits chacun. Le résultat du calcul effectué sur chaque bloc est représenté par un ensemble de 4 mots (4 buffers) de 32 bits : A, B, C, et D.

Étape 2 : Initialisation

MD5 prend 4 variables en entrée initialisées en hexadécimal de la manière suivante :

A=01234567

B=89abcdef

C=fedcba98

D=76543210

Exemple de fonction de hachage : MD5

Étape 3 : Calcul itératif

- MD5 comprend 4 rondes qui exécutent chacune 16 opérations.
- Les 4 fonctions non-linéaires prévues pour chaque ronde sont les suivantes :

F(X,Y,Z) = (X AND Y) OR (not(X) AND Z)

G(X,Y,Z) = (X AND Z) OR (Y AND not(Z))

H(X,Y,Z) = X xor Y xor Z

I(X,Y,Z) = Y xor (X OR not(Z))

- Les fonctions F, G, H, et I prennent des arguments codés sur 32 bits et renvoie une valeur sur 32 bits, les opérations se faisant bit à bit.

Exemple de fonction de hachage : MD5

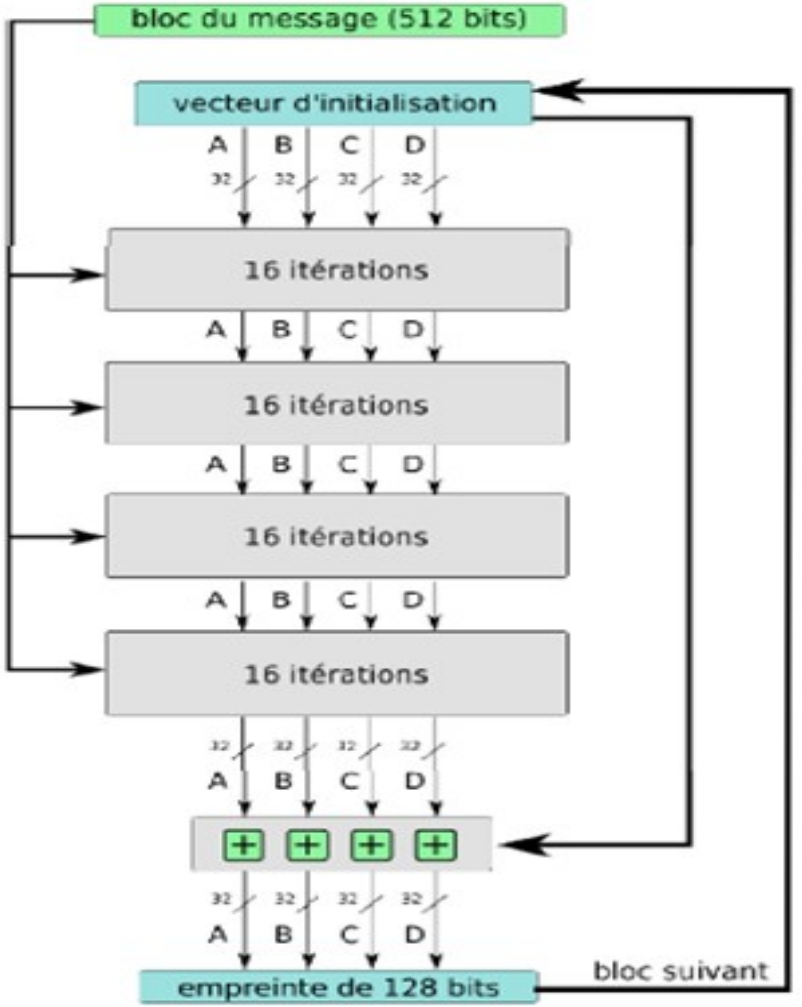
Ainsi, pour chaque bloc de 512 bits du texte, on fait les opérations suivantes :

1- On sauvegarde les valeurs initiales de A, B, C, et D dans AA, BB, CC, et DD.

2- On calcule de nouvelles valeurs pour A, B, C, et D à partir de leurs anciennes valeurs, à partir des bits du bloc qu'on est entrain d'étudier et à partir des 4 fonctions F, G, H, et I.

3- On détermine les valeurs finales de A, B, C, et D :

A = A + AA
B = B + BB
C = C + CC
D = D + DD



Exemple de fonction de hachage : MD5

Étape 4 : Écriture du résumé

Le résumé sur 128 bits est obtenu en mettant bout à bout les 4 buffers A, B, C, et D de 32 bits.

Détails du calcul du MD5

Voir document ci-joint.

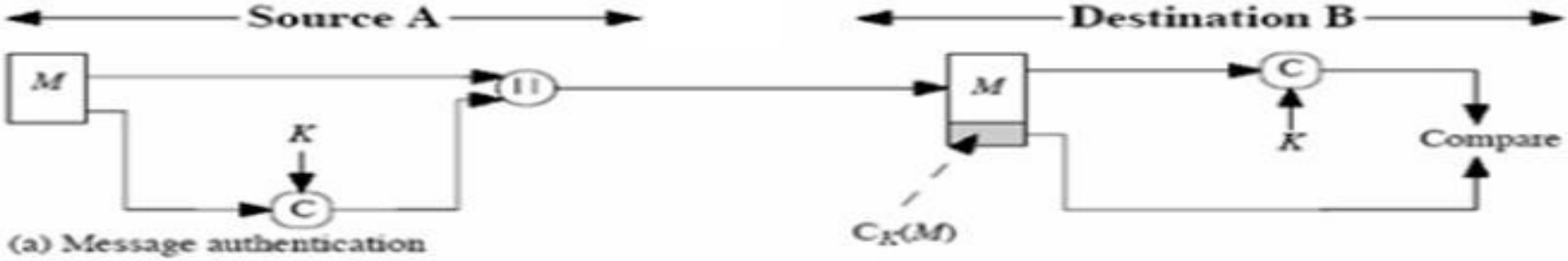
Plan du chapitre 4 (1/2)

Cryptographie

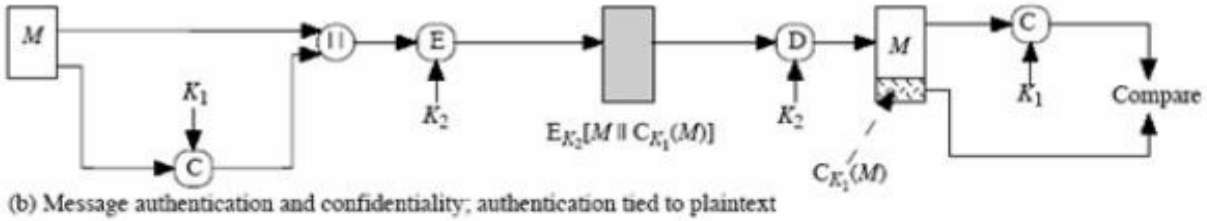
1. Introduction
2. La cryptographie
   - Confidentialité des messages
     - Chiffrement symétrique
     - Chiffrement asymétrique
     - Chiffrement mixte ou hybride
   - **Authentification, Non-répudiation et Intégrité des messages**
     - Fonctions de hachage
     - **MAC – Message Authentication Code**
     - Signature numérique
   - Gestion de clés
     - Distribution des clés
     - Protocole d'échange de clés de Diffie-Hellman
     - Certificat de clé publique
     - PKI – Public Key Infrastructure
3. La cryptanalyse
   - Attaque sur texte chiffré seul
   - Attaque à texte clair connu
   - Attaque à texte clair choisi
   - Attaque à texte chiffré choisi

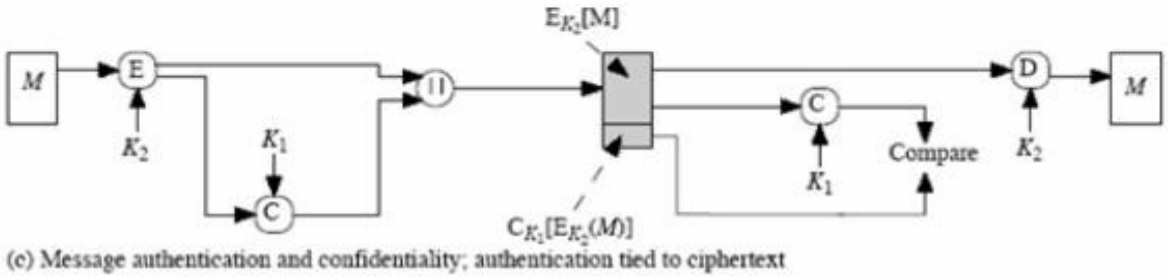
- Le concept est relativement semblable aux fonctions de hachage. Il s'agit ici aussi d'algorithmes qui créent un petit bloc authentificateur de taille fixe.
- La grande différence est que ce bloc authentificateur ne se base plus uniquement sur le message, mais également sur une clé secrète.
- Tout comme les fonctions de hachage, les MACs n'ont pas besoin d'être réversibles. En effet, le récepteur exécutera le même calcul sur le message et le comparera avec le MAC reçu.
- Le MAC assure que le message est inchangé (intégrité) et vient de l'expéditeur (authentification, par l'utilisation de la clé secrète). Il peut également être employé comme un chiffrement supplémentaire (rare) et peut être calculé avant ou après le chiffrement principal, bien qu'il soit généralement conseillé de le faire avant.

Il existe 3 techniques de base pour l'utilisation d'un MAC :



– Seule l'authentification du message (de la source A) a lieu. En effet, le message est envoyé en clair sur le réseau, et le MAC lui est concaténé. Cependant, par l'utilisation d'une clé secrète, et comme seul A connait cette clé, le destinataire B est sûr que l'expéditeur est bien A.





– En (b) et (c), on ajoute la confidentialité.
– En (b), le MAC est concaténé au message, et le tout est chiffré et envoyé à B. (la technique la plus utilisée en pratique)
– En (c), seul le message est chiffré. Le MAC est concaténé par la suite, et le tout est envoyé à B.

Exemples d'algorithmes pour les codes d'authentification de message (MAC) :

- ✓ CBC-MAC
- ✓ HMAC

CBC-MAC

- CBC-MAC est l'un des algorithmes pour les codes d'authentification de message (MAC). Il est basé sur un chiffrement par bloc utilisé selon un mode d'opération CBC (cipher block chaining).

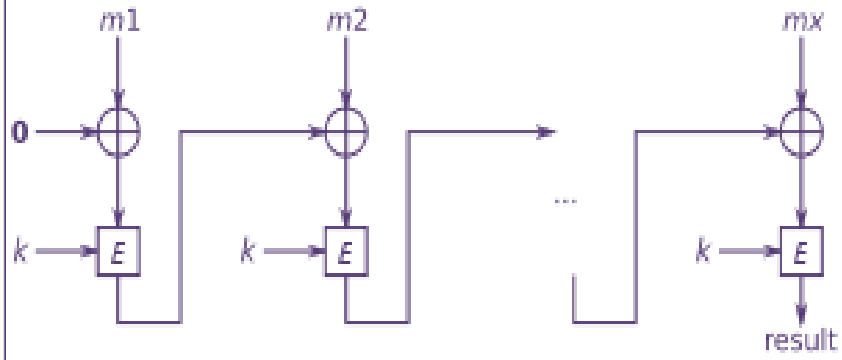
- Pour chiffrer, on découpe les données en blocs de taille adéquate (au minimum un chiffrement par bloc de 32 bits). Les blocs sont chiffrés les uns après les autres, le résultat chiffré du bloc précédent est transmis au bloc suivant.

Soit $E_k(M_i)$ l'opération de chiffrement sur un bloc de données M_i avec la clé k.

Le chiffrement se fait comme suit :

1- On découpe les données en blocs de taille fixe $M_0, \ldots, M_{n-1}$
2- On définit un vecteur d'initialisation $C_0 = 0$
3- On traite les blocs au fur et à mesure : $C_{i+1} = E_k(C_i \oplus M_{i+1})$

Le code d'authentification correspond à une partie du dernier bloc chiffré C_{n-1}.



HMAC

- Un HMAC (keyed-Hash Message Authentication Code) est un type de MAC calculé en utilisant une fonction de hachage cryptographique en combinaison avec une clé secrète.
- N'importe quelle fonction itérative de hachage (MD5 ou SHA-1) peut être utilisée dans le calcul d'un HMAC ; le nom de l'algorithme résultant est donc HMAC-MD5 ou HMAC-SHA-1.
- La qualité cryptographique du HMAC dépend de la qualité cryptographique de la fonction de hachage et de la taille et la qualité de la clé.
- Une fonction itérative de hachage découpe un message en blocs de taille fixe et itère dessus avec une fonction de compression.
- La taille de la sortie HMAC est la même que celle de la fonction de hachage (128 ou 160 bits dans les cas du MD5 et SHA-1).

HMAC

La fonction HMAC est définie comme suit :

$$\mathrm{HMAC}_K(m) = h\Big((K \oplus opad) \,||\, h\big((K \oplus ipad) \,||\, m\big)\Big)$$

avec :

- h : une fonction de hachage itérative,
- K : la clé secrète complétée avec des zéros pour qu'elle atteigne la taille de bloc de la fonction h
- m : le message à authentifier,
- "||" désigne une concaténation et " $\oplus$ " un « ou » exclusif,
- ipad et opad, chacune de la taille d'un bloc, sont définies par :
  ipad = 0x363636...3636 et opad = 0x5c5c5c...5c5c.
  Donc, si la taille de bloc de la fonction de hachage est 512 bits, ipad et opad sont 64 répétitions des octets, respectivement, 0x36 et 0x5c.

PLAN DU CHAPITRE 4 (1/2)

CRYPTOGRAPHIE

1. Introduction
2. La cryptographie
   - Confidentialité des messages
     - Chiffrement symétrique
     - Chiffrement asymétrique
     - Chiffrement mixte ou hybride
   - **Authentification, Non-répudiation et Intégrité des messages**
     - Fonctions de hachage
     - MAC – Message Authentication Code
     - **Signature numérique**
   - Gestion de clés
     - Distribution des clés
     - Protocole d'échange de clés de Diffie-Hellman
     - Certificat de clé publique
     - PKI – Public Key Infrastructure
3. La cryptanalyse
   - Attaque sur texte chiffré seul
   - Attaque à texte clair connu
   - Attaque à texte clair choisi
   - Attaque à texte chiffré choisi

- L'authentification des messages par Hachage ou MAC protège les deux entités communicantes d'un intrus. Cependant, cela ne les protège en rien l'un de l'autre. En cas de conflit, rien n'empêche la répudiation du ou des messages. => **Signature numérique**
- La signature numérique (appelée aussi signature électronique ou digitale) est un mécanisme permettant de garantir l'intégrité d'un document électronique et d'en authentifier l'auteur, par analogie avec la signature manuscrite d'un document papier.
- Un mécanisme de signature numérique doit présenter les propriétés suivantes :
  - Il doit permettre au lecteur d'un document d'identifier la personne ou l'organisme qui a apposé sa signature (propriété d'identification).
  - Il doit garantir que le document n'a pas été altéré entre l'instant où l'auteur l'a signé et le moment où le lecteur le consulte (propriété d'intégrité).
- Pour cela, les conditions suivantes doivent être réunies :
  - ✓ **Authentique** : l'identité du signataire doit pouvoir être retrouvée de manière certaine.
  - ✓ **Infalsifiable** : la signature ne peut pas être falsifiée. Quelqu'un ne peut se faire passer pour un autre.
  - ✓ **Non réutilisable** : la signature n'est pas réutilisable. Elle fait partie du document signé et ne peut être déplacée sur un autre document.
  - ✓ **Inaltérable** : un document signé est inaltérable. Une fois qu'il est signé, on ne peut plus le modifier.
  - ✓ **Irrévocable :** la personne qui a signé ne peut le nier.

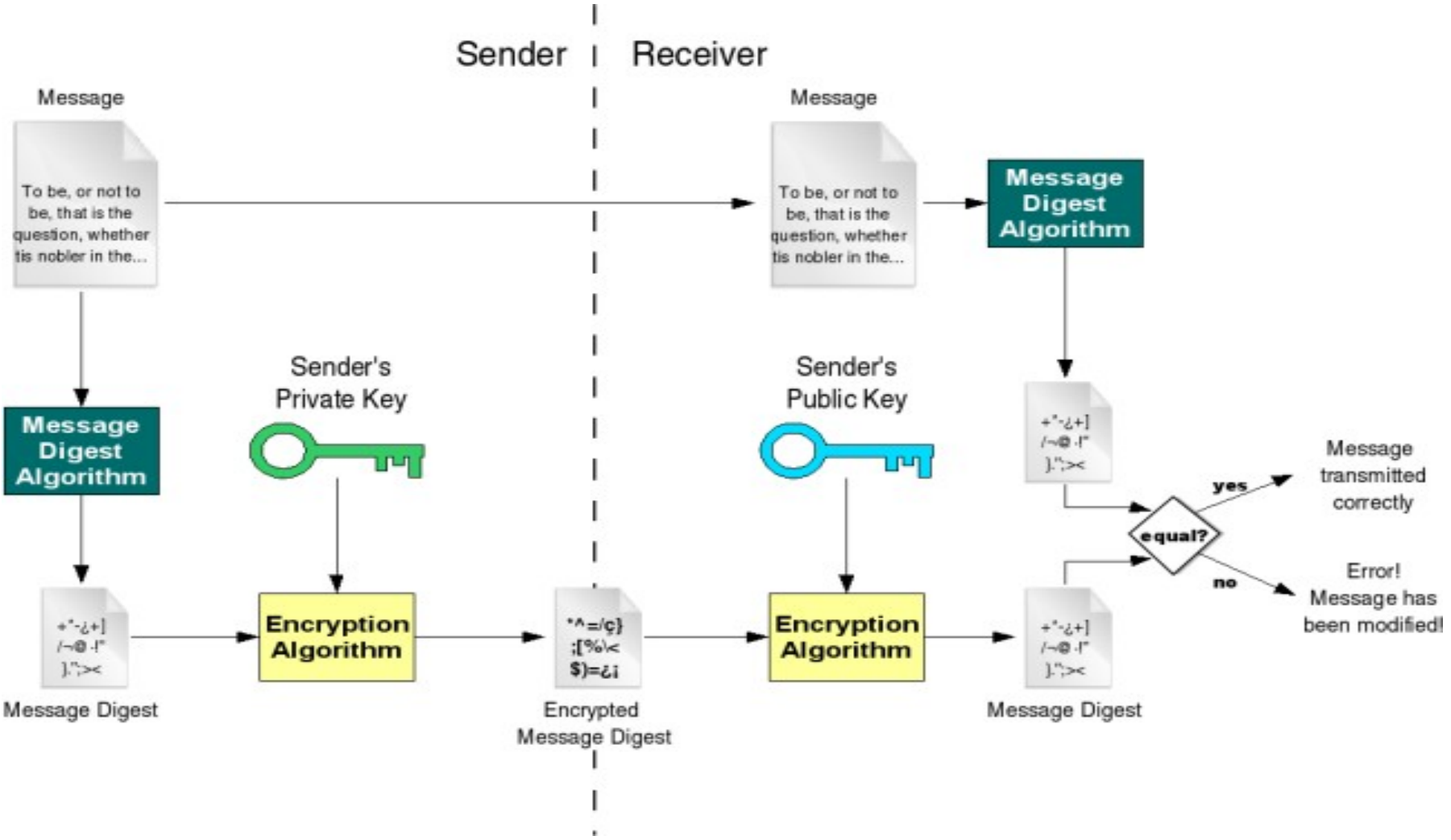
- Supposons qu'Alice souhaite envoyer à Bob un message dont il puisse vérifier l'authenticité.
- Alice et Bob ont convenu au préalable des choix :
  - Un chiffrement asymétrique constitué d'une fonction de chiffrement **C** et d'une fonction de déchiffrement **D** ;
  - Une fonction de hachage notée **H**.
- Pour le chiffrement choisi, Alice a généré une clé privée K_{pr} et une clé publique K_{pb} :
  - elle transmet la clé K_{pb} à Bob par un canal non sécurisé (la clé publique n'est pas secrète) ;
  - elle garde la clé K_{pr} secrète.

Préparation du message signé

- Alice prépare le message signé, pour cela :
  - elle produit un condensat du message par la fonction de hachage choisie **H(M)** ;
  - elle chiffre ce condensat grâce à la fonction de chiffrement **C** en utilisant sa clé privée K_{pr}. Le résultat obtenu est la signature du message : $\mathbf{S_M} = \mathbf{C}(K_{pr}, \mathbf{H}(\mathbf{M}))$;
  - elle prépare le message signé en plaçant le message en clair **M** et la signature $\mathbf{S_M}$ dans un conteneur quelconque : $\mathbf{M_{signé}} = (\mathbf{S_M}, \mathbf{M})$.
- Alice transmet $\mathbf{M_{signé}}$, le message signé, à Bob par un canal non sécurisé.

Réception du message signé

- Bob réceptionne le message signé, pour vérifier l'authenticité du message :
  - il produit un condensat du texte clair en utilisant la fonction de hachage convenue : $\mathbf{H(M)}$;
  - il déchiffre la signature en utilisant la fonction de déchiffrement **D** avec la clé publique K_{pb} soit : $\mathbf{D_{Sm}=D}(K_{pb},\mathbf{S_M})$;
  - il compare $\mathbf{D_{Sm}}$ avec $\mathbf{H(M)}$.
- Dans le cas où la signature est authentique, $\mathbf{D_{Sm}}$ avec $\mathbf{H(M)}$ sont égaux car, par les propriétés du chiffrement asymétrique : $\mathbf{D_{Sm}=D}(K_{pb},\mathbf{S_M})=\mathbf{D}(K_{pb},\mathbf{C}(K_{pr},\mathbf{H(M)}))=\mathbf{H(M)}$, le message est alors authentifié.


Sender
Receiver
Message
To be, or not to be, that is the question, whether tis nobler in the...
Message
To be, or not to be, that is the question, whether tis nobler in the...
Message Digest Algorithm
Message Digest Algorithm
Sender's Private Key
Sender's Public Key
+"-¿+]
/¬@·!"
}.";><
Encryption Algorithm
*^=/ç}
;[%\<
$)=¿¡
Encryption Algorithm
+"-¿+]
/¬@·!"
}.";><
+"-¿+]
/¬@·!"
}.";><
equal?
yes
no
Message transmitted correctly
Error! Message has been modified!
Message Digest
Encrypted Message Digest
Message Digest

Plan du chapitre 4 (1/2)

Cryptographie

1. Introduction
2. La cryptographie
   - Confidentialité des messages
     - Chiffrement symétrique
     - Chiffrement asymétrique
     - Chiffrement mixte ou hybride
   - Authentification, Non-répudiation et Intégrité des messages
     - Fonctions de hachage
     - MAC – Message Authentication Code
     - Signature numérique
   - **Gestion de clés**
     - Distribution des clés
     - Protocole d'échange de clés de Diffie-Hellman
     - Certificat de clé publique
     - PKI – Public Key Infrastructure
3. La cryptanalyse
   - Attaque sur texte chiffré seul
   - Attaque à texte clair connu
   - Attaque à texte clair choisi
   - Attaque à texte chiffré choisi

La gestion des clés est principalement constituée de quatre domaines :

1. **La génération des clés** : il faut prendre garde aux caractères choisis, aux clés faibles, ... et veiller à utiliser des générateurs fiables ;
2. **Le transfert de la clé** : l'idéal est de se rencontrer, ou d'utiliser un canal de transmission protégé. Mais cela est souvent impossible ;
3. **La vérification des clés** : par hachage, ou utilisation de certificats ;
4. **Le stockage des clés** : que ce soit dans des fichiers, sur supports extérieurs, ...

Plan du chapitre 4 (1/2)

Cryptographie

1. Introduction
2. La cryptographie
   - Confidentialité des messages
     - Chiffrement symétrique
     - Chiffrement asymétrique
     - Chiffrement mixte ou hybride
   - Authentification, Non-répudiation et Intégrité des messages
     - Fonctions de hachage
     - MAC – Message Authentication Code
     - Signature numérique
   - **Gestion de clés**
     - **Distribution des clés**
     - Protocole d'échange de clés de Diffie-Hellman
     - Certificat de clé publique
     - PKI – Public Key Infrastructure
3. La cryptanalyse
   - Attaque sur texte chiffré seul
   - Attaque à texte clair connu
   - Attaque à texte clair choisi
   - Attaque à texte chiffré choisi

Cas des clés symétriques	Cas des clés asymétriques
– Physiquement : par une rencontre, un canal de transmission protégé, ... – Utiliser un tiers de confiance. Celui-ci choisit et fournit la clé – Utiliser une ancienne clé pour chiffrer une nouvelle clé (ce qui suppose cependant un échange préalable de cette ancienne clé) – Utilisation du protocole d'échange de clés de Diffie-Hellman	– Annuaire publiquement disponible – Autorité de clés publique – Certificats de clé publique

PLAN DU CHAPITRE 4 (1/2)

CRYPTOGRAPHIE

1. Introduction
2. La cryptographie
   - Confidentialité des messages
     - Chiffrement symétrique
     - Chiffrement asymétrique
     - Chiffrement mixte ou hybride
   - Authentification, Non-répudiation et Intégrité des messages
     - Fonctions de hachage
     - MAC – Message Authentication Code
     - Signature numérique
   - **Gestion de clés**
     - Distribution des clés
     - **Protocole d'échange de clés de Diffie-Hellman**
     - Certificat de clé publique
     - PKI – Public Key Infrastructure
3. La cryptanalyse
   - Attaque sur texte chiffré seul
   - Attaque à texte clair connu
   - Attaque à texte clair choisi
   - Attaque à texte chiffré choisi

- Un protocole d'échange de clés symétriques totalement sécurisé qui repose sur l'arithmétique modulaire.

- **Principe :**

  Alice et Bob veulent s'échanger un message crypté en utilisant un algorithme de chiffrement symétrique. Pour ce faire, ils commencent par s'échanger la clé secrète par le protocole de Diffie-Hellman.

	Alice	Bob
Étape 1 :	Alice et Bob choisissent ensemble un grand nombre premier p et un entier $1 \leq a \leq p-1$. Cet échange n'a pas besoin d'être sécurisé.	
Étape 2 :	Alice choisit secrètement x_1.	Bob choisit secrètement x_2.
Étape 3 :	Alice calcule $y_1 = a^{x_1} \pmod{p}$.	Bob calcule $y_2 = a^{x_2} \pmod{p}$.
Étape 4 :	Alice et Bob s'échangent les valeurs de y_1 et y_2. Cet échange n'a pas besoin d'être sécurisé.	
Étape 5 :	Alice calcule $y_2^{x_1} = (a^{x_2})^{x_1} = a^{x_1 x_2} \pmod{p}$ et appelle ce nombre K, la clé secrète à partager avec Bob.	Bob calcule $y_1^{x_2} = (a^{x_1})^{x_2} = a^{x_1 x_2} \pmod{p}$ et appelle ce nombre K, la clé secrète à partager avec Alice.

- Si quelqu'un a espionné leurs conversations, il connait p, a, y_1 et y_2. Il ne peut pas retrouver K comme le font Alice ou Bob, car il lui manque toujours l'une des informations nécessaires, à savoir x_1 ou x_2. Et il ne peut pas retrouver x_1 connaissant $y = a^{x1} \pmod p$, a et p, puisque la résolution du logarithme discret est un problème difficile.

- **Défauts majeurs de ce protocole :**

  - ✓ Il exige la simultanéité des actions d'Alice et de Bob. Si Alice veut envoyer un e-mail à Bob alors que celui n'est pas connecté, elle ne pourra pas le faire immédiatement. C'est pourquoi ce protocole fut en réalité très vite supplanté par les méthodes de chiffrement à clé publique.

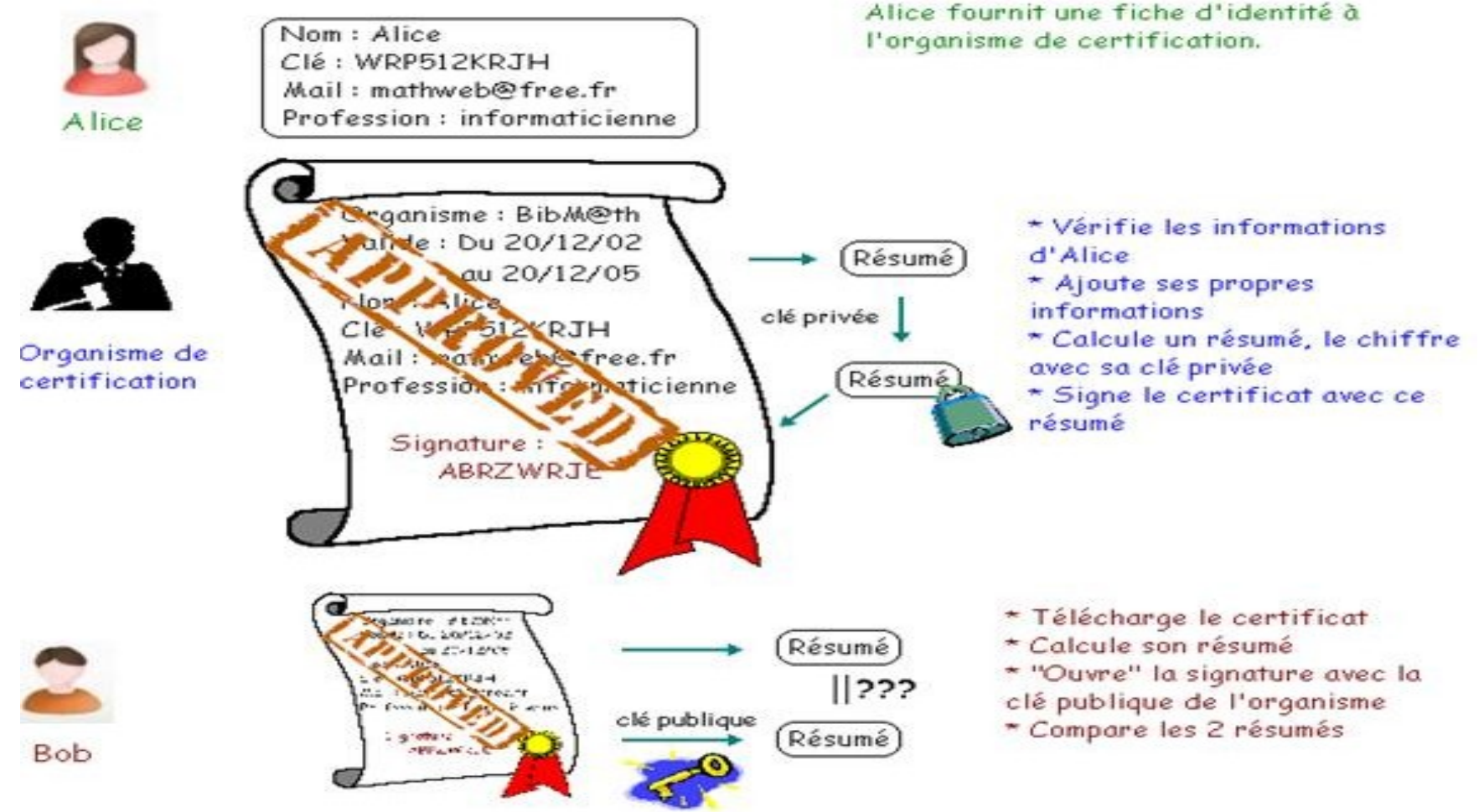
  - ✓ Sans authentification, il est très vulnérable à l'attaque de l'homme au milieu (Man-in-the-Middle).

Plan du chapitre 4 (1/2)

Cryptographie

1. Introduction
2. La cryptographie
   - Confidentialité des messages
     - Chiffrement symétrique
     - Chiffrement asymétrique
     - Chiffrement mixte ou hybride
   - Authentification, Non-répudiation et Intégrité des messages
     - Fonctions de hachage
     - MAC – Message Authentication Code
     - Signature numérique
   - **Gestion de clés**
     - Distribution des clés
     - Protocole d'échange de clés de Diffie-Hellman
     - **Certificat de clé publique**
     - PKI – Public Key Infrastructure
3. La cryptanalyse
   - Attaque sur texte chiffré seul
   - Attaque à texte clair connu
   - Attaque à texte clair choisi
   - Attaque à texte chiffré choisi

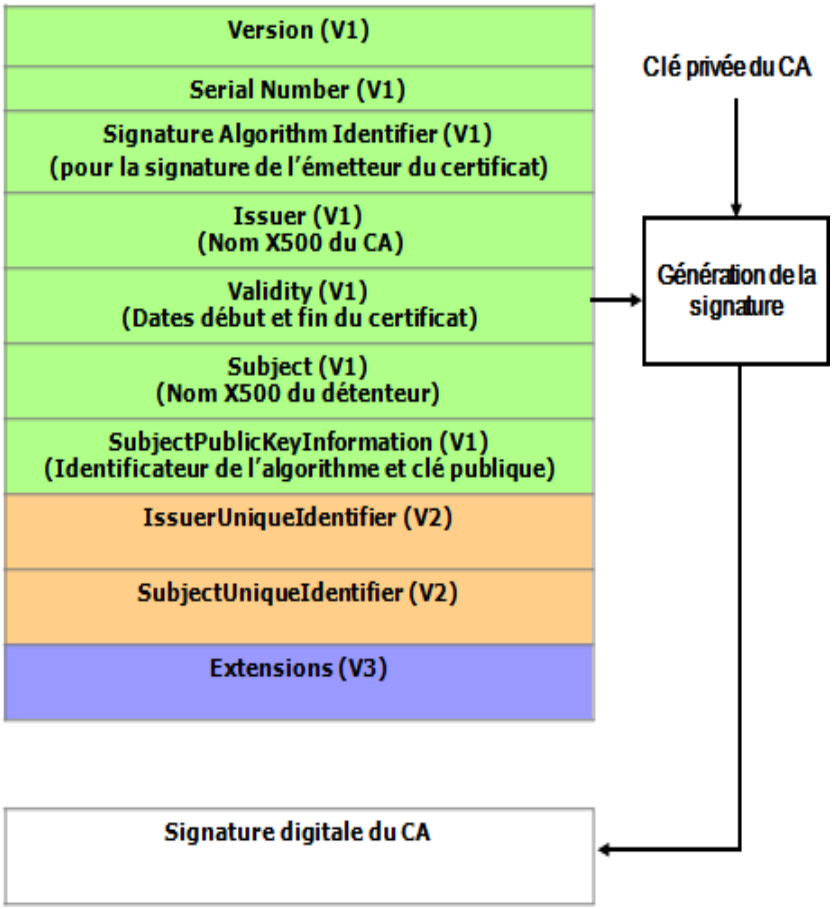
- Un certificat à clé publique est un certificat numérique qui lie l'identité d'un système à une clé publique, et éventuellement à d'autres informations;
- C'est une structure de donnée signée numériquement qui atteste sur l'identité du possesseur de la clé privée correspondante à une clé publique.
- Un certificat est signé numériquement par une autorité de certification (CA) à qui font confiance tous les usagers et dont la clé publique est connue par tous d'une manière sécurisée.
- Ainsi, afin de publier sa clé publique, son possesseur doit fournir un certificat de sa clé publique signé par l'autorité de certification.
- Après vérification de la signature apposée sur le certificat en utilisant la clé publique de l'autorité de certification, le récepteur peut déchiffrer et vérifier les signatures de son interlocuteur dont l'identité et la clé publique sont inclus dans le certificat.



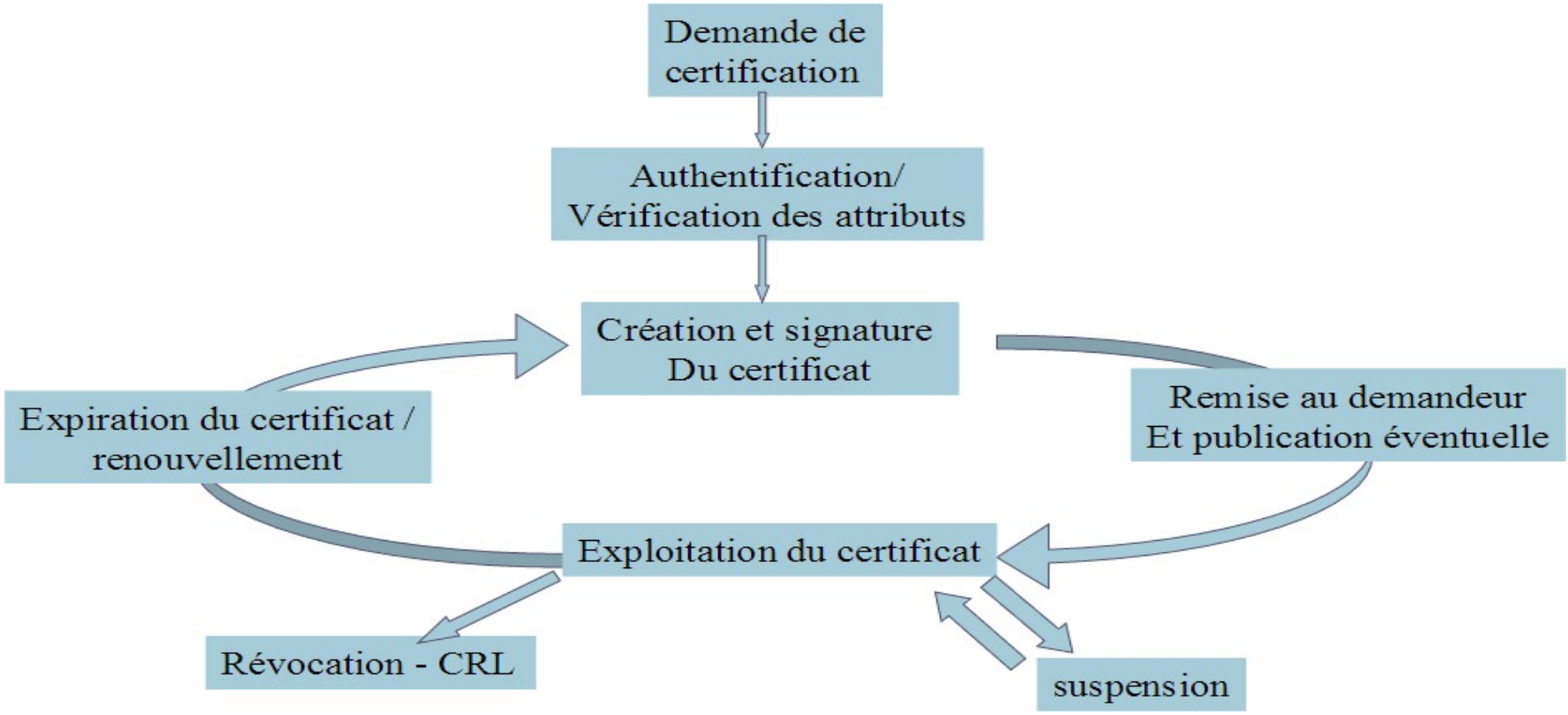
Certification numérique d'une clé publique

Structure d'un certificat X.509

- Version
- Numéro de série
- Algorithme de signature du certificat (le plus souvent SHA1-RSA)
- Signataire du certificat
- Validité (dates limite)
  - ✓ Pas avant
  - ✓ Pas après
- Détenteur du certificat
- Informations sur la clé publique
  - ✓ Algorithme de la clé publique
  - ✓ Clé publique
- Id unique du signataire (Facultatif)
- Id unique du détenteur du certificat (Facultatif)
- Extensions (Facultatif)



Cycle de vie de certificats



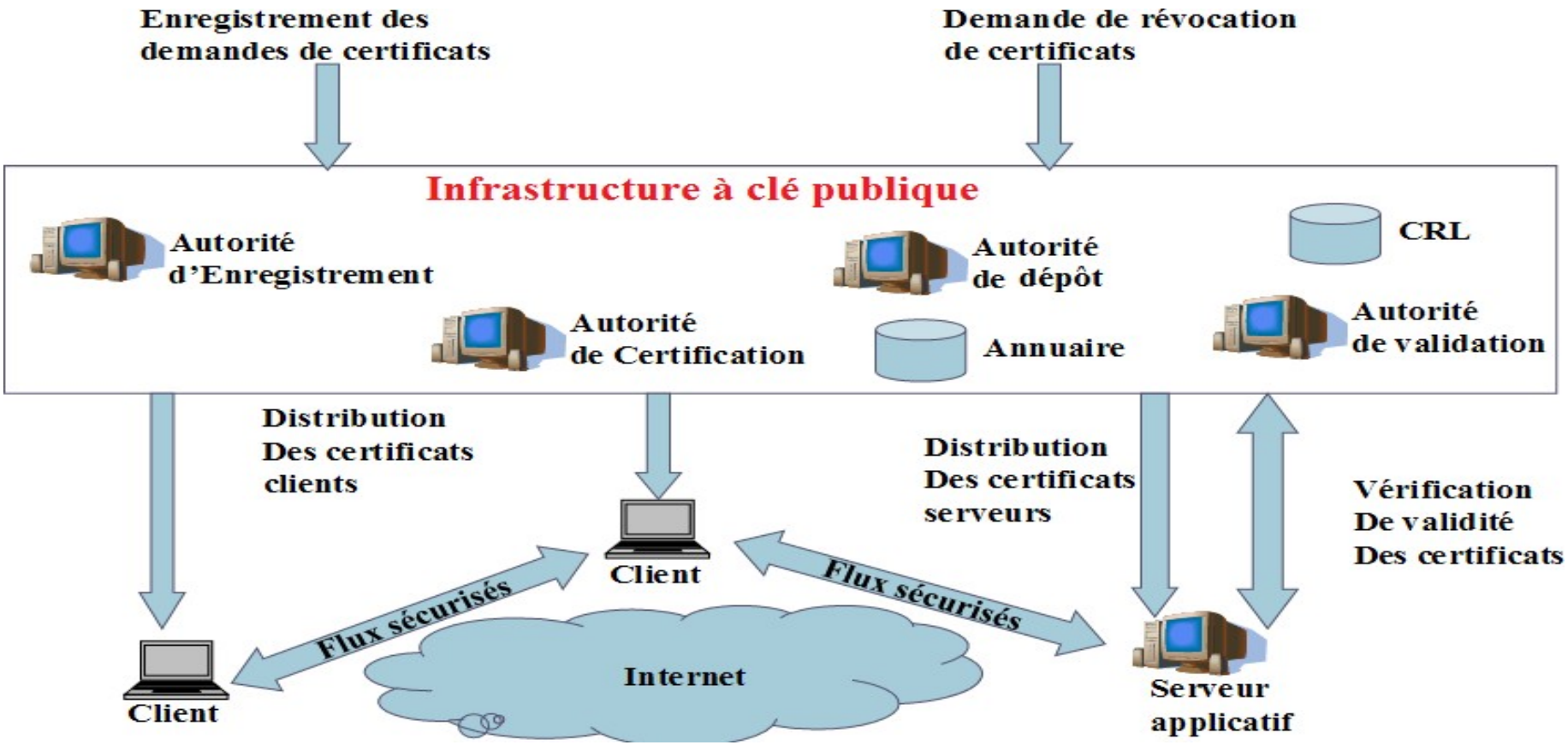
Révocation : lorsque la clé privée correspondante à la clé publique présentée dans le certificat n'est plus secrète.

Plan du chapitre 4 (1/2)

Cryptographie

1. Introduction
2. La cryptographie
   - Confidentialité des messages
     - Chiffrement symétrique
     - Chiffrement asymétrique
     - Chiffrement mixte ou hybride
   - Authentification, Non-répudiation et Intégrité des messages
     - Fonctions de hachage
     - MAC – Message Authentication Code
     - Signature numérique
   - **Gestion de clés**
     - Distribution des clés
     - Protocole d'échange de clés de Diffie-Hellman
     - Certificat de clé publique
     - **PKI – Public Key Infrastructure**
3. La cryptanalyse
   - Attaque sur texte chiffré seul
   - Attaque à texte clair connu
   - Attaque à texte clair choisi
   - Attaque à texte chiffré choisi

Ensemble de composants, fonctions et procédures dédiés à la gestion de clés et de certificats utilisés par des services de sécurité basés sur la cryptographie à clé publique.



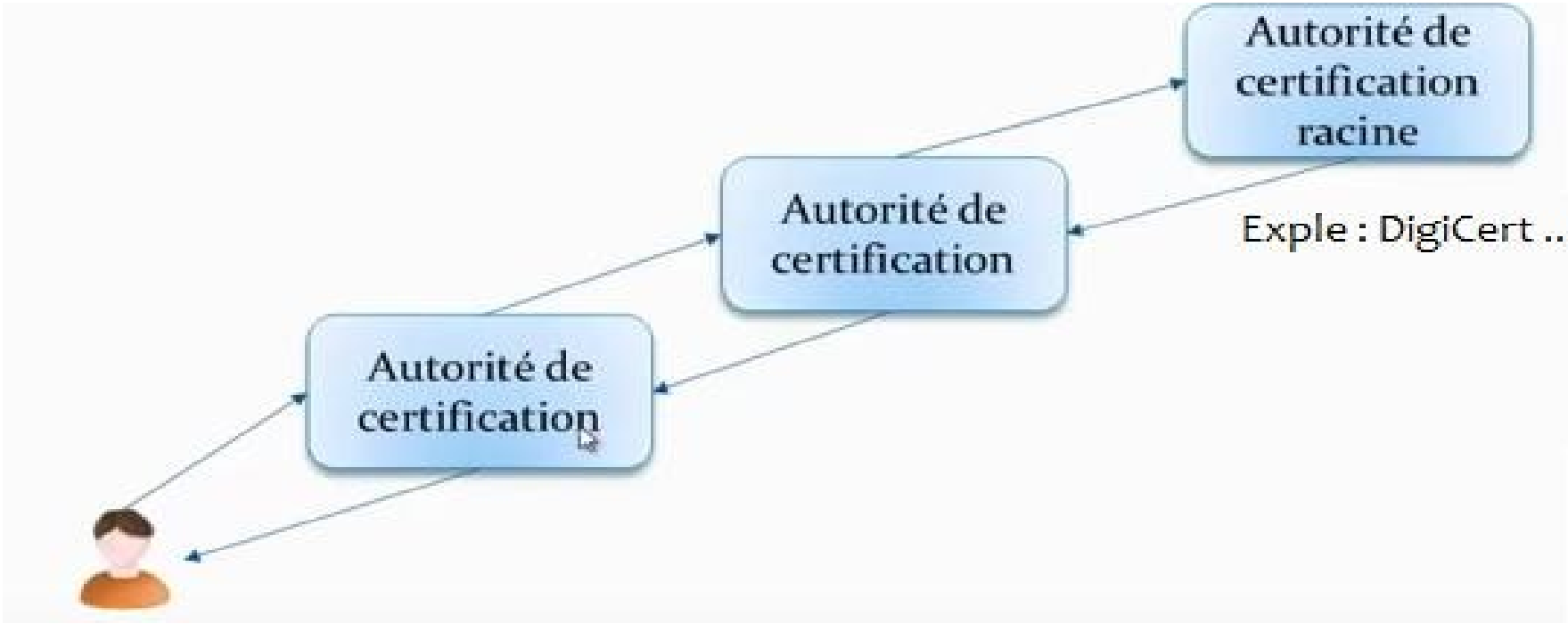
Les principales fonctions d'une PKI :

- Enregistrer et vérifier les demandes de certificats
  - Autorité d'enregistrement
- Créer et distribuer des certificats
  - Autorité de certification
- Vérification de validité de certificats
  - Autorité de validation
- Gérer à tout moment l'état des certificats et prendre en compte leur révocation
  - Autorité de dépôt : Dépôt de listes de certificats révoqués (CRL, Certificate Revocation List)
- Publier les certificats dans un dépôt
  - Autorité de dépôt : Dépôt de certificats (Annuaire)

Chaine de confiance dans les PKIs

Délégation de pouvoir de certification :

Une CA peut déléguer le pouvoir de certification à d'autres entités qui deviennent CA à leur tour, en leur fournissant un certificat qui certifie leur capacité d'être CA.



PLAN DU CHAPITRE 4 (1/2)

CRYPTOGRAPHIE

1. Introduction
2. La cryptographie
   - Confidentialité des messages
     - Chiffrement symétrique
     - Chiffrement asymétrique
     - Chiffrement mixte ou hybride
   - Authentification, Non-répudiation et Intégrité des messages
     - Fonctions de hachage
     - MAC – Message Authentication Code
     - Signature numérique
   - Gestion de clés
     - Distribution des clés
     - Protocole d'échange de clés de Diffie-Hellman
     - Certificat de clé publique
     - PKI – Public Key Infrastructure
3. La cryptanalyse
   - Attaque sur texte chiffré seul
   - Attaque à texte clair connu
   - Attaque à texte clair choisi
   - Attaque à texte chiffré choisi

Plan du chapitre 4 (1/2)

Cryptographie

1. Introduction
2. La cryptographie
   - Confidentialité des messages
     - Chiffrement symétrique
     - Chiffrement asymétrique
     - Chiffrement mixte ou hybride
   - Authentification, Non-répudiation et Intégrité des messages
     - Fonctions de hachage
     - MAC – Message Authentication Code
     - Signature numérique
   - Gestion de clés
     - Distribution des clés
     - Protocole d'échange de clés de Diffie-Hellman
     - Certificat de clé publique
     - PKI – Public Key Infrastructure
3. La cryptanalyse
   - Attaque sur texte chiffré seul
   - Attaque à texte clair connu
   - Attaque à texte clair choisi
   - Attaque à texte chiffré choisi

- En théorie de l'information, Shannon définit deux notions que tout bon crypto-système doit posséder : la confusion et la diffusion.

- **Confusion** : C'est le fait que la méthode de calcul du message crypté à partir du message clair doit être suffisamment complexe. C'est-à-dire qu'il ne doit pas exister de relation simple entre les bits du message clair et les bits du message crypté.

- **Diffusion** : C'est le fait qu'une différence, même minime, entre deux messages clairs doit entraîner une très grande différence entre les messages cryptés. Ainsi, chaque bit du message clair doit contribuer au calcul de chaque bit du message crypté.

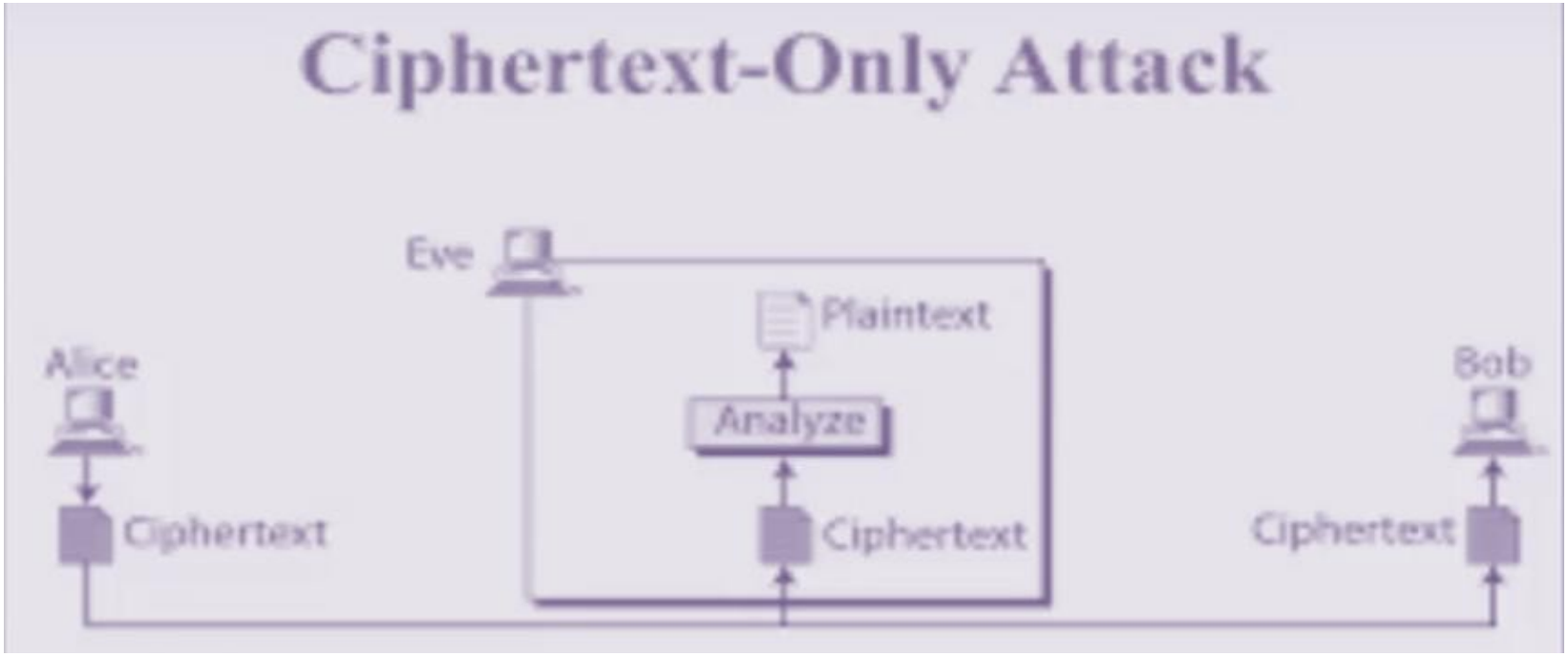
- La cryptanalyse est l'ensemble des méthodes et procédés de décryptage visant à rétablir en clair un cryptogramme, sans connaissance préalable de la clé de chiffrement.
- Plus généralement, la cryptanalyse est utilisée pour étudier la sécurité des procédés de chiffrement utilisés en cryptographie et mettre à l'épreuve leur robustesse et efficacité.
- Le processus par lequel on tente de comprendre un message en particulier est appelé une attaque. Une attaque est souvent caractérisée par les données qu'elle nécessite.
- Les attaques de cryptanalyse peuvent être, généralement, regroupées en quatre familles différentes :
  - Attaque sur texte chiffré seul (ciphertext-only en anglais)
  - Attaque à texte clair connu (known-plaintext attack en anglais)
  - Attaque à texte clair choisi (chosen-plaintext attack en anglais)
  - Attaque à texte chiffré choisi (chosen-ciphertext attack en anglais)

Plan du chapitre 4 (1/2)

Cryptographie

1. Introduction
2. La cryptographie
   - Confidentialité des messages
     - Chiffrement symétrique
     - Chiffrement asymétrique
     - Chiffrement mixte ou hybride
   - Authentification, Non-répudiation et Intégrité des messages
     - Fonctions de hachage
     - MAC – Message Authentication Code
     - Signature numérique
   - Gestion de clés
     - Distribution des clés
     - Protocole d'échange de clés de Diffie-Hellman
     - Certificat de clé publique
     - PKI – Public Key Infrastructure
3. La cryptanalyse
   - **Attaque sur texte chiffré seul**
   - Attaque à texte clair connu
   - Attaque à texte clair choisi
   - Attaque à texte chiffré choisi

- Le cryptanalyste connait l'algorithme utilisé pour le chiffrement et ne dispose que de quelques exemplaires de messages chiffrés, sans avoir d'informations sur leur signification en clair. Son objectif est de collecter suffisamment de messages chiffrés afin d'en déduire la clé. Il s'agit de l'attaque la plus difficile vu le manque d'informations à disposition.



❑ Exemples :

- Les algorithmes de chiffrement symétrique par flux tq RC4 sont vulnérables à ce genre d'attaque lorsqu'ils sont utilisés plusieurs fois de suite avec la même clé secrète.

- Wired Equivalent Privacy (WEP) – le premier protocole de sécurité utilisé pour le Wi-Fi – a été prouvé être vulnérable à plusieurs attaques de cryptanalyse dont la majorité sont de type texte chiffré seul.

- Techniques utilisées dans une attaque de type texte chiffré seul :
  - Attaque par force brute
  - Analyse fréquentielle

Attaque par force brute

- L'attaque par force brute consiste à tester toutes les solutions possibles de clés. C'est le seul moyen de récupérer la clé dans les crypto-systèmes les plus modernes et encore inviolés comme AES. Cette technique est très coûteuse en temps de calcul surtout pour des clés de taille assez grande.

- **Exemple :**

  En écoutant la communication entre Alice et Bob, Eve a pu intercepter le texte chiffré suivant : UVACLYFZLJBYL. Eve essayera alors toutes les clés possibles (toutes les combinaisons possibles de lettres de 1 à 25) jusqu'à trouver une séquence de caractères intelligible (texte en clair).

Key	Plaintext
1	TUZBKXEYKIAXK
2	STYAJWDXJHZWJ
3	RSXZIVCWIGYVI
4	QRWYHUBVHFXUH
5	PQVXGTAUGEWTG
6	OPUWFSZTFDVSF
7	**NOTVERYSECURE**

Analyse fréquentielle

- L'analyse fréquentielle, découverte au 9ème siècle par Al-Kindi, examine les répétitions des lettres du message chiffré afin de trouver la clé. Elle est inefficace contre les chiffrements modernes tels que DES, RSA. Elle est principalement utilisée contre les chiffrements mono-alphabétiques qui substituent chaque lettre par une autre.

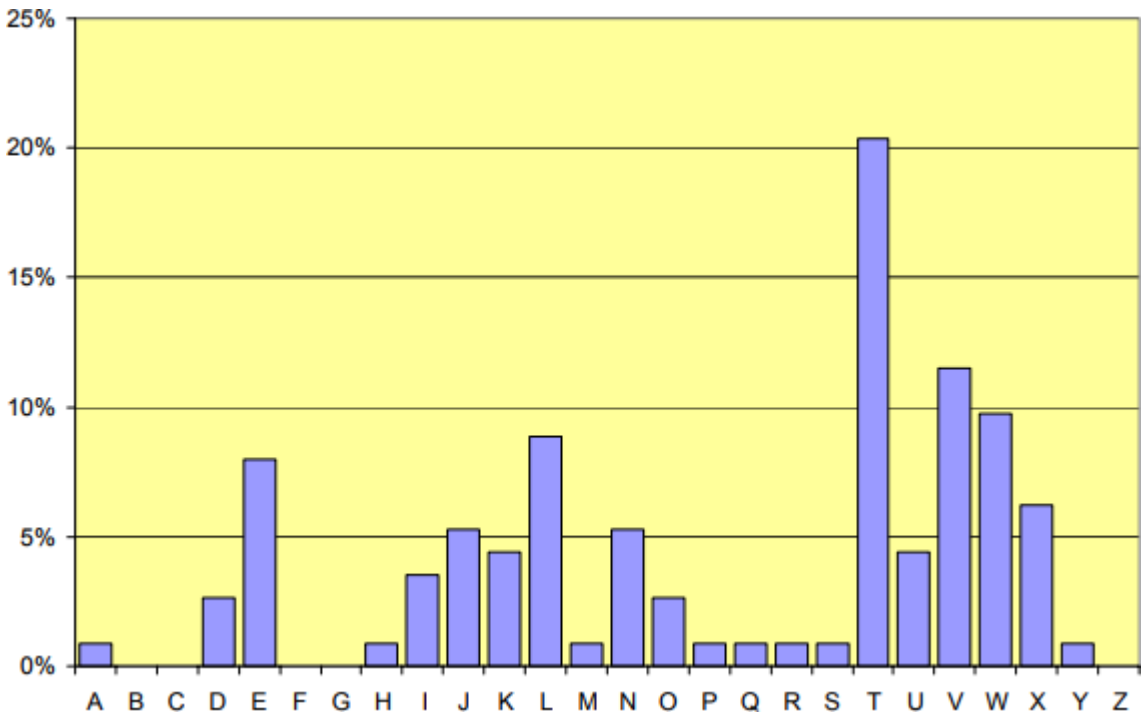
- **Exemple :**

Texte chiffré intercepté

JTVMNKKTVLDEVVTLWTWITKTXUTLWJ
ERUTVTWTHDXATLIUNEWV.
JTVIEVWELOWENLVVNOEDJTVLTPTXYT
LWTWUTSNLITTVQXTVXUJXWEJEWTON
KKXLT.

Analyse fréquentielle

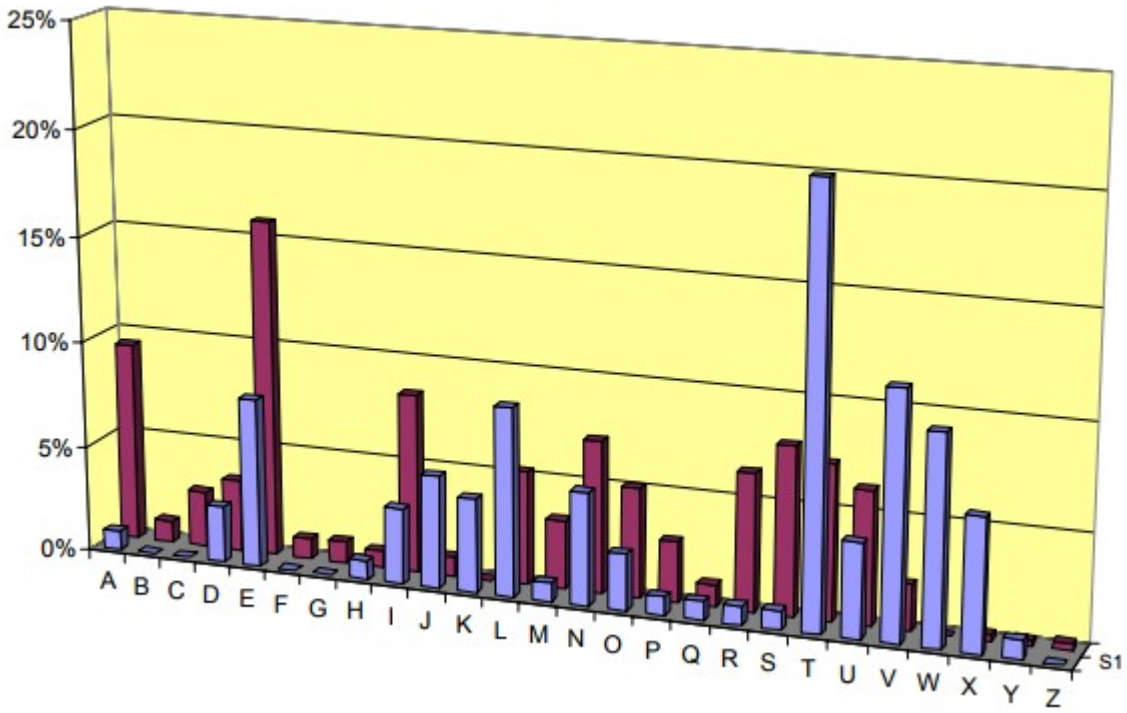
- Exemple (suite) :



Analyse fréquentielle

- **Exemple (suite) :**

Comparaison de fréquence



Analyse fréquentielle

- **Exemple (suite) :**

Déchiffrement

« Les hommes naissent et demeurent
libres et égaux en droits.
Les distinctions sociales ne peuvent être
fondées que sur l'utilité commune. »

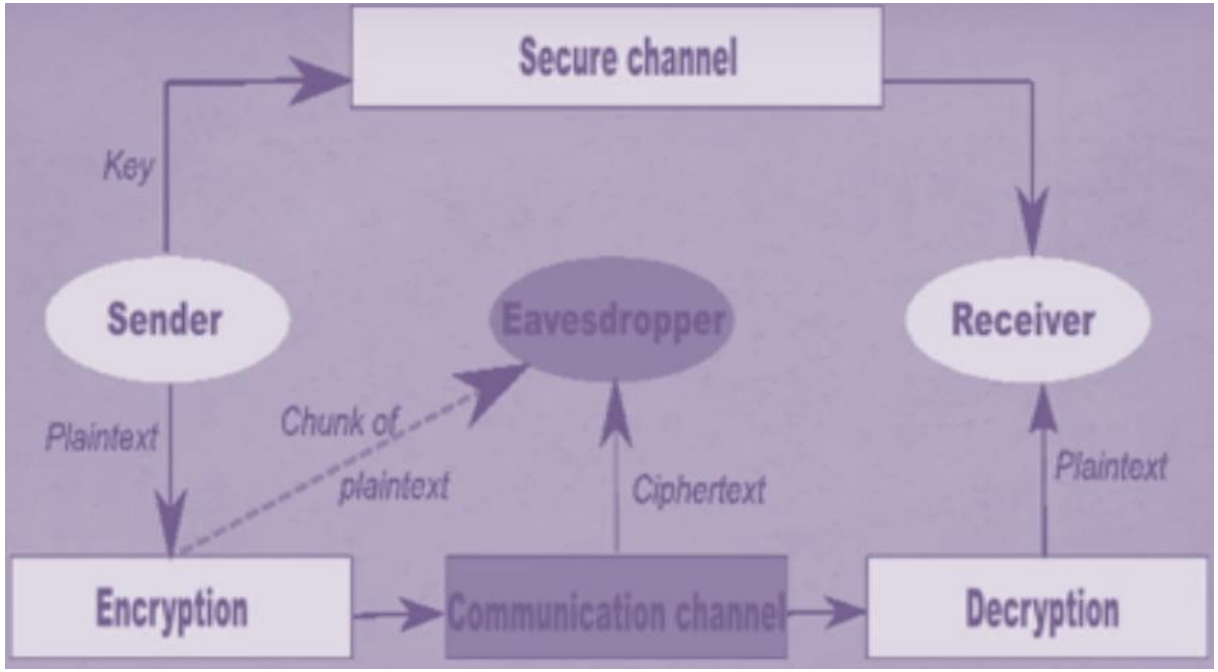
a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
D	R	O	I	T	S	H	M	E	F	G	J	K	L	N	P	Q	U	V	W	X	Y	Z	A	B	C

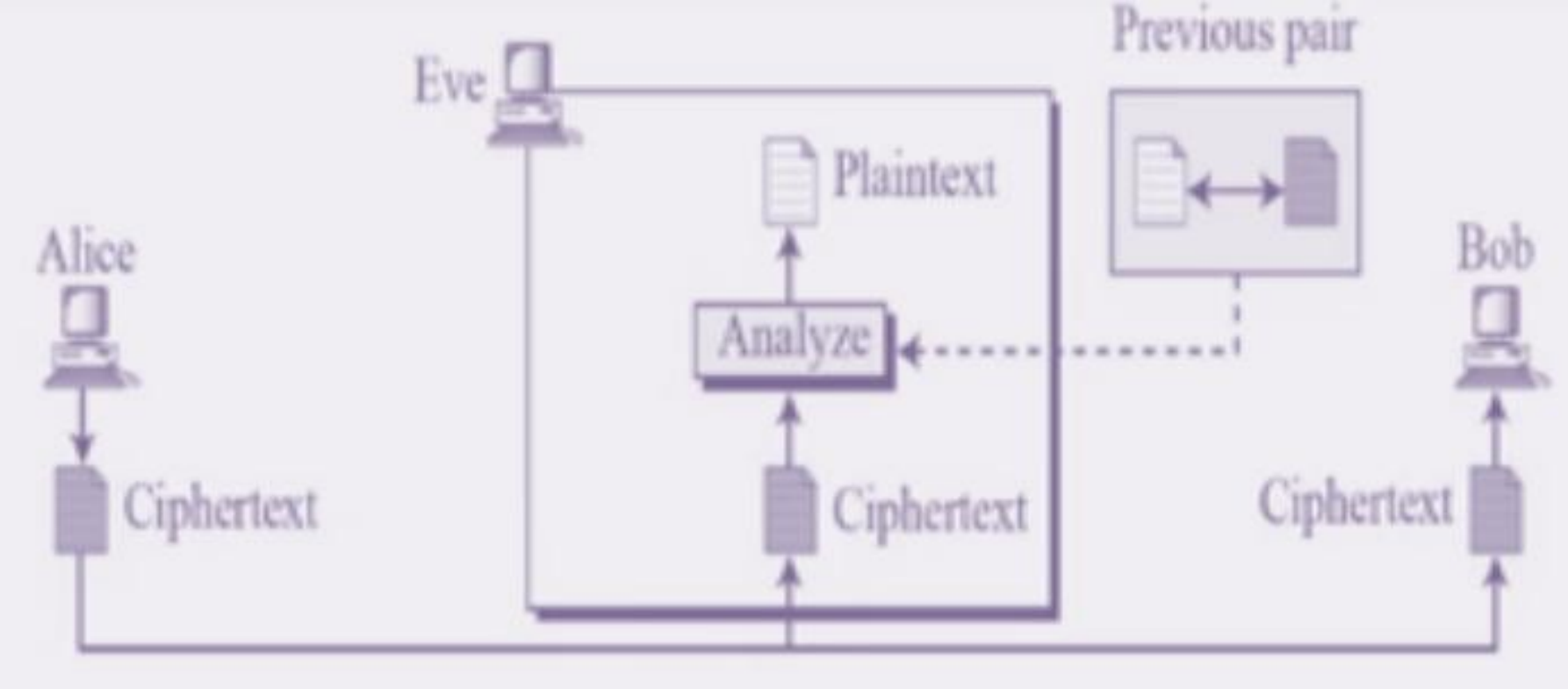
Plan du chapitre 4 (1/2)

Cryptographie

1. Introduction
2. La cryptographie
   - Confidentialité des messages
     - Chiffrement symétrique
     - Chiffrement asymétrique
     - Chiffrement mixte ou hybride
   - Authentification, Non-répudiation et Intégrité des messages
     - Fonctions de hachage
     - MAC – Message Authentication Code
     - Signature numérique
   - Gestion de clés
     - Distribution des clés
     - Protocole d'échange de clés de Diffie-Hellman
     - Certificat de clé publique
     - PKI – Public Key Infrastructure
3. La cryptanalyse
   - Attaque sur texte chiffré seul
   - **Attaque à texte clair connu**
   - Attaque à texte clair choisi
   - Attaque à texte chiffré choisi

- Le cryptanalyste connait l'algorithme de chiffrement utilisé et a accès aux textes chiffrés de plusieurs messages (qu'il n'a pas choisis) ainsi qu'à leurs textes clairs correspondants. Son objectif est de retrouver la clé qui a été utilisée pour chiffrer ces messages ou un algorithme qui permet de déchiffrer d'autres messages chiffrés avec la même clé.




Previous pair
Eve
Plaintext
Alice
Analyze
Bob
Ciphertext
Ciphertext
Ciphertext

Exemples (1/4)

- Les chiffres basés sur l'opération binaire XOR comme fonction d'encryptage et utilisant une clé répétée sont vulnérables aux attaques de type texte clair connu.

  La clé secrète est répétée autant de fois que nécessaire jusqu'à avoir la même taille que le texte à chiffrer. Chaque lettre de la clé répétée (Key Stream) et du texte en clair est remplacée par son code ASCII codé sur 8 bits (1 octet).

$$\text{CipherText} = \text{PlainText} \oplus \text{KeyStream}$$

$$\text{PlainText} = \text{CipherText} \oplus \text{KeyStream}$$

$$\text{KeyStream} = \text{CipherText} \oplus \text{PlainText}$$

En examinant la clé répétée obtenue, on peut déduire la clé secrète utilisée.

Exemples (2/4)

- Le chiffrement par substitution monoalphabétique – qui consiste à substituer chaque lettre du message à chiffrer par une autre – peut être facilement cassé par une attaque de type texte clair connu.

Exple : Chiffre de césar

- ✓ Fonctionne par décalage des lettres de l'alphabet.
- ✓ Le paramètre de décalage (noté généralement n) est la clé de chiffrement.

$$LETTRE_{CipherText} = (LETTRE_{PlainText} + n) \bmod 26$$
$$LETTRE_{PlainText} = (LETTRE_{CipherText} - n) \bmod 26$$

$$n = LETTRE_{CipherText} - LETTRE_{PlainText}$$

Exemples (3/4)

- Les chiffres affines sont également vulnérables aux attaques à texte en clair (Exple : chiffre de Hill). L'idée dans ces chiffres est d'utiliser comme fonction de cryptage une fonction affine du type **y = ax + b**, où :
  - ✓ a et b sont des constantes comprises entre 0 et 25. a doit être premier avec 26 (i.e, a doit avoir une de ces valeurs 1,3,5,7,9,11,15,17,19,21,23,25)
  - ✓ x et y sont, respectivement, les rangs dans l'alphabet (nombres entre 0 et 25 modulo 26) des lettres du texte en clair et de son chiffré correspondant.
- La paire (a,b) constitue la clé secrète.

Exemples (3/4)

- Pour casser un chiffre affine, on procède comme suit :

We have,

Ciphertext: PQ

Plaintext: if

Formula:

$Y = aX + b$ (Y is cipher, X is plain)

Two equations:

$16 = a.9 + b$ (i ⟶ P)

$17 = a.6 + b$ (f ⟶ Q)

$16 = a.9 + b$

$17 = a.6 + b$

$-1 = a.3 + 0$ (by subtracting)

$25 \pmod{26} = a.3$

$a = 25/3$

$a = 25.(3^{-1})$

$a = 25.9$ as $3^{-1} \pmod{26} = 9$ using Euclidean algorithm

$a = 17 \pmod{26}$

Also by solving two equations we get,

$b = 9 \pmod{26}$

So, we get the final equation $Y = 17.X + 9$

Key is (17,9)

Exemples (4/4)

- Pour les chiffrements modernes, une des principales techniques d'attaque à texte clair connu est la cryptanalyse linéaire.

Cryptanalyse linéaire

- La cryptanalyse linéaire est une attaque à texte clair connu contre les protocoles de cryptographie dont la confusion est faible. Inventée par Mitsuru Matsui en 1993, elle a été développée à l'origine pour casser l'algorithme de chiffrement symétrique DES. Les algorithmes plus récents (AES, RSA …) sont insensibles à cette attaque.
- La cryptanalyse linéaire consiste à simplifier l'algorithme de chiffrement en faisant une approximation linéaire.
- En augmentant le nombre de couples (texte en clair,texte chiffré) disponibles, on améliore la précision de l'approximation et on peut en extraire la clé.
- Matsui a effectué une cryptanalyse de DES avec un taux de succès de 88% en utilisant 2^{26} couples (clair,chiffré) en 20 secondes sur un DES à 8 tours. Il a généralisé cette méthode à un Des complet en connaissant 2^{47} couples (clair,chiffré) et en déduisant 14 bits de la clé secrète, ce qui permet de réduire le temps d'une cryptanalyse exhaustive qui demanderait d'explorer les 2^{56} clés possibles.

Cryptanalyse linéaire

❑ Méthodologie de cette attaque (1/3) :

- La méthode consiste à chercher une expression linéaire fonction de **u** bits du texte en clair, de **v** bits en sortie de l'avant dernier tour et de bits des clés de tours intermédiaires, qui a une forte (ou faible) probabilité d'occurrence.

$$X_{i1} \oplus X_{i2} \oplus \ldots \oplus X_{iu} \oplus Y_{j1} \oplus Y_{j2} \oplus \ldots \oplus Y_{jv} = \sum K \quad \text{avec} \quad \sum K = 0 \text{ ou } 1 \quad (1)$$

- **N.B :** Si on choisit aléatoirement **u+v** bits indépendants et qu'on calcule la probabilité associée à l'équation (1), celle-ci devrait être exactement égale à ½.
- C'est l'écart par rapport à cette valeur de référence – ou le biais – qui est exploité par la cryptanalyse linéaire. Plus la probabilité associée à l'équation (1) est loin de ½ (en valeur absolue), meilleure sera la cryptanalyse.

Cryptanalyse linéaire

❑ Méthodologie de cette attaque (2/3) :

- Pour construire une approximation linéaire du chiffre :

1. On considère le seul composant non-linéaire du chiffre: ses boites-S. En énumérant les propriétés non linéaires des boîtes-S, on peut mettre en exergue certaines approximations linéaires entre certains bits d'entrée et certains bits de sortie.
2. On concatène les approximations linéaires obtenues des différentes boîtes-S de telle sorte que les bits intermédiaires des clés de tours puissent être éliminés laissant une expression qui présente un fort biais et qui ne soit fonction que des bits du clair et des bits d'entrée du dernier tour. La probabilité de l'expression linéaire résultante est calculée selon le lemme de Matsui suivant :

Soient X_1 , X_2 , ... , X_n n variables aléatoires binaires indépendantes :

$P(X_1 \oplus X_2 \oplus \ldots \oplus X_n = 0) = \frac{1}{2} + 2^{n-1} \prod_{i=1}^{n} \varepsilon_i$ Où ε_i est le biais de chaque X_i

Cryptanalyse linéaire

❑ Méthodologie de cette attaque (3/3) :

- Une fois qu'on a approché linéairement les (r-1) tours d'un système cryptographique à r rondes avec un biais suffisant, on peut tenter de retrouver la clé du dernier tour. En effet, pour chaque valeur possible des bits cherchés de la dernière clé de tour, on détermine l'entrée de la boîtes-S du dernier tour.
- On effectue cette opération pour tous les couples (clair,chiffré) connus et on compte le nombre de fois où l'expression linéaire déterminée est vérifiée.
- La valeur de la clé qui satisfait l'expression le plus souvent est la valeur la plus probable de la clé cherchée.
- Pour découvrir les autres clés intermédiaires, on attaque l'algorithme en remontant progressivement dans les tours jusqu'à arriver à la première clé.

Cryptanalyse linéaire

❑ Exemple (1/2) :

On considère le système

$$\begin{array}{ccccccccccccccc} & & k^1 & & & & k^2 & & & & k^3 & & & & \\ & & \downarrow & & & & \downarrow & & & & \downarrow & & & & \\ x \to & \oplus & \to & u^1 & \xrightarrow{S} & v^1 \to & \oplus & \to & u^2 & \xrightarrow{S} & S(u^2) \to & \oplus & \to & y \end{array}$$

où les x représentent 3 bits. La boîte S est une substitution sur $\mathbf{F}_2^3$.

On suppose que le premier bit d'une boîte S est conservé avec probabilité supérieure à $1/2$, c'est-à-dire

$$(0,*,*) \xrightarrow{S} (0,*,*) \quad \text{et} \quad (1,*,*) \xrightarrow{S} (1,*,*)$$

Cela veut dire que

$$P(u_1^1 + v_1^1 = 0) = 1/2 + \epsilon$$

Cryptanalyse linéaire

- **Exemple (2/2) :**

Ainsi, on déduit que :

$$x_1 + u_1^2 = u_1^1 + v_1^1 + k_1^1 + k_1^2 = k_1^1 + k_1^2$$

Si l'on considère des cryptogrammes tous chiffrés par la même clé, la variable $k_1^1 + k_1^2$ a donc une valeur fixée à 0 ou 1.

Si $k_1^1 + k_1^2 = 0$, alors $P(x_1 + u_1^2 = 0) = P(x_1 + k_1^1 + u_1^2 + k_1^2 = 0) = 1/2 + \epsilon$

Si $k_1^1 + k_1^2 = 1$, alors $P(x_1 + u_1^2 = 1) = 1 - P(x_1 + k_1^1 + u_1^2 + k_1^2 = 0) = 1/2 - \epsilon$

Comme $P(x_1 + u_1^2 = 0) > P(x_1 + u_1^2 = 1)$

L'approximation linéaire du chiffre est : $x_1 + u_1^2 = 0$

Pour des clairs-chiffrés (x, y) donnés, et pour chaque valeur possible κ de k^3, on calcule $u_1^2 = \left(S^{-1}(y + \kappa)\right)_1$. On compte le nombre de fois où κ satisfait l'expression $x_1 + \left(S^{-1}(y + \kappa)\right)_1 = 0$

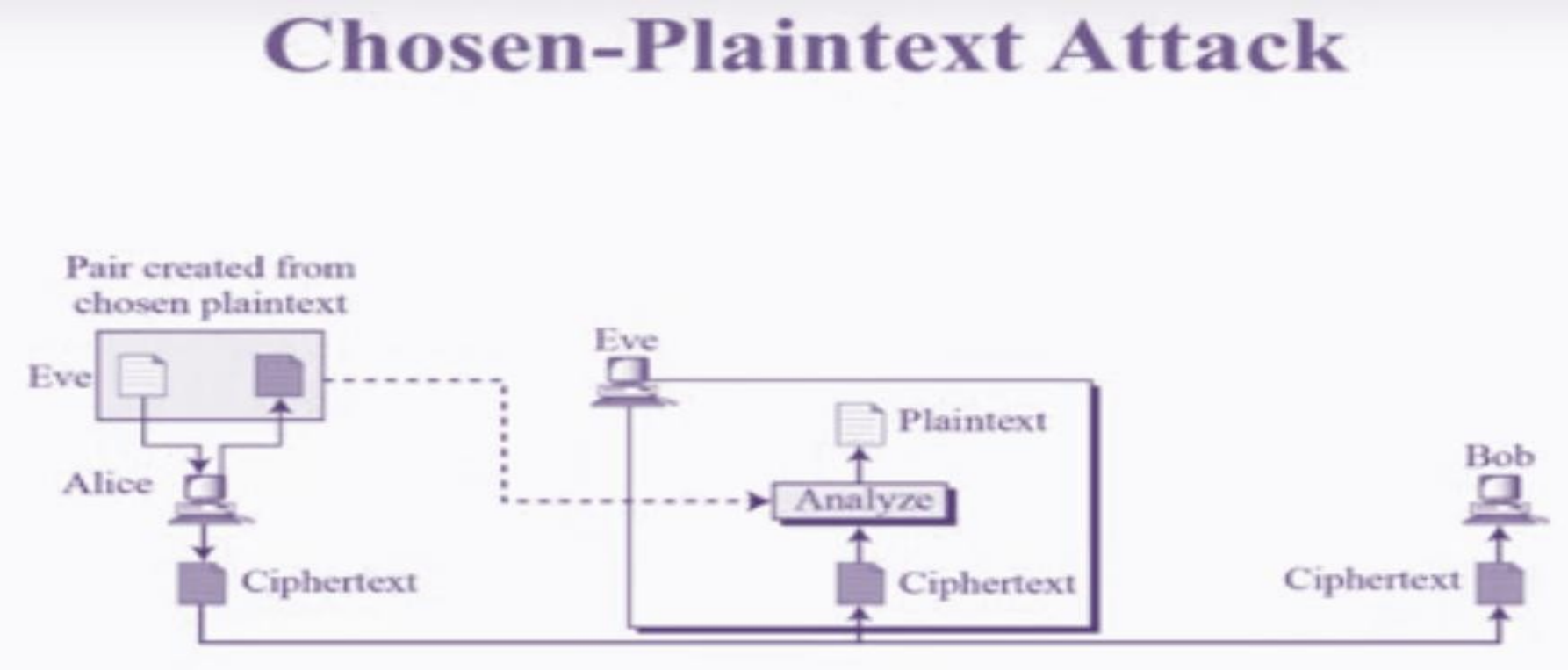
La valeur κ qui satisfait l'expression le plus souvent est la valeur la plus **probable** pour k^3.

Plan du chapitre 4 (1/2)

Cryptographie

1. Introduction
2. La cryptographie
   - Confidentialité des messages
     - Chiffrement symétrique
     - Chiffrement asymétrique
     - Chiffrement mixte ou hybride
   - Authentification, Non-répudiation et Intégrité des messages
     - Fonctions de hachage
     - MAC – Message Authentication Code
     - Signature numérique
   - Gestion de clés
     - Distribution des clés
     - Protocole d'échange de clés de Diffie-Hellman
     - Certificat de clé publique
     - PKI – Public Key Infrastructure
3. La cryptanalyse
   - Attaque sur texte chiffré seul
   - Attaque à texte clair connu
   - **Attaque à texte clair choisi**
   - Attaque à texte chiffré choisi

- Le cryptanalyste connait l'algorithme de chiffrement utilisé et a non seulement accès aux textes chiffrés et aux textes clairs correspondants, mais de plus il peut choisir les textes en clair.
- Cette attaque est plus efficace que l'attaque à texte clair connu, car le cryptanalyste peut choisir des textes en clair spécifiques qui donneront plus d'informations sur la clé.
- Les attaques de type texte clair choisi deviennent plus intéressantes et plus efficaces dans un contexte de chiffrement à clé publique.
- Un système cryptographique est dit incassable s'il peut résister à une attaque du type texte en clair choisi.
- La cryptanalyse différentielle est une des principales techniques d'attaque à clair choisi.


Chosen-Plaintext Attack
Pair created from chosen plaintext
Eve
Alice
Ciphertext
Eve
Plaintext
Analyze
Ciphertext
Bob
Ciphertext

Cryptanalyse différentielle

- La cryptanalyse différentielle a été découverte par Eli Biham et Adi Shamir en 1991.
- Le principe général de cette attaque consiste à considérer des couples de clairs X et X′ présentant une différence ΔX fixée et à étudier la propagation de cette différence initiale à travers le chiffrement.
- Dans cette cryptanalyse, on exploite le fait suivant : certaines occurrences de différences entre des clairs et certaines occurrences de différences entre leurs chiffrés correspondants ont une forte probabilité d'apparition.
- Les différences sont définies par une loi de groupe, en général le XOR bit à bit.
- La cryptanalyse différentielle utilise la comparaison du XOR de deux entrées avec le XOR des deux sorties correspondantes.

Cryptanalyse différentielle

- On considère :
  - $x' = (x'_1, x'_2 \ldots x'_n)$ et $x'' = (x''_1, x''_2 \ldots x''_n)$ deux entrées
  - $y' = (y'_1, y'_2 \ldots y'_n)$ et $y'' = (y''_1, y''_2 \ldots y''_n)$ les sorties correspondantes

  On note : $\Delta x = x' \oplus x''$ et $\Delta y = y' \oplus y''$

- Si le système cryptographique était parfait alors la probabilité pour qu'un Δy particulier provienne d'un Δx précis devrait être de $1/2^n$ où n est le nombre de bits de X.

- La cryptanalyse différentielle exploite le fait qu'il peut arriver qu'un Δy particulier apparaît pour un certain Δx avec une très grande probabilité, $p >> 1/2^n$.

- Le couple $(\Delta x, \Delta y)$ est appelée une différentielle.

- Il s'agit donc de construire les différentielles $(\Delta x, \Delta y)$ les plus probables pour un système cryptographique donné et ce en examinant les propriétés de ses boîtes-S.

Cryptanalyse différentielle

❑ Méthodologie de cette attaque (1/2) :

- On examine, pour chaque boîte-S du système cryptographique, toutes les différentielles (Δx, Δy) et on détermine la probabilité d'apparition de Δy étant donné Δx.
- En énumérant les différentes possibilités des différentielles, on peut construire une table de différentielles qui synthétise toutes les valeurs possibles. Dans cette table, la somme de tous les éléments d'une ligne ou d'une colonne vaut 2^n.
- **N.B** : Biham et Shamir ont prouvé que les bits des clés de tours n'ont aucune influence sur les différentielles (voir exemple).
- Il faut ensuite réussir à combiner les informations obtenues sur l'ensemble du système cryptographique en une seule caractéristique différentielle ayant une probabilité égale au produit des probabilités des caractéristiques différentielles des différentes boîtes-S du système.

Cryptanalyse différentielle

Méthodologie de cette attaque (2/2) :

- Une fois qu'on a trouvé une caractéristique différentielle de r-1 rondes d'un système cryptographique à r rondes avec une probabilité suffisante, on peut tenter de retrouver des bits de la clé du dernier tour en procédant ainsi :
  1. Choisir une différentielle d'entrée appropriée.
  2. Construire un nombre adéquat de couples de clairs qui mènent à cette différentielle d'entrée, les chiffrer et ne conserver que les paires de ciphers correspondantes.
  3. Pour chaque différentielle d'entrée choisie, construire la différentielle de sortie correspondante à la sortie de la caractéristique différentielle.
  4. Pour chaque valeur possible de la clé du dernier tour, on compte le nombre de fois où la caractéristique différentielle déterminée est satisfaite. La valeur maximale du compteur indique normalement la clé la plus probable. Cette opération est effectuée en considérant un grand nombre de couples (clair,chiffré).
- Pour découvrir les autres clés intermédiaires, on attaque l'algorithme en remontant progressivement dans les tours jusqu'à arriver à la première clé.

Cryptanalyse différentielle

- **Exemple (1/2) :**

N.B : Δx et Δy sont notées dans cet exemple par α et β, respectivement.

On considère le système

$$x \to \underset{\downarrow k^1}{\oplus} \to x^0 \xrightarrow{S} S(x^0) \to \underset{\downarrow k^2}{\oplus} \to x^1 \xrightarrow{S} S(x^1) \to \underset{\downarrow k^3}{\oplus} \to y$$

où les x représentent trois bits. La boîte S est une substitution sur $\mathbf{F}_2^3$.

On suppose que

$$P\left(S(x) + S(x') = \beta \mid x + x' = \alpha\right) > 1/8$$

Si $x + x' = \alpha$, on a $x^0 + x^{0'} = (x + k^1) + (x' + k^1) = x + x' = \alpha$,

et la probabilité que $S(x^0) + S(x^{0'}) = \beta$ est importante.

Cryptanalyse différentielle

❑ Exemple (2/2) :

N.B : Δx et Δy sont notées dans cet exemple par α et β, respectivement.

Si c'est réalisé, $x^1 + x^{1'} = \left(S(x^0) + k^2\right) + \left(S(x^{0'}) + k^2\right) = S(x^0) + S(x^{0'}) = \beta$

D'où

$$P_{\alpha,\beta} = P\left(x^1 + x^{1'} = \beta \mid x + x' = \alpha\right) > 1/8$$

Pour un couple clairs-chiffrés (x, y) et (x', y') donné tel que $x + x' = \alpha$, et pour chaque valeur possible κ de k^3, on calcule

$$z^1 = S^{-1}(y + \kappa) \quad \text{et} \quad z^{1'} = S^{-1}(y' + \kappa).$$

Si $z^1 + z^{1'} = \beta$, on sait que κ est une valeur **probable** pour k^3.

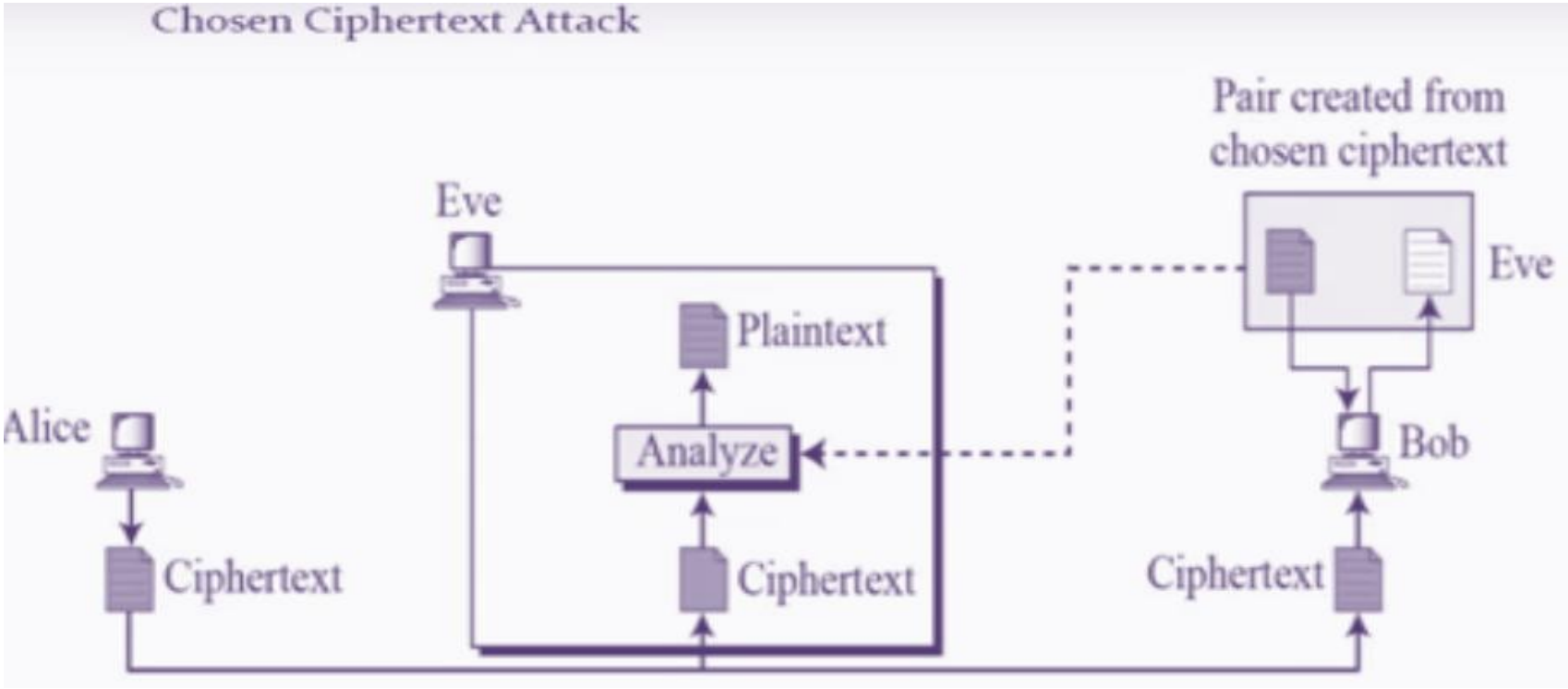
Si $z^1 + z^{1'} \neq \beta$, on sait que κ est une valeur **improbable** pour k^3.

Plan du chapitre 4 (1/2)

Cryptographie

1. Introduction
2. La cryptographie
   - Confidentialité des messages
     - Chiffrement symétrique
     - Chiffrement asymétrique
     - Chiffrement mixte ou hybride
   - Authentification, Non-répudiation et Intégrité des messages
     - Fonctions de hachage
     - MAC – Message Authentication Code
     - Signature numérique
   - Gestion de clés
     - Distribution des clés
     - Protocole d'échange de clés de Diffie-Hellman
     - Certificat de clé publique
     - PKI – Public Key Infrastructure
3. La cryptanalyse
   - Attaque sur texte chiffré seul
   - Attaque à texte clair connu
   - Attaque à texte clair choisi
   - **Attaque à texte chiffré choisi**

- Le cryptanalyste connait l'algorithme de chiffrement utilisé et possède des messages chiffrés de son choix et demande la version en clair de certains de ces messages pour mener l'attaque. Le but est de trouver la clé.



Exemple :

- Attaque à texte chiffré choisi appliquée à RSA :
  - Eve intercepte un cipher C qu'elle désire déchiffrer en écoutant une communication confidentielle entre Alice et Bob.
  - On rappelle que : $C = M^e \bmod n$ et $M = C^d \bmod n$

    avec M le plaintext que Eve cherche à trouver et (e,n) (respectivement, (d,n)) est la clé publique (respectivement, la clé privée de Bob) utilisée pour le chiffrement (respectivement, déchiffrement) des messages envoyés par Alice à Bob (respectivement, envoyés par Bob à Alice) .
  - Ensuite, elle calcule un nouveau cipher C' tel que : $C' = C * r^e \bmod n$

    Avec r un nombre aléatoirement choisi par Eve.
  - Si Eve arrive à récupérer le message déchiffré correspondant à C' , elle pourra ainsi déterminer le message en clair M recherché en divisant le texte chiffré C récupéré initialement par la valeur aléatoire r. En effet :

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

PLAN DU CHAPITRE 4 (2/2)

CRYPTOGRAPHIE

4 La stéganographie

- Stéganographie linguistique
- Stéganographie technique
- Propriétés des systèmes de stéganographie
- Techniques de stéganographie

5 La stéganalyse

- Types d'attaques de stéganalyse
- Méthodes de stéganalyse

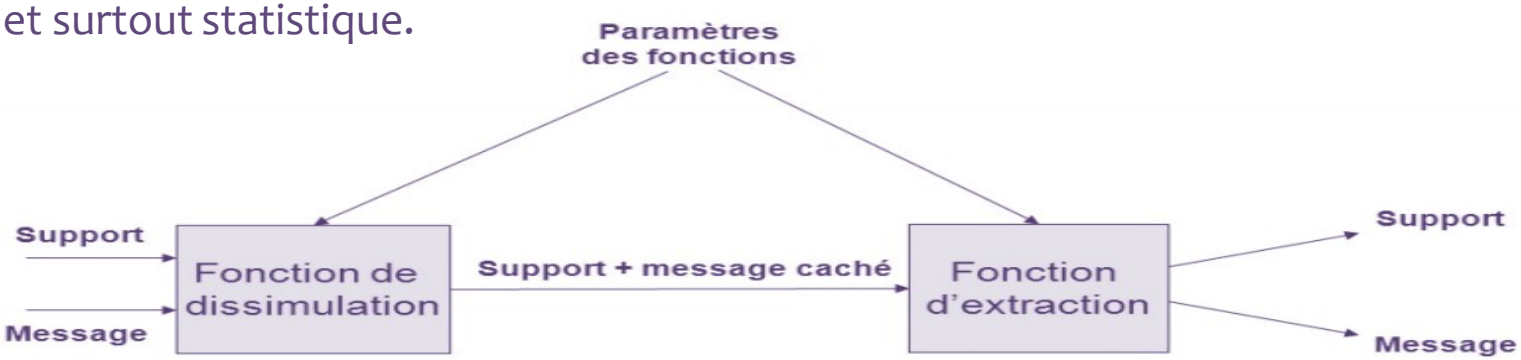
6 Le tatouage numérique

Plan du chapitre 4 (2/2)

Cryptographie

4. La stéganographie
   - Stéganographie linguistique
   - Stéganographie technique
   - Propriétés des systèmes de stéganographie
   - Techniques de stéganographie
5. La stéganalyse
   - Types d'attaques de stéganalyse
   - Méthodes de stéganalyse
6. Le tatouage numérique

- La stéganographie, ou la science de communication secrète, est un procédé permettant de cacher un message secret au sein d'un médium hôte (***original, cover, carrier***) anodin, de telle sorte que le médium résultant (***stégo-médium***) semble inaffecté (dissimulation indétectable) par l'insertion du message secret. L'objectif est de rendre difficile, ou impossible, la distinction entre un médium original et un médium modifié par l'insertion d'un message secret.
- Alors qu'avec la cryptographie, la sécurité repose sur le fait que le message chiffré soit incompréhensible pour les personnes non autorisées, avec la stéganographie, la sécurité repose sur le fait que la présence même d'un message secret ne sera sans doute pas soupçonnée et détectée.
- Le concept clé de la sécurité d'un système de stéganographie est son indétectabilité visuelle mais aussi et surtout statistique.



Principe d'un système stéganographique

Domaine d'utilisation de la stéganographie :

- Communiquer en toute liberté même dans des conditions de censure et de surveillance ;
- Contrebalancer toutes les législations ou barrières possibles empêchant l'usage de la cryptographie ;
- Publier des informations ouvertement mais à l'insu de tous des informations qui pourront ensuite être révélées et dont l'antériorité sera incontestable et vérifiable par tous :

  -Les attestations officielles (de diplômes, par exemple), faites sur du papier spécial, comportant éventuellement un filigrane, des dessins, une signature manuscrite, des tampons…

  -Les procédés nombreux employés pour sécuriser les billets de banque, de telle sorte que les destinataires soient sûrs qu'ils proviennent bien de l'établissement habilité à les émettre et non de quelque faux-monnayeur.

  - …

Il existe deux types de stéganographie :

- La sténographie linguistique
- La stéganographie technique

Plan du chapitre 4 (2/2)

Cryptographie

4. La stéganographie
   - Stéganographie linguistique
   - Stéganographie technique
   - Propriétés des systèmes de stéganographie
   - Techniques de stéganographie
5. La stéganalyse
   - Types d'attaques de stéganalyse
   - Méthodes de stéganalyse
6. Le tatouage numérique

- Dans la stéganographie linguistique, les propriétés linguistiques d'un texte sont modifiées pour cacher l'information. Ses formes les plus connues sont :
  - ***Sémagramme*** : Ce procédé permet de transmettre un message qui n'est pas composé de lettres ou de chiffres mais dont le sens est véhiculé par une combinaison d'objets, de signes ou de symboles. Le système stéganographique échappe donc totalement à l'observateur.
  - ***Acrostiche*** : Ce procédé permet de transmettre des données au travers de lettres initiales dans chaque vers de poème et qui, lus de haut en bas, forment un mot ou une expression.
  - ***Ponctuation :*** L'utilisation de points, hauteur de lettres et virgules par les prisonniers de guerre a également permis de transmettre des messages à leur famille.
  - ***Nulles :*** Les codes camouflés, aussi appelés les nulles, consistent à marquer d'un signe particulier certaines lettres d'un texte (par des piqûres d'aiguilles sur ou sous les lettres). Il suffit alors de rassembler les lettres marquées pour former un mot.
  - ***Insertion d'erreurs :*** Mise en valeur de l'information au travers d'erreurs ou de formes de style dans un texte.
- Ces différents procédés restent néanmoins difficiles et longs à réaliser et laissent vite suspecter la possibilité d'un message dissimulé. De nombreuses censures ont été ainsi appliquées afin de limiter l'usage de ces techniques.

Plan du chapitre 4 (2/2)

Cryptographie

4. La stéganographie
   - Stéganographie linguistique
   - Stéganographie technique
   - Propriétés des systèmes de stéganographie
   - Techniques de stéganographie
5. La stéganalyse
   - Types d'attaques de stéganalyse
   - Méthodes de stéganalyse
6. Le tatouage numérique

- La stéganographie technique (dite aussi stéganographie informatique) regroupe toutes les techniques qui ne jouent pas sur les mots.
- La stéganographie technique est intéressante car elle permet de dissimuler des données dans plusieurs types de supports :
  - Image (BMP, JPEG, GIF…)
  - Vidéo
  - Fichier au format textuel (TXT, HTML…)
  - Audio
  - Programme
  - Disque dur ou CD
- Le support conteneur doit avoir une taille suffisante pour pouvoir accueillir les données secrètes à cacher. Dans certains cas, sa taille initiale va varier, dans d'autres non. Mais ceci n'a strictement aucune importance tant que personne ne peut comparer le médium original et le stégo-médium correspondant.
- Pour les images par exemple, les modifications visuelles sont indécelables tant que la taille des informations cachées ne dépasse pas 20-25% de celle du fichier initial.

- Si le but de la stéganographie est de dissimuler un message secret sans éveiller l'attention humaine, avec la stéganographie informatique il faut également veiller à ne pas éveiller l'attention des logiciels d'analyse.
- Lorsqu'un support informatique est soupçonné de contenir un message stéganographié, on pourra toujours le soumettre à un logiciel d'analyse qui en effectuant une stéganalyse tentera de récupérer le message dissimulé. Pour cela, il faut que le message à cacher soit en tout point comparable à une suite de bits aléatoires.
- La solution est donc de crypter le message secret avant de l'insérer dans le médium de couverture en utilisant une ***stégo-clé***.
- On distingue ainsi entre :
  - ***La stéganographie pure*** : les données secrètes sont dissimulées en clair dans le médium de couverture.
  - ***La stéganographie à clé secrète*** : les données à cacher sont préalablement cryptées avec une clé secrète qui devra être connue pour pouvoir les extraire.
  - ***La stéganographie à clé publique*** : Un système à clé publique/privée est utilisé pour crypter/décrypter les données lors de leur dissimulation/extraction.

Plan du chapitre 4 (2/2)

Cryptographie

4. La stéganographie
   - Stéganographie linguistique
   - Stéganographie technique
   - Propriétés des systèmes de stéganographie
   - Techniques de stéganographie
5. La stéganalyse
   - Types d'attaques de stéganalyse
   - Méthodes de stéganalyse
6. Le tatouage numérique

Les propriétés d'un système de stéganographie sont principalement :

- ***L'imperceptibilité :*** les données ne doivent pas être « perceptibles » dans le stégo-médium. En d'autres termes, il faut assurer une indétectabilité statistique de sorte qu'une personne surveillant le canal de communication ne doit pas pouvoir différencier un médium d'un stégo-médium.
- ***La capacité d'insertion :*** est la quantité de bits significatifs dissimulés dans le stégo-médium par unité d'accès (par exemple, le nombre de bits par seconde en musique). La capacité d'insertion relative est le rapport entre la taille du message secret à dissimuler et la taille du médium de couverture utilisé.
- ***La robustesse :*** correspond à l'aptitude de préservation des données cachées face aux modifications du stégo-médium.

Plan du chapitre 4 (2/2)

Cryptographie

4 La stéganographie

- Stéganographie linguistique
- Stéganographie technique
- Propriétés des systèmes de stéganographie
- Techniques de stéganographie

5 La stéganalyse

- Types d'attaques de stéganalyse
- Méthodes de stéganalyse

6 Le tatouage numérique

Support de couverture : Fichier au format textuel

- La stéganographie sur un support texte est relativement facilement identifiable.
- Deux techniques principales :
  - ***Méthodes des "espaces" en fin de phrases*** :

    Ex: *0 espace en fin de ligne correspondent à 0, 1 espace correspond à 1.*

    Avantage : Très facile et donc très simple à implémenter.

    Inconvénients :

    - Il faut énormément de lignes pour coder peu de texte (8 lignes pour coder 1 octet).

    -Super visible par une personne extérieure qui s'y attend un peu. Et donc facilement manipulable.
  - ***Méthodes des "espaces" entre les mots*** :

    Ex: *Un espace entre 2 mots suivit de deux espaces entre les 2 mots suivants <=> 0, deux espaces entre 2 mots suivit d'un espace entre les 2 mots suivants <=> 1.*

    Avantage : Le rapport texte codé sur texte hôte est beaucoup plus important.

    Inconvénient : *C'est encore plus visible que la méthode précédente.*

Support de couverture : Image

- L'image n'est ni plus ni moins que le stockage dans un fichier de tous les pixels RVB composant l'image finale. Sachant qu'un pixel est constitué de 3 octets : un octet pour la composante rouge, un octet pour la composante verte et un octet pour la composante bleue. Par exemple, une image 800x600 pixels correspond à 800x600x3 = 1440000 octets.
- L'astuce est donc d'utiliser le bit de poids faible à chaque octet RVB qui compose chaque pixel de l'image pour la dissimulation des données secrètes. En effet, en retirant 1 bit, on dégrade l'image, mais ce n'est pas perceptible à l'œil nu...
- Cette technique – largement connue sous le nom de ***LSB (Least Significant Bit)*** – est de loin la technique la plus répandue pour la dissimulation des données dans une image. Son succès provient d'une grande facilité de mise en œuvre, ce qui permet d'en trouver de nombreuses implémentations. Toutefois, elle possède le gros désavantage de modifier de manière significative les statistiques du conteneur.
- Pour les images au format JPEG qui est le format le plus utilisé sur le net, les pixels de l'image sont transformés en coefficients représentant chacun une fréquence particulière contenue dans l'image. Les bits les moins signifiants des coefficients transformés sont utilisés comme des bits redondants pour dissimuler le message secret, ce qui a pour effet d'apporter plus de robustesse contre les attaques de stéganalyse.

Support de couverture : Fichier audio

- Les fichiers audio sont encore une meilleure manière de cacher de l'information grâce à leur taille.
- L'avantage majeur de la stéganographie sur fichiers audio sur celle sur images par exemple est que la quantité d'informations qui peuvent y être cachées est plus importante grâce à la taille du support, ainsi que les techniques de dissimulation qui sont plus nombreuses.
- Avec des fichiers audio, on ne joue plus sur les palettes de couleurs comme avec les images, mais plutôt sur les pistes de son qui peuvent s'inverser.
- Il y a plusieurs techniques de stéganographie sur des fichiers audio :
  - *LBE (Low Bit Encoding)* : cela ressemble à la technique LSB pour les images ;
  - *SSE (Spread Spectrum Encoding)* : on rajoute du bruit aléatoire (dans lequel sera le message) à la mélodie originale ;
  - *EDH (Echo Data Hiding)* : il existe souvent dans les mélodies un écho associé au son original. La distance séparant ces deux sons peut être utilisée pour y coder des informations ;
  - *Masque de perception* : on cache un son derrière un son plus puissant mais de même intensité.

Support de couverture : Page HTML

- Il s'agit d'apporter des modification au fichier source pour camoufler le message secret en insérant des espaces entre balises, variant minuscules et majuscules dans les balises, utilisant les indications qui n'apparaissent pas sur l'écran des navigateurs tq les commentaires et les balises 'META'....
- Astucieux mais cela peut toutefois se détecter facilement par analyse statistique et même par un coup d'œil au source dont l'indentation exotique pourra attirer l'attention.

Support de couverture : Programme

- Dans les "zones mortes" du code (commentaires, branche morte d'un organigramme) ;
- Dans le programme lui-même à l'aide d'une commande jamais utilisée ;

Support de couverture : Disque dur

- Il est également possible de cacher des fichiers à l'intérieur de l'espace libre d'un disque dur ou d'une disquette.
- En effet, pour répertorier tous les fichiers d'un disque, l'OS accède et met à jour une table d'allocation des fichiers : seules les entrées de fichiers figurant dans cette table sont accessibles par le système d'exploitation. L'idée est alors d'inscrire un fichier physique sur le disque dur sans que la table d'allocation en ait connaissance : ainsi le fichier est sur le disque mais le système d'exploitation ne le voit pas. Pour ce faire il suffit d'utiliser un logiciel spécialisé écrivant et lisant directement sur le disque sans passer par le système d'exploitation.
- Ce système de stéganographie est efficace si l'intercepteur n'en a pas conscience, par contre si celui-ci est au courant alors il n'aura aucune difficulté pour retrouver le fichier.
- L'autre point faible de cette technique est que le disque ne doit subir aucune modification en écriture par l'OS une fois que la stéganographie a été effectuée car sinon l'OS pourrait écraser sans en avoir connaissance le fichier caché car les secteurs du disque abritant ce fichier sont considérés comme espace libre pour l'OS.
- Les fichiers cachés peuvent être facilement récupérés à l'aide de logiciels spécialisés. En plus, puisqu'ils sont non répertoriés, ils sont considérés comme inexistants et donc l'OS peut malencontreusement réécrire dessus.

Plan du chapitre 4 (2/2)

Cryptographie

4. La stéganographie
   - Stéganographie linguistique
   - Stéganographie technique
   - Propriétés des systèmes de stéganographie
   - Techniques de stéganographie
5. La stéganalyse
   - Types d'attaques de stéganalyse
   - Méthodes de stéganalyse
6. Le tatouage numérique

Plan du chapitre 4 (2/2)

Cryptographie

4. La stéganographie
   - Stéganographie linguistique
   - Stéganographie technique
   - Propriétés des systèmes de stéganographie
   - Techniques de stéganographie
5. La stéganalyse
   - Types d'attaques de stéganalyse
   - Méthodes de stéganalyse
6. Le tatouage numérique

- Contrairement à la stéganographie, la stéganalyse – appelée aussi analyse stéganographique – est l'art et la science de détecter si un medium donné cache un message secret, et si possible, de récupérer ce message caché.
- La stéganalyse est une tâche très difficile, vu la grande diversité des médiums, la grande variation des données, les différents algorithmes d'insertion et généralement la faible distorsion due à l'intégration du message.
- Toutefois, la stéganalyse est toujours possible car la présence d'un message caché rend un médium original et sa version stégo correspondante différents à certains aspects, bien que cette présence est souvent imperceptible à l'œil humain.

Plan du chapitre 4 (2/2)

Cryptographie

4 La stéganographie

- Stéganographie linguistique
- Stéganographie technique
- Propriétés des systèmes de stéganographie
- Techniques de stéganographie

5 La stéganalyse

- Types d'attaques de stéganalyse
- Méthodes de stéganalyse

6 Le tatouage numérique

- Il existe deux types d'attaques de stéganalyse :
  - ***Les attaques actives :*** l'attaquant (ou le stéganalyste) souhaite non seulement détecter le message caché mais, en plus, chercher à extraire, modifier ou supprimer ces données. Cette destruction aura souvent lieu par l'intermédiaire de modification du support de couverture (ex : par compression, changement de format, recadrage, symétrie,…).
  - ***Les attaques passives :*** l'attaquant ne cherche qu'à détecter la présence des messages dissimulés. Ce type d'attaques peut prendre plusieurs formes :
    - La lecture ou l'écoute de fichier ;
    - La comparaison avec le fichier original (s'il est disponible) ;
    - Certaines attaques statistiques (exple : attaque sur le LSB) ;
    - La détection des signatures des logiciels de stéganographie utilisés (exple : étude du code hexadécimal) ;

Selon les moyens dont dispose le stéganalyste, on distingue entre plusieurs formes d'attaques :

- ***Attaque avec stégo-médium seul (stego-only attack) :*** seul le stégo-médium est connu. L'insertion d'un message change certaines caractéristiques statistiques du cover-médium. L'attaque est basée sur cette altération.
- ***Attaque avec cover et stégo médium (known cover attack) :*** le médium de couverture et le stégo-médium sont disponibles. Ce type d'attaques est basé sur la comparaison entre le cover-médium et le stégo-médium (ex : attaque visiuelle).
- ***Attaque sur message connu (known message attack)*** : certaines parties du message caché sont connues de l'attaquant qui essayera de les retrouver dans le stégo-médium afin de faciliter l'analyse des documents futurs. Cette attaque est très difficile et généralement considérée comme équivalente à l'attaque stego-only.
- ***Attaque avec un algorithme choisi (chosen stego attack)*** : l'algorithme et le stégo-médium sont connus.
- ***Attaque avec un message choisi (chosen message attack)*** : le stéganalyste génère un stégo-médium à l'aide du message choisi. Le but est d'observer le résultat produit pour pouvoir cracker l'algorithme.
- ***Attaque avec un algorithme connu (known stego attack)*** : l'algorithme, le médium de couverture et le stégo-médium sont connues.

Plan du chapitre 4 (2/2)

Cryptographie

4 La stéganographie

- Stéganographie linguistique
- Stéganographie technique
- Propriétés des systèmes de stéganographie
- Techniques de stéganographie

5 La stéganalyse

- Types d'attaques de stéganalyse
- Méthodes de stéganalyse

6 Le tatouage numérique

- Les méthodes de stéganalyse peuvent être classées en deux catégories :
  - ***La stéganalyse spécifique*** : vise à casser un algorithme stéganographique spécifique ;
  - ***La stéganalyse universelle*** : tente à briser tous les algorithmes de stéganographie ;

- En général, les approches spécifiques ont une précision de détection plus élevée par rapport à celles universelles, car elles ont une connaissance préalable de la méthode d'insertion. Cependant, la stéganalyse universelle est plus attrayante dans l'utilisation pratique, car elle fonctionne indépendamment de la technique d'insertion et donc elle s'applique à des algorithmes d'insertion inconnus.

Les principales méthodes de stéganalyse sont :

- Stéganalyse visuelle (BMP, JPEG, GIF, …)
- Stéganalyse audible (WAV, MPEG, …)
- Stéganalyse statistique
- Stéganalyse structurelle
- Détection de signature d'un logiciel donné

Stéganalyse visuelle (1/2)

- L'insertion d'un message dans le dernier plan de bit (insertion LSB) peut se faire de façon aléatoire sur l'ensemble des pixels de l'image ou de façon séquentielle à partir de début de l'image.
- L'idée de l'analyse visuelle est basée sur le fait que pour une image peu texturée, le plan LSB est corrélé avec l'image d'origine. Ainsi, toute insertion de message perturbe le plan LSB. Cette perturbation est d'autant plus importante que la taille du message à dissimuler est grande.
- L'attaque consiste à appliquer des filtres sur l'image originale et l'image stéganographiée supprimant les composantes les plus visibles (bits de poids forts) et renforçant les autres (bits de poids faible).
- Pour une image originale, le dernier plan de bit montre une régularité qui correspond à la régularité de l'image avant toute insertion de message. Cependant, pour une image stéganographiée , le plan LSB est très bruitée et laisse apparaître de façon claire la présence d'un message dans l'image.

Stéganalyse visuelle (2/2)

Le dernier plan de bit de l'image originale (à gauche) et l'image stéganographiée (à droite)

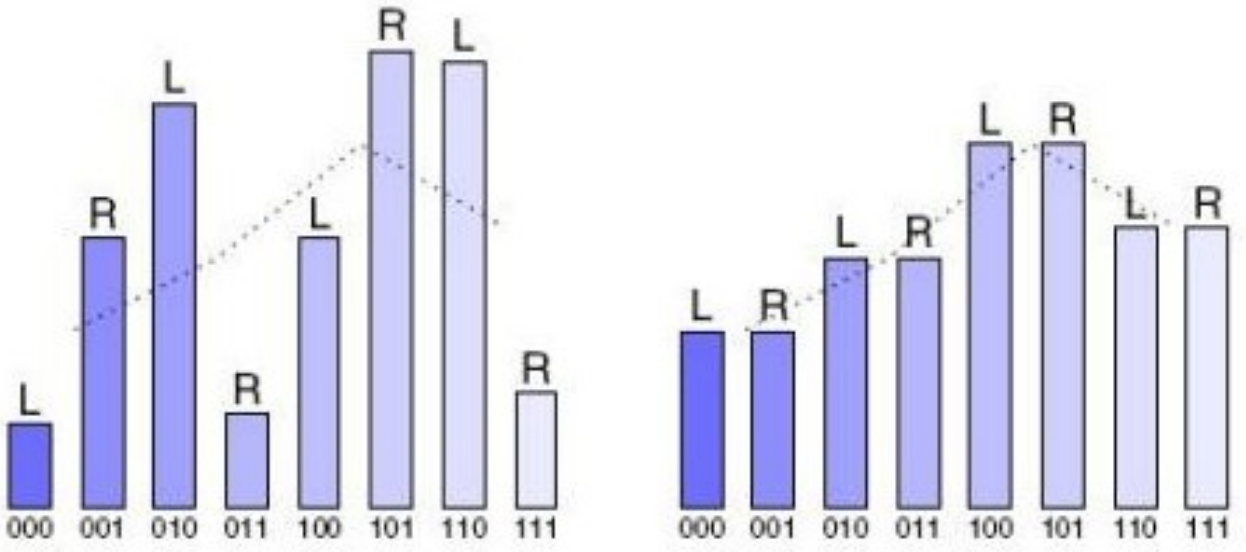
***N.B* : il est à noter que l'analyse visuelle n'est pas très efficace contre les méthodes d'insertion aléatoire ni sur les images très texturées.**

Stéganalyse statistique (1/2)

- Il existe différentes méthodes de stéganalyse basées sur l'analyse statistique.
- Le principe de ces méthodes se base sur le choix de sous ensembles de paires de pixels vérifiant des hypothèses proposées pour la stéganalyse (exple: égalité de sous-ensembles). La détection de la stéganographie se base sur le fait que l'insertion de données dans les bits de poids faibles peut modifier ou ne pas vérifier une des hypothèses proposées.
- **Exemple** : ***Analyse statistique à base de χ^2***

  La stéganalyse χ^2 a été proposée par Westfeld et Pfitzmann. Elle est basée sur le principe que toute surécriture des LSB dans les fichiers images en modifie les caractéristiques d'un point de vue statistique. En effet, étant donné que les bits du message à dissimuler sont uniformément distribués, les fréquences d'apparition des nuances d'une paire de valeurs de pixels deviennent quasiment identiques contrairement à une image non stéganographiée.

Stéganalyse statistique (2/2)



Fréquences d'apparition des nuances des paires de valeurs de pixels pour une image originale (à gauche) et une image stéganographiée (à droite)

Stéganalyse structurelle

Elle consiste en la comparaison des propriétés ou/et du contenu du support de couverture et du stégo-médium :

- Différence de taille ;
- Différence de la date de création ou de la dernière date de mise à jour ;
- Différence du contenu ;
- Checksum …

Détection de signature

- Certains programmes laissent derrière eux, de manière intentionnelle ou par erreur de conception, des artefacts permettant de caractériser leur passage.
- Cette façon de faire possède l'avantage de directement fournir le nom de l'application utilisée pour la dissimulation de données.
- Toutefois, chaque application de stéganographie doit faire l'objet d'une analyse spécifique.

Plan du chapitre 4 (2/2)

Cryptographie

4 La stéganographie
- Stéganographie linguistique
- Stéganographie technique
- Propriétés des systèmes de stéganographie
- Techniques de stéganographie

5 La stéganalyse
- Types d'attaques de stéganalyse
- Méthodes de stéganalyse

6 **Le tatouage numérique**

- Le tatouage numérique (ou watermarking) cherche à répondre au problème de la protection des droits d'auteur.
- Il s'agit bien de dissimulation d'information puisque, pour y parvenir, on insère un tatouage, appelé aussi marque ou filigrane, dans le médium spécifique au propriétaire. L'insertion doit minimiser les modifications subies par le médium afin d'être imperceptible.
- La dissimulation en tatouage ne signifie pas la même chose qu'en stéganographie : le stéganalyste ici sait qu'un tatouage est présent dans le stégo-médium, mais cette connaissance ne doit cependant pas lui permettre de le retirer.
- Contrairement à la stéganographie, l'information à dissimuler, en tatouage numérique ,est en rapport (direct ou indirect) avec le médium de couverture.

- Le tatouage numérique comprend deux types :
  - Tatouage fragile ;
  - Tatouage robuste ;

Tatouage numérique fragile

- Le tatouage numérique fragile n'est utilisé que pour prouver l'authenticité des documents et l'intégrité des données.
- La protection de la marque (ou tatoue) est très faible. Ainsi, toute modification du support de couverture affecte également le marquage.
- Ce type de tatouage pose un problème, car même s'il permet de prouver qu'un document a subi une transformation, il ne prouve pas pour autant qui est l'auteur du document.

Tatouage numérique robuste

- Comparé au tatouage numérique fragile, le tatouage robuste est plus dur à contourner et doit résister à diverses attaques.
- Généralement, le tatouage robuste doit vérifier les deux contraintes suivantes :
  - La marque doit être très résistante vis à vis des différentes attaques connues telles que : ré-échantillonnage, impression puis scanne, compression, coupure, bruits et changements de format.
  - la marque doit être facilement reconnaissable après extraction et ceci malgré le dommage subi à cause des différentes attaques. Ce type de tatouage est utilisé surtout dans les applications de protection de copyright et de contrôle de copies.