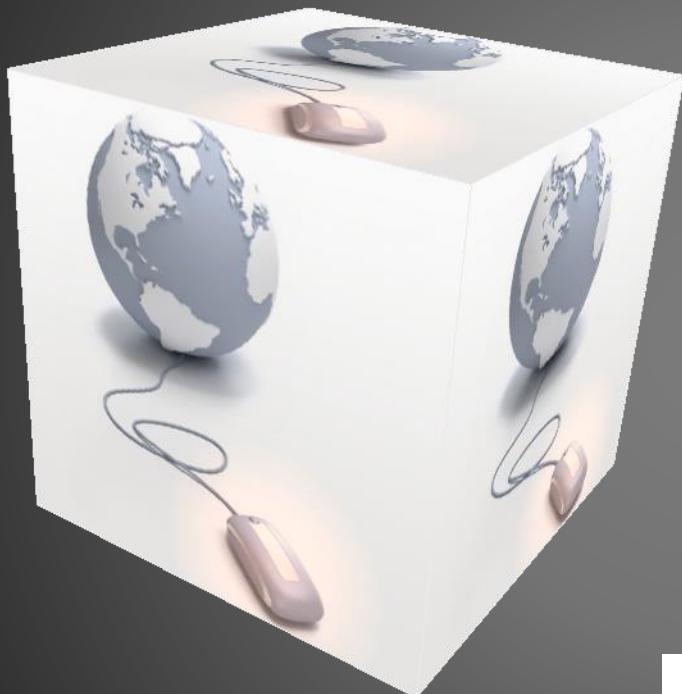


Txhotxhv sdv hq  
fubswrjudsklh...

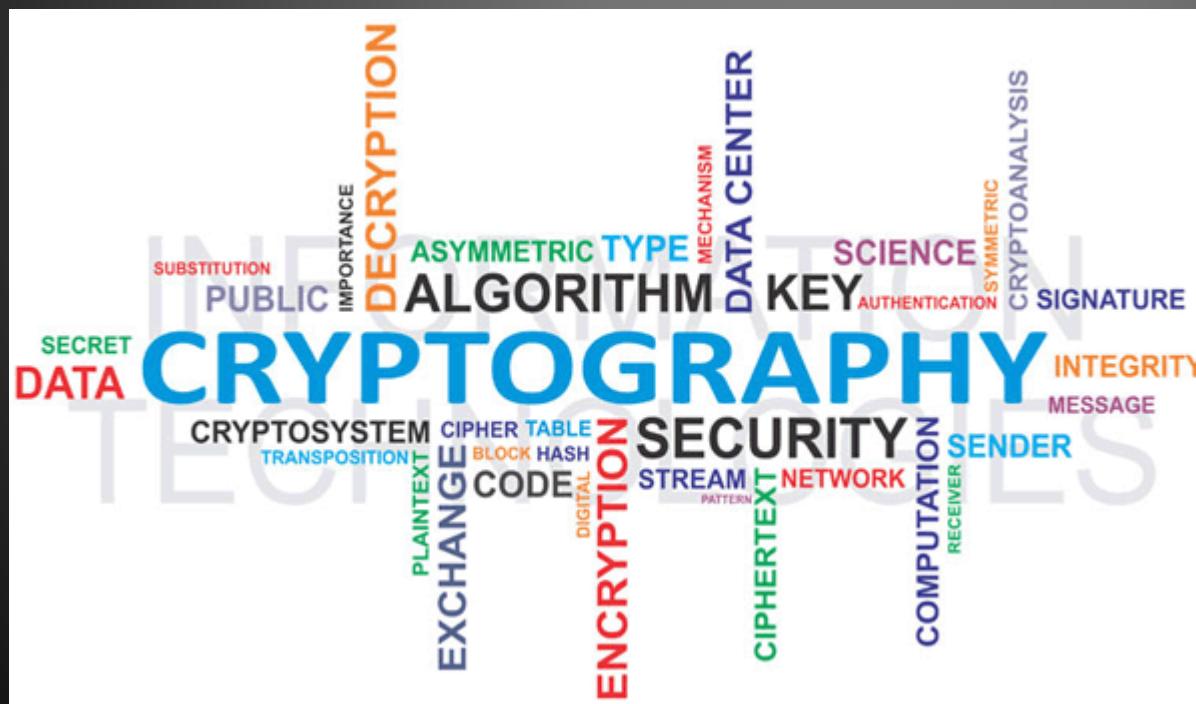
Quelques pas en  
cryptographie...



# Plan

- Un petit tour dans le monde de la cryptographie
- Algorithmes à clé secrète
- Algorithmes à clé publique
- Signatures numériques
- Certificats numériques
- Stéganographie
- Cryptographie quantique
- Et dans les langages
- Quelques pas en cryptanalyse

# Un petit tour dans le monde de la cryptographie



# Contenu

- Introduction
- Cryptologie
- Vocabulary
- Algorithmes
- Signature numérique
- Certificat numérique
- Stéganographie
- Cryptographie quantique

# Introduction

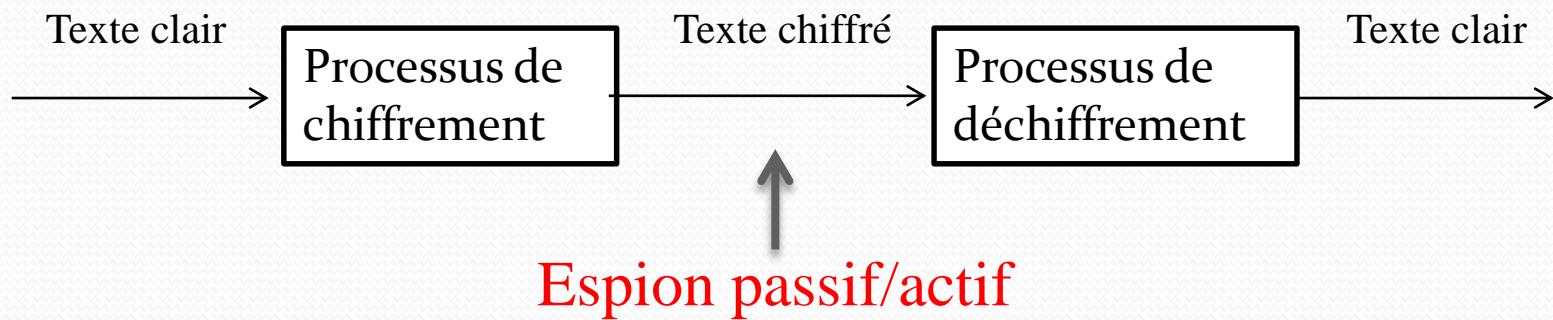
- **Problème de sécurité d'échanges d'informations depuis l'Antiquité**
- **Problèmes de sécurité sur les réseaux**
  - Indiscrétion (Eavesdropping)
    - Informations non altérées mais confidentialité compromise (espion passif)
    - Exemple : récupération du numéro de la carte de crédit et du code
  - Falsification (Tampering)
    - Informations modifiées avant d'arriver au destinataire (espion actif)
    - Exemple : changer le montant d'un virement bancaire
  - Imitation (Impersonation)
    - Mystification (Spoofing)
      - Une entité se fait passer pour une autre
      - Exemple : utilisation frauduleuse de l'adresse e-mail d'une personne

# Introduction

- **Problèmes de sécurité sur les réseaux**
  - Imitation
    - Imposture (Misrepresentation)
      - Une entité prétend être ce qu'elle n'est pas
      - Exemple : le site [www.escroc.be](http://www.escroc.be) prétend commercialiser des fournitures alors qu'il ne fait qu'encaisser les paiements par carte de crédit sans jamais livrer
- **Conséquence**
  - Cacher les informations cruciales en les chiffrant (en les cryptant)

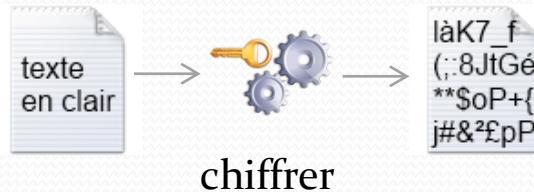
# Introduction

- Procédé de chiffrement/déchiffrement

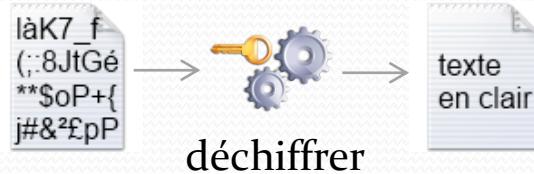


# Introduction

- Présence de clé(s) pour chiffrer/déchiffrer
  - Chiffrement : passage d'un texte en clair à un texte chiffré par du codage avec des clés



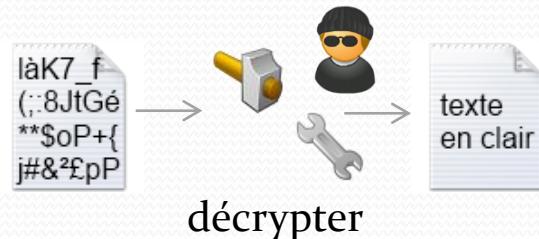
- Déchiffrement : repasser au texte en clair en ayant la ou les clés



# Introduction

## Décrypter

- À partir du texte chiffré, essayer d'obtenir le texte en clair sans nécessairement connaître la(es) clé(s)



- Système relativement sûr si l'espion ne parvient pas à décrypter en un temps record

# Cryptologie

- Cryptologie : science des secrets
  - Subdivisée en deux disciplines
  - Cryptographie et cryptanalyse
- Cryptographie ≡ dissimuler
  - Rendre de l'information incompréhensible pour celui qui n'a pas la(les) clé(s) de chiffrement
  - Stéganographie
    - Dissimuler de l'information dans un support (image, texte, ...) sans nécessairement la chiffrer

# Cryptologie

- Cryptanalyse ≡ briser le secret
  - Analyser un texte chiffré pour le décrypter
  - Déchiffrement
    - Opération inverse du chiffrement : obtenir le texte initial en connaissant la méthode de chiffrement et la(les) clé(s)
  - Décryptage
    - Restauration des données qui avaient été chiffrées à leur état premier ("en clair"), sans disposer de clé(s) théoriquement nécessaires

# Vocabulary

- **Some terms**

- *cryptography* → cryptographie
- *encryption* → chiffrement
- *decryption* → action de déchiffrer / décrypter
- *cipher* → algorithme de chiffrement
- *plaintext* → message dans sa forme originelle
- *ciphertext* → message chiffré

# Historique

## • Antiquité

### • Scytale

- Système grec à transposition permettant de changer l'ordre des lettres du message



→ épaisseur de la scytale



<https://www.youtube.com/watch?v=7uq4hIV0DkU>

### • Atbash

- Chiffrement par substitution simple employé par les Hébreux

Plain: אַבְגָּדְהַזְּחִתִּי כְּלֹמְנְסְנֶפְצָקָרְשָׁת  
Cipher: תְּשִׁרְקָצְפָּעֵסְנְמָלְכִּיתְזְׁוֹהַגְּבָא

<http://cryptography.wikia.com/wiki/Atbash>

- Exemple : iesn devient RVHM

Plain: abcdefghijklmnopqrstuvwxyz  
Cipher: zyxwvutsrqponmlkjihgfedcba



→ nombre de lettres

# Historique

## • Antiquité

### • Chiffre de César

- Tenant son nom de Jules César qui l'a utilisé, système romain
- Principe de substitution mono-alphabétique : remplacer une lettre par une autre
- Décalage de 3 par César

CLAIR	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
-> décalage = 3																										
CODE	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C

- Exemple : IESN devient LHVQ

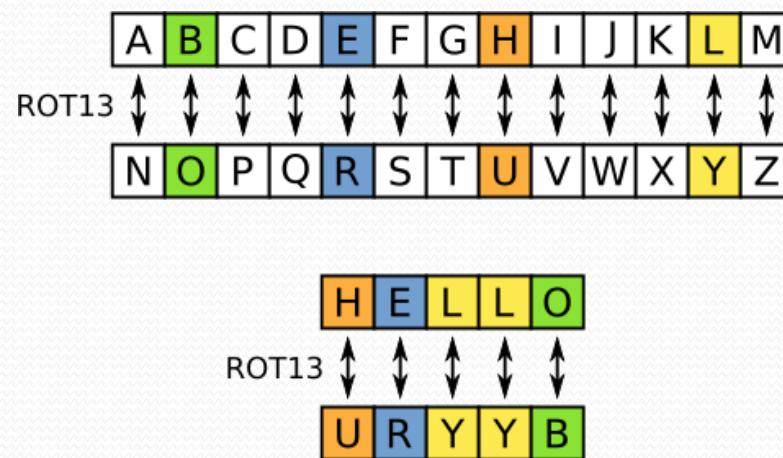


→ nombre de lettres de décalage = 3

# Historique

## • Antiquité

- Substitution mono-alphabétique plus générale
  - Décalage de n lettres
  - Exemple : n = 13



→ nombre de lettres de décalage

# Historique

- **Substitution mono-alphabétique encore plus évoluée**

- Clé

- Chaîne constituée de toutes les lettres de l'alphabet dans un autre ordre
- Exemple

Texte à chiffrer : iesn

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

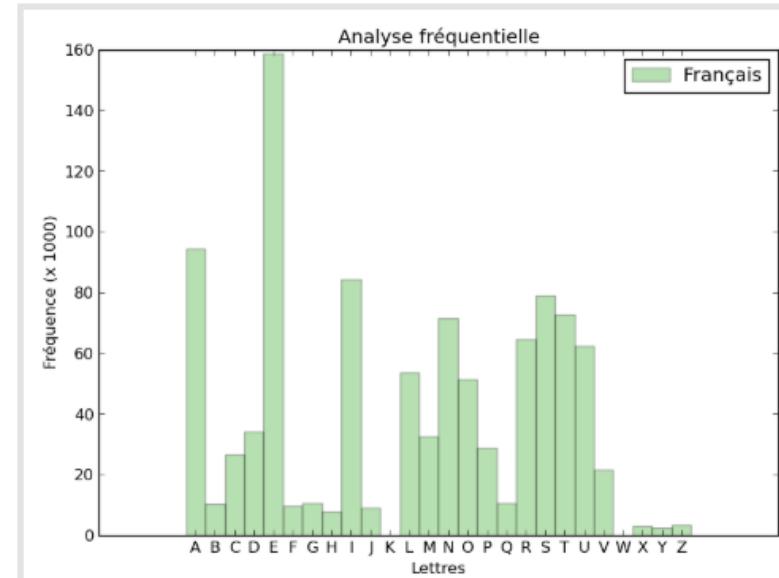
Texte chiffré : kxid

# Historique

## • Analyse

- Trop facilement cassable
  - Toute lettre est remplacée par la même
  - Selon la langue, fréquence de certaines lettres et de certains groupes de lettres

La répartition des lettres en français est donnée par l'histogramme suivant :



<http://blogs.univ-poitiers.fr/laurentsianc/2013/10/30/dechiffrer-automatiquement-le-chiffre-de-vigenere/>

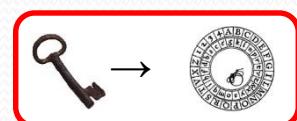
# Historique

- Moyen-Âge

- Cadran d’Alberti (XVe siècle)
    - Deux disques composés de l’alphabet
    - Grand disque : lettres dans l’ordre
    - Petit, dans le désordre
    - Dans un premier temps, les 2 « A » alignés
    - Toutes les 4 lettres du message à coder, on tourne le petit disque d’une lettre
    - Premier **procédé de chiffrement poly-alphabétique**
    - Pas très sécurisé car il suffit de posséder le cadran pour décrypter
    - Cf. <http://www.bibmath.net/crypto/index.php?ac>



<https://www.youtube.com/watch?v=eEWi4p5KYM8>



# Historique

## Renaissance

- Code de Vigenère (16<sup>ième</sup> siècle)
  - Pendant 3 siècles, incassable
  - Procédé
    - On prend un tableau

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

<http://www.bibmath.net/crypto/index.php?action=affiche&quoi=index.php?action=affiche&quoi=poly/vigenere>

# Historique

- Renaissance

- Code de Vigenère (16<sup>ième</sup> siècle)

- Procédé

- On écrit en-dessous du texte autant de fois que nécessaire une clé de longueur arbitraire
      - Texte : « CRYPTOGRAPHIE » ; clé : « MATHWEB »

C	R	Y	P	T	O	G	R	A	P	H	I	E
M	A	T	H	W	E	B	M	A	T	H	W	E

- On cherche (C,M) dans la table : C en ligne, M en colonne ; on obtient O (première lettre ainsi chiffrée)
      - Etc...
      - On obtient ainsi ORRWPSHDAIOEI

# Historique

- **XIX<sup>e</sup> siècle**
  - Le principe de Kerckoffs (1883)

*Il faut qu'il (le système cryptographique) n'exige pas de secret, et qu'il puisse sans inconvénient tomber entre les mains de l'ennemi.*

*La sécurité d'un système de chiffrement ne doit pas dépendre de la préservation du secret de l'algorithme.*

*La sécurité ne repose que sur le secret de la clé.*

*[Extrait du Journal des Sciences Militaires]*

- Si deux correspondants conservent le secret de la clé, un tiers interceptant le message qu'ils échangent ne pourra pas le déchiffrer

# Historique

- **XX<sup>e</sup> siècle**

- Le chiffre de Vernam (masque jetable)
  - Algorithme inventé en 1917 par G. Vernam
  - La clé doit répondre aux trois critères :
    - aussi longue que le texte à chiffrer
    - parfaitement aléatoire;
    - utilisée pour chiffrer un seul message, puis détruite de suite
  - Inconvénients :
    - Taille des clés très élevée
    - Échange des clés sécurisé et donc difficile à réaliser !
    - Clés utilisées parfaitement aléatoires -> pas facile à garantir

# Historique

- **XX<sup>e</sup> siècle**
  - Le chiffre de Vernam
    - Utilisé par Che Guevara et Castro à Cuba lors de la révolution cubaine
    - Utilisé pour sécuriser le téléphone rouge du temps de la guerre froide
      - Les clés circulaient dans les valises diplomatiques, transportées dans des avions bourrés d'agents secrets
      - Pour produire des clés aléatoires, les Soviétiques employaient des "lanceurs de dés" : leur travail consistait à lancer des dés toute la journée et à noter le résultat

# Historique

- **XX<sup>e</sup> siècle**
  - Enigma (à partir de 1918)
    - Machine électronique portable permettant le chiffrement et déchiffrement
    - Fort utilisée lors de la Seconde Guerre Mondiale par les Allemands
    - Une fois les travaux polonais poursuivis par les Britanniques (Alan Turing), les Alliés ont pu déchiffrer les messages allemands et gagner 2 ans de guerre

# Historique

- XX<sup>e</sup> siècle
  - Enigma (1919)
    - Basé sur le chiffre de Vigenère de longueur 26<sup>nbre</sup> de rotors.



# Algorithmes

## • Introduction

- Principe de Kerchoffs : la ou les clés sont les plus importantes
- Conséquence : les algorithmes sont publiés pour que les chercheurs essaient de les « craquer »; s'ils n'y parviennent pas en un temps record, l'algorithme est fiable (à condition que certaines clés restent secrètes)
- NIST : National Institute of Standards and Technology
  - Sécurité dans la ou les clé(s), longueur minimale conseillée
- Plusieurs catégories
  - À clé secrète
  - À clé publique
  - Hybride

# Algorithmes

## • Algorithmes à clé secrète

- Clé de déchiffrement calculée à partir de la clé de chiffrement
- Algorithmes symétriques : clés identiques



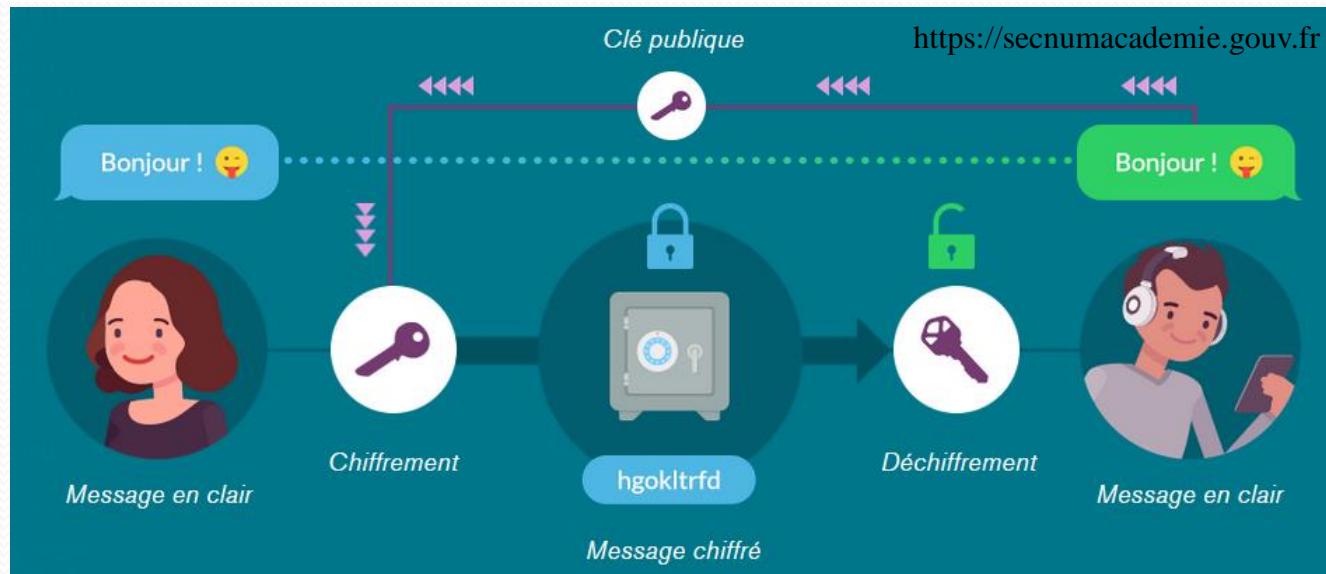
<https://secnumacademie.gouv.fr>

- 2 catégories
  - Algorithmes de chiffrement en continu (bit par bit)
  - Algorithmes de chiffrement par blocs ou groupes de bits
- Partage de la clé entre 2 interlocuteurs
  - Sécurité de la clé ?
  - Taille de la clé pour ne pas être devinée rapidement!

# Algorithmes

## • Algorithmes à clé publique

- Préambule
  - Bob envoie son cadenas sans clé en position ouverte.
  - Alice met la lettre dans la valise et la ferme à l'aide du cadenas.
  - Alice envoie la valise fermée à Bob. L'espion n'ayant pas la clé, il ne sait pas l'ouvrir.
  - Bob reçoit la valise et ouvre le cadenas à l'aide de sa clé.



# Algorithmes

- **Algorithmes à clé publique**

- Clés différentes, non calculables l'une à partir de l'autre
- Clé de chiffrement publique ( annuaires)
- Notion de fonction
  - À sens unique
    - Facile à calculer, impossible à inverser en un temps raisonnable avec une puissance de calcul raisonnable
    - Exemple :  $\text{nbre1} * \text{nbr2} = \text{produit}$ ; si l'espion voit produit, il faut beaucoup de temps pour retrouver nbr1 et nbr2 (de l'ordre de 100 chiffres décimaux)
  - Avec trappe
    - Information supplémentaire facilitant le calcul de l'inverse

# Algorithmes

- **Clés**

- Taille de la clé primordiale
  - Clé codée sur  $n$  bits :  $2^n$  valeurs possibles



<https://www.silicon.fr/cybersecurite-un-pas-de-geant-franchi-dans-la-cryptographie-quantique-191661.html>

- Plus la clé est longue,
  - plus le nombre de possibilités est grand
  - plus le temps pour décrypter le message sera long
- *DES (1977)*
  - Clé chiffrement/déchiffrement de 56 bits ( $7,20576E+16$  possibilités)
  - En 1997, une recherche exhaustive de clés a été réussie
    - Algorithme : plus fiable
- Secret clé(s) : comment la(les) protéger? qui peut y accéder?

# Signature numérique

- Objectifs de la cryptographie (et de la sécurité)

## CONFIDENTIALITÉ

Le message doit être incompréhensible pour les autres personnes



## INTÉGRITÉ

Le message transmis n'a pas été falsifié ou détruit



## NON

## RÉPUDIATION

On ne peut nier que le message a été envoyé, reçu, ...



## AUTHENTIFICATION

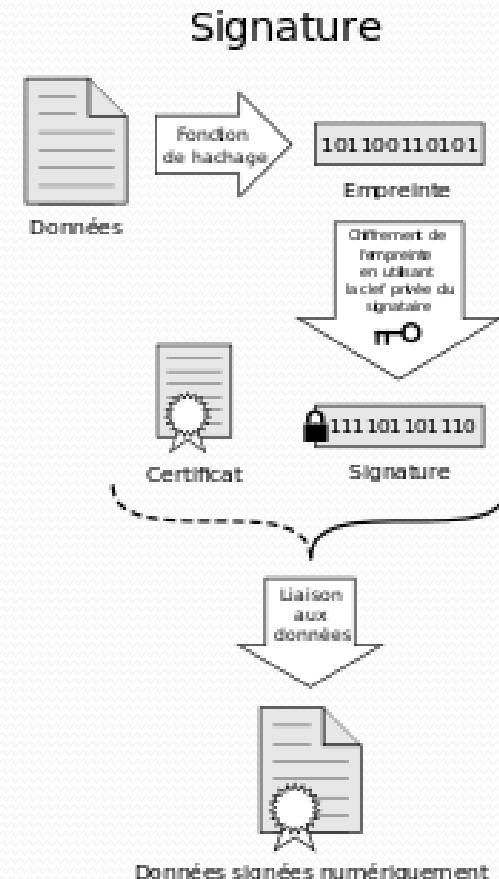
On est certain de l'identité des personnes en contact



# Signature numérique

## • Ajout d'une signature numérique

- Technique de validation mathématique de l'authenticité et de l'intégrité d'un message, d'un logiciel ou d'un document électronique
- Construction
- Propriétés :
  - Aucune contrefaçon
  - Message modifié → signature non valide



# Certificat numérique

- **But**

- Garantir l'authenticité de la signature par un document
- Contenu
  - Informations sur l'entité dont on authentifie la signature, la signature de l'entité et l'autorité de certification qui émet le certificat

# Stéganographie

- Cacher texte/image dans texte/image
- Travail sur les bits de poids les plus faibles des couleurs de base (rouge, vert, bleu) associées à chaque pixel



# Cryptographie quantique

- Voie de l'avenir
- Transfert de photons
  - Création d'une clé vraiment secrète « en direct » (« au moment du besoin »)



- A l'aide de satellites

# Algorithmes à clé secrète

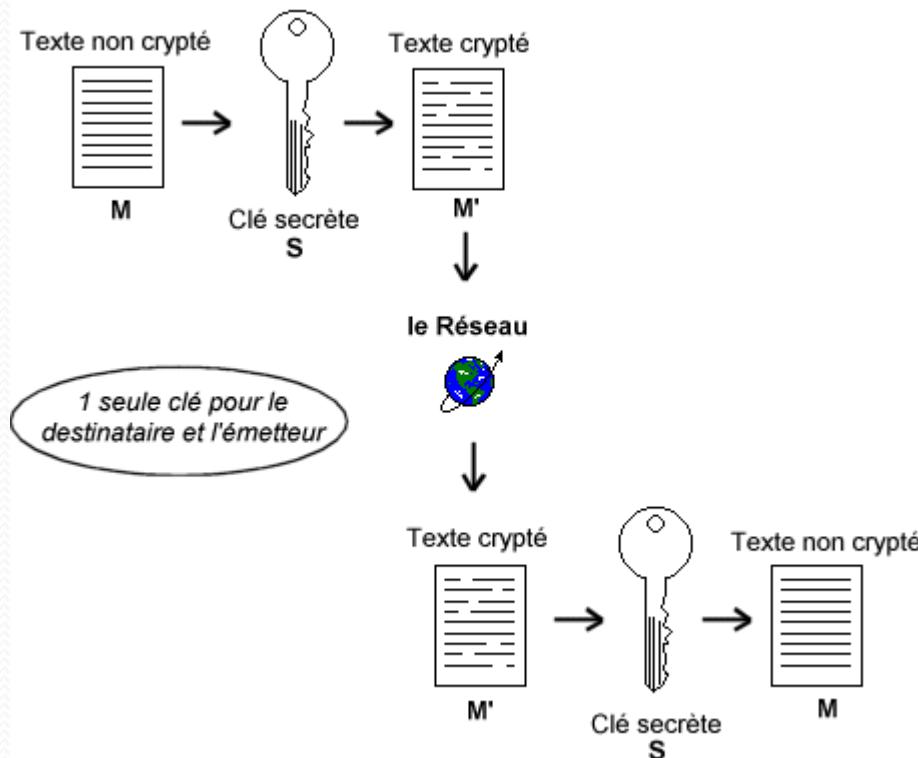


# Contenu

- Introduction
- Clés
- Algorithmes
- Algorithme DES
- Algorithme TDES
- Algorithme AES

# Introduction

- Rappel



[http://www.linux-france.org/prj/edu/archinet/systeme/index\\_monopage.html](http://www.linux-france.org/prj/edu/archinet/systeme/index_monopage.html)

# Clés

## Distribution des clés

- Chaque couple d'interlocuteurs doit posséder sa propre clé secrète.
- Si trois personnes veulent échanger des messages entre eux, il faudra 3 clés uniques pour maintenir les échanges confidentiels.
- Si une quatrième personne veut dialoguer, on doit ajouter 3 clés.
- Il faut donc disposer d'autant de clés qu'il y a de paire de correspondants.
- N personnes nécessitent  $N*(N-1)/2$  clés.
- 1000 utilisateurs doivent échanger 499 500 clés qui doivent rester secrètes.



<https://secnumacademie.gouv.fr>

# Clés

- **Comment générer cette clé de manière aléatoire ?**
  - Soit utiliser des paramètres tels que l'heure et la date de l'ordinateur à un moment donné
  - Soit utiliser des formules de nombres pseudo-aléatoires
  - Soit prendre des mesures d'appareils physiques tels que la mesure de radiations, d'un flux de liquide épais dans de l'eau
  - Mieux : prendre un algorithme!

# Clés

- Espion
  - Un seul message, déchiffrer sans clé ?
  - Deux messages avec la même clé
    - Par analyse des fréquences des deux messages cryptés, en fonction de la propriété suivante
$$C_1 \oplus C_2 = M_1 \oplus M_2,$$
fréquences dans les messages en clair
- Algorithme à chiffre à usage unique (inconditionnellement sûr)
  - Longueur clé = longueur texte

# Algorithmes

- **Caractéristiques**

- Plus complexes que les méthodes mono-alphabétiques ou poly-alphabétiques
- Taille de clé plus petite que le texte
- Plus les technologies sont améliorées, plus rapide est le temps pour retrouver la clé et ainsi déchiffrer le message...

- **Algorithmes**

- DES : abandon : seul, tout type d'attaque le met ko
- TDES
- AES

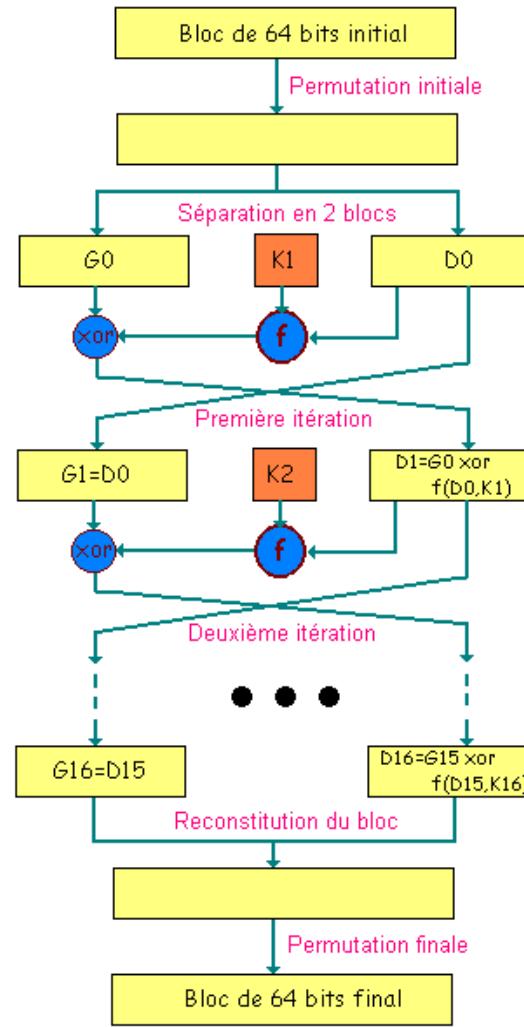
# Algorithme DES

- Obsolète sauf si appliqué 3 fois TDES
- Data Encryption Standard
  - Crée par IBM, publié en 1977 suite à un appel de NBS (NIST)
  - Norme ANSI (1991) : DEA
  - Norme ISO : DEA\_1
  - Abandonné vers la fin des années 1990
- Algorithme symétrique
  - Texte en clair découpé en blocs de 64 bits
  - Clé de 64 bits mais 56 seront utilisés en cours d'algorithme après transformation de la clé de départ

# Algorithme DES

## Principe

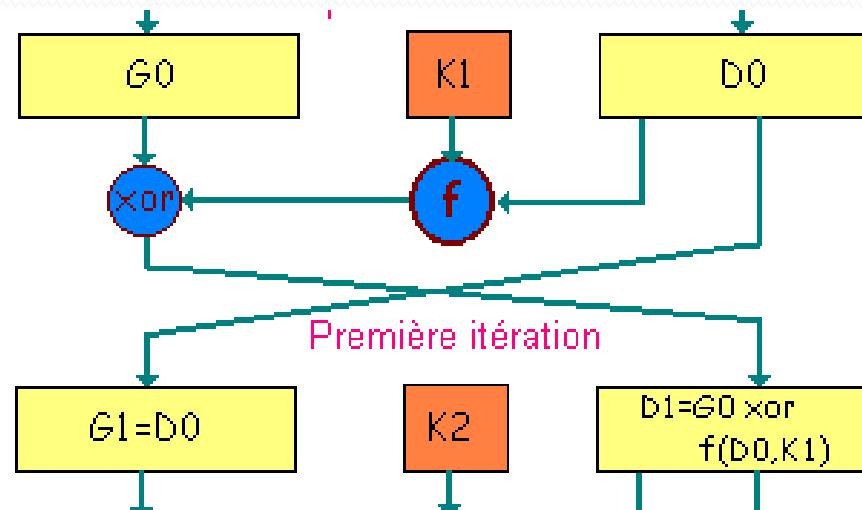
- Itération = ronde
- Chaque bloc sera transformé par 16 rondes



# Algorithme DES

## • Itération

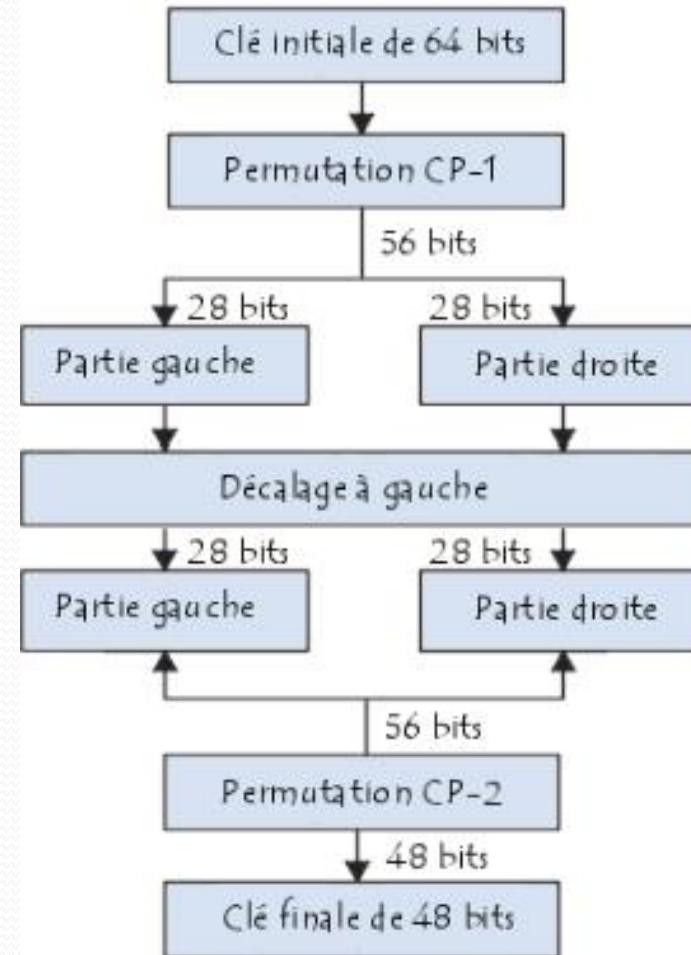
- Chaque sous-bloc de droite de 32 bits prend la place du sous-bloc de gauche
- Chaque sous-bloc de gauche de 32 bits est transformé par une formule : Gauche  $\oplus$  fonction(droite, sous-clé)



# Algorithme DES

## • Itération

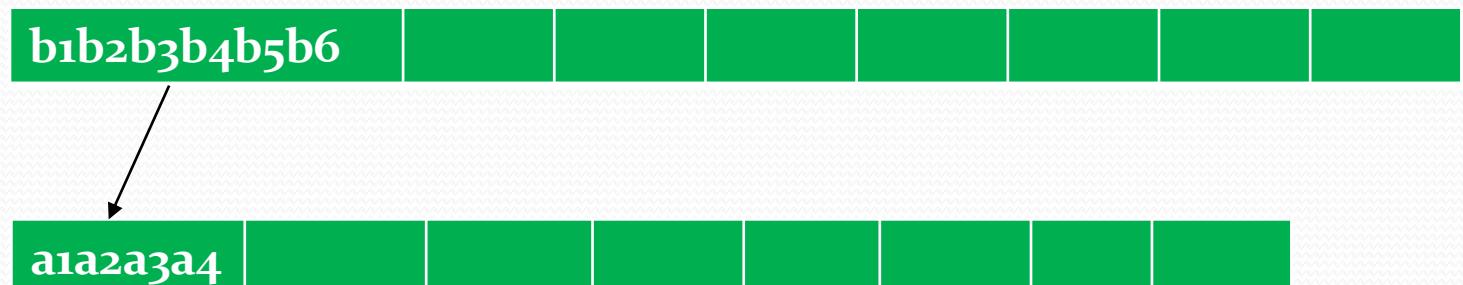
- Sous-clé  $K_i$ ?
- Les décalages varient selon la ronde



# Algorithme DES

## • Itération

- Qu'est-ce la fonction(droite, Key<sub>i</sub>) où i = numéro ronde?
  - Droite : 32 bits
  - Droite subit une permutation expansive (certains bits sont dupliqués – 48 bits)
  - Droite transformée  $\oplus$  Key<sub>i</sub>
  - Ensemble de 48 bits : 8 groupes de 6 bits -> 8 groupes de 4 bits



# Algorithme DES

- **Itération**

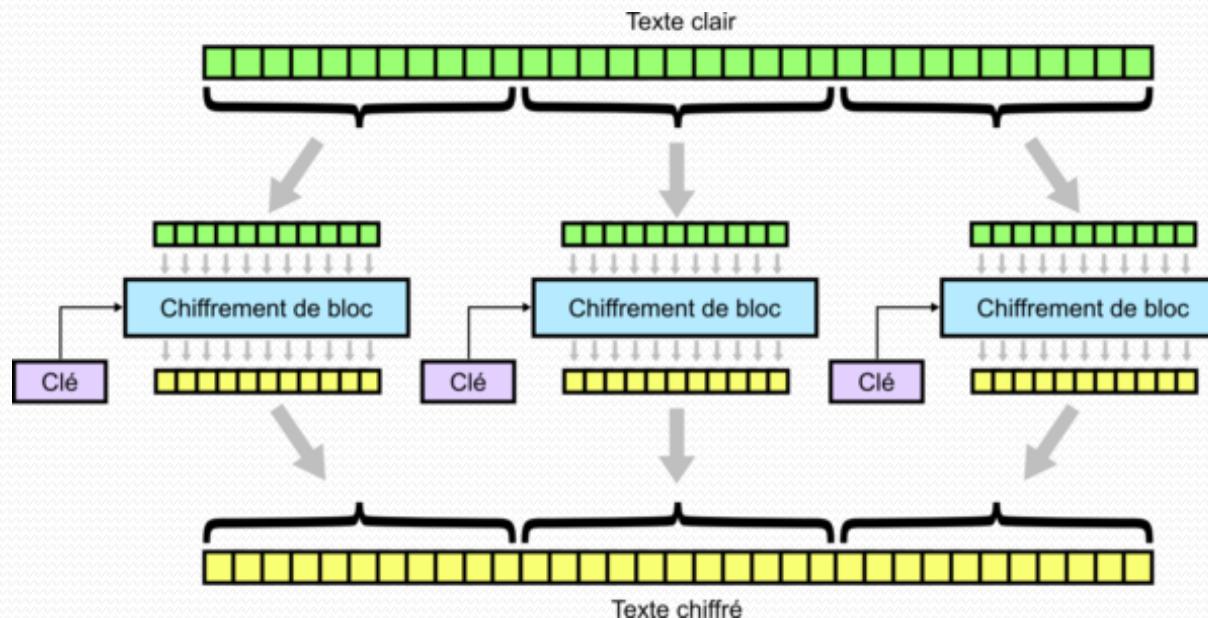
- Qu'est-ce la fonction(droite, Key<sub>i</sub>) où i = numéro ronde?
  - b<sub>1</sub>b<sub>2</sub>b<sub>3</sub>b<sub>4</sub>b<sub>5</sub>b<sub>6</sub> transformé via des tableaux notés S
    - b<sub>1</sub>b<sub>6</sub> donne la ligne,
    - b<sub>2</sub>b<sub>3</sub>b<sub>4</sub>b<sub>5</sub> donne la colonne
    - → valeur en 4 bits reprise dans le tableau
  - Droite de 8 blocs de 4 bits (32 bits) subit une permutation
  - Ainsi est terminée la fonction dont le résultat subira un xor avec la gauche pour former la nouvelle droite

- **Attaqué via les tableaux S et la taille de la clé trop petite**

# Algorithme DES

## Modes

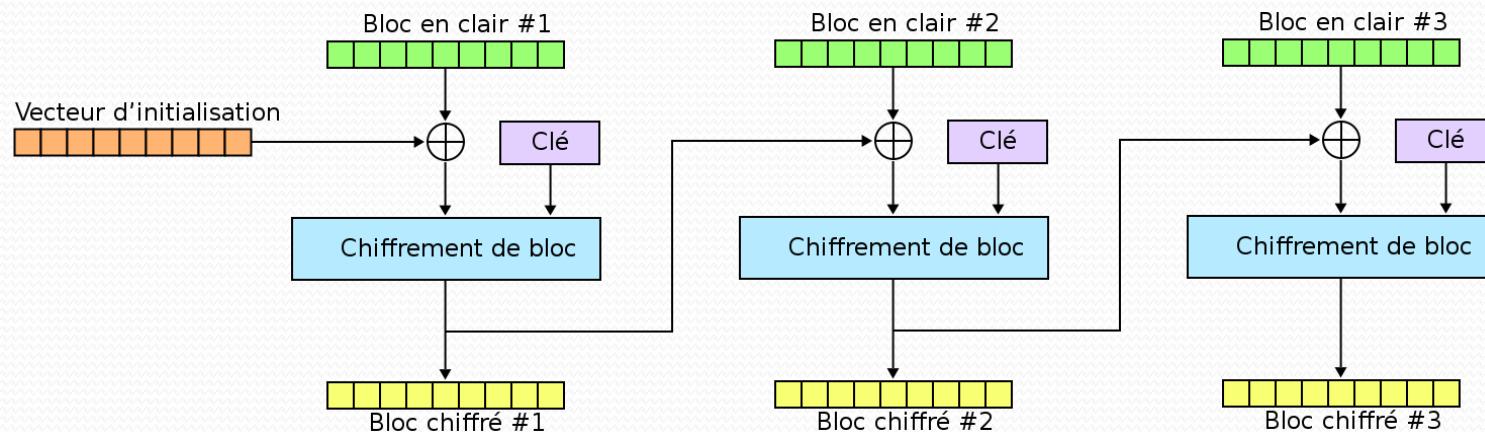
- ECB (Electronic Code Book)
  - Découpe en blocs cryptés les uns indépendamment des autres
  - Désavantages :
    - présence de blocs redondants
    - un bloc crypté avec la même clé donnera le même bloc chiffré



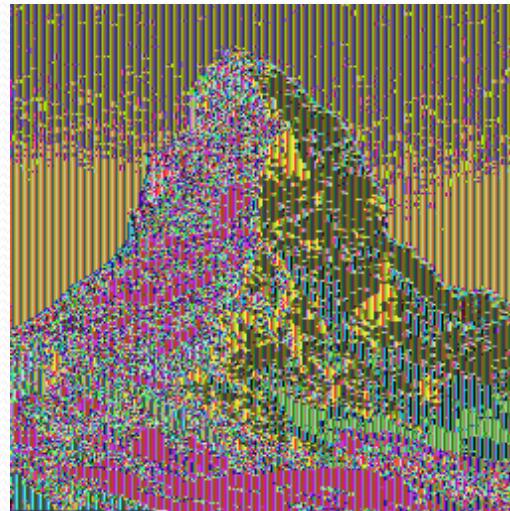
# Algorithme DES

- Modes

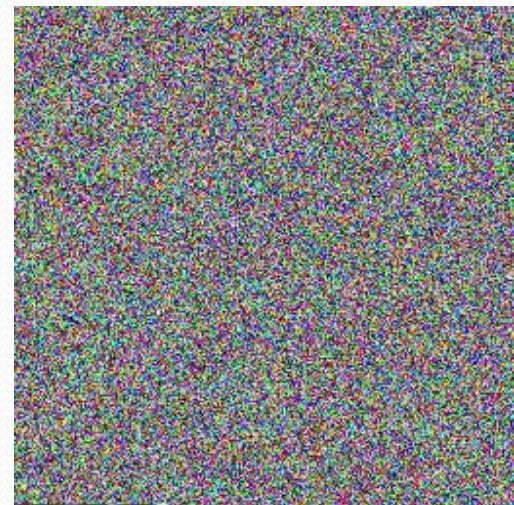
- CBC (Chain Block Cipher)
  - bloc clair  $\oplus$  bloc précédent crypté



# Algorithme DES



Mode ECB



Mode CBC

Et d'autres modes existent...

# Algorithme TDES

- Fonctionnement
  - 3 étapes DES successives
    - 3 clés identiques
    - clé<sub>1</sub>, clé<sub>2</sub>, clé<sub>1</sub>
    - 3 clés différentes (NIST)
- Modes d'opération
  - EDE
    - Chiffrement avec clé<sub>1</sub>, déchiffrement avec clé<sub>2</sub>, chiffrement avec clé<sub>1</sub>
  - EEE
    - Chiffrement avec clé<sub>1</sub>, chiffrement avec clé<sub>2</sub>, chiffrement avec clé<sub>3</sub>
  - Et d'autres...

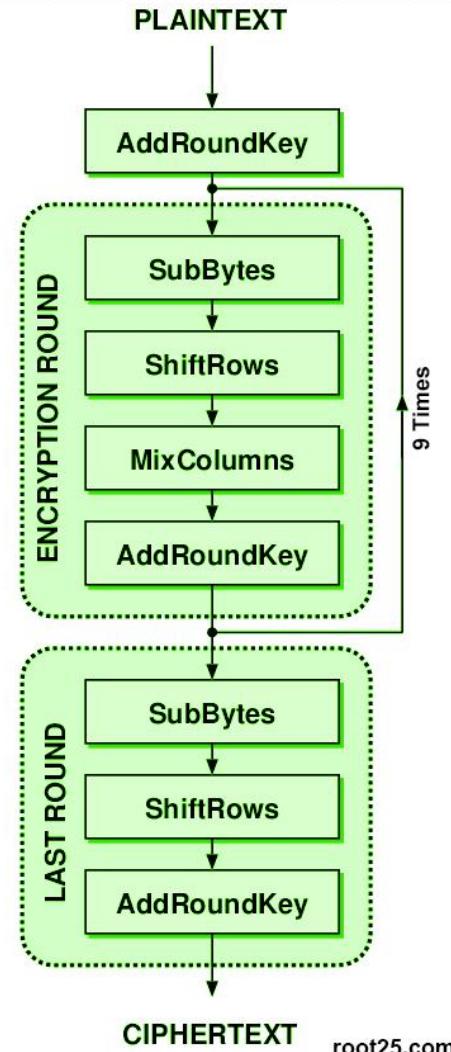
# Algorithme AES

- Advanced Encryption Standard
  - 1998 : algorithme Rijndaël écrit par J. Daemen et V. Rijmen
  - Adopté par NIST en 2001 sous forme AES
- Découpe
  - Différentes combinaisons [longueur de clé]-[longueur de bloc] : 128-128, 192-128 et 256-128 bits (Rijndael supporte également des tailles de blocs variables, mais cela n'est pas retenu dans le standard)
- Principe
  - Découpe du texte en clair en blocs chiffrés par une ronde de 4 transformations
  - Ronde répétée 10, 12 ou 14 fois selon la longueur de la clé
  - Blocs chiffrés placés les uns à la suite des autres

# Algorithme AES

- Transformation d'un bloc

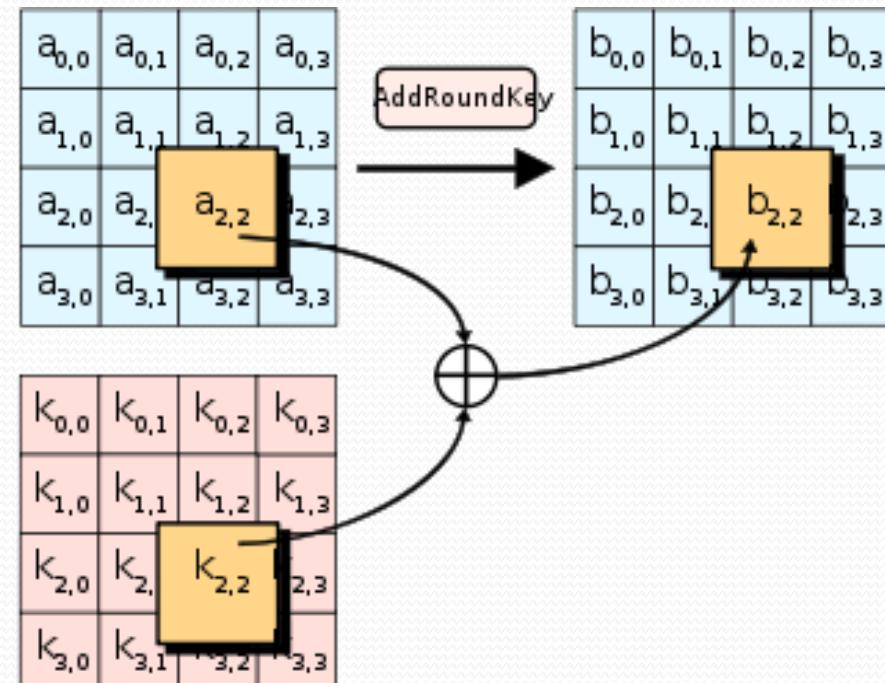
- 3 parties
  - AddRoundKey
  - 9 rondes
  - Dernière ronde



# Algorithme AES

## • AddRoundKey

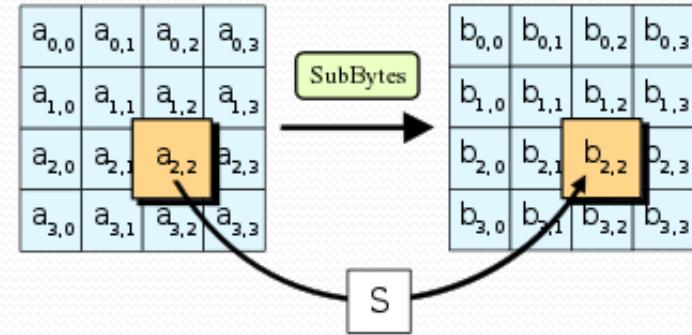
- Dans l'algorithme, tout ensemble de 128 bits est considéré comme une matrice  $4 \times 4$  (octets)
- Travail dans cette étape avec un clé de ronde construite à partir de la clé initiale



# Algorithme AES

## • SubBytes

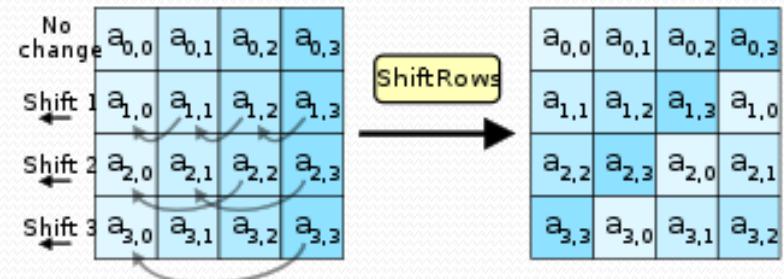
- Chaque élément est changé via des tables notées S (élément = 8 bits, 4 bits gauche donnent ligne, 4 de droite la colonne)



[https://en.wikipedia.org/wiki/Advanced\\_Encryption\\_Standard](https://en.wikipedia.org/wiki/Advanced_Encryption_Standard)

## • ShiftRows

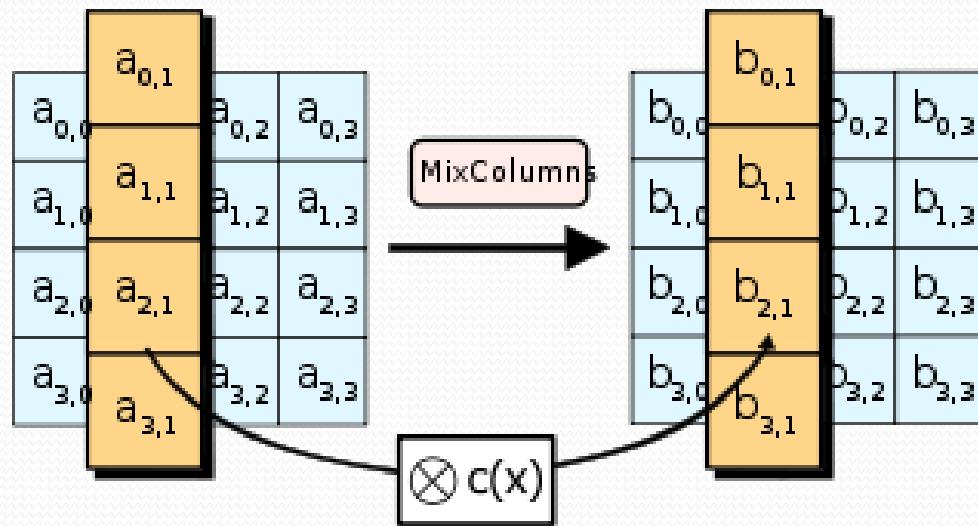
- Shifts circulaires sur chaque ligne exceptée la première



[https://en.wikipedia.org/wiki/Advanced\\_Encryption\\_Standard](https://en.wikipedia.org/wiki/Advanced_Encryption_Standard)

# Algorithme AES

- MixColumns

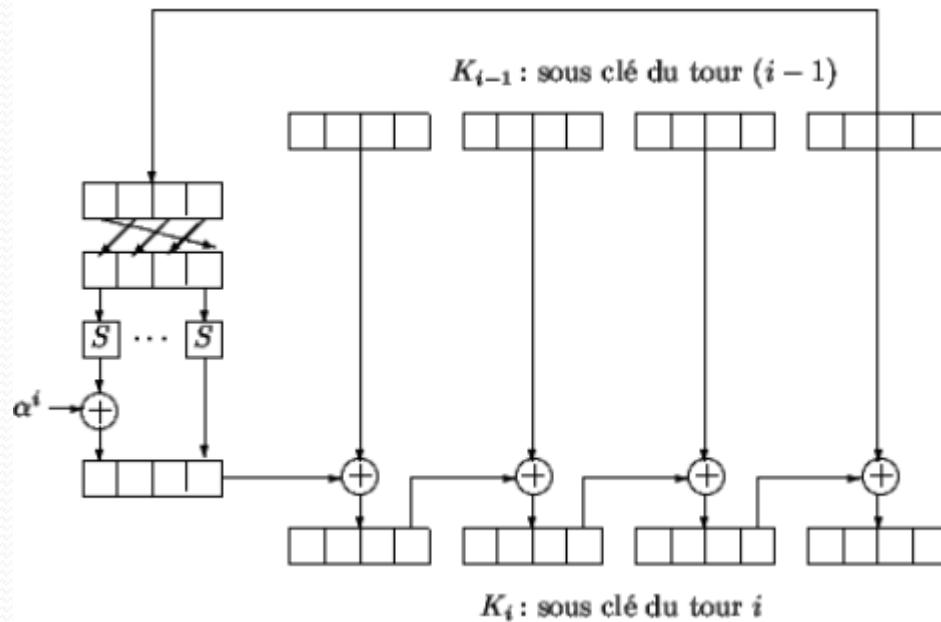


$$\begin{bmatrix} b_{0,j} \\ b_{1,j} \\ b_{2,j} \\ b_{3,j} \end{bmatrix} = \begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix} \begin{bmatrix} a_{0,j} \\ a_{1,j} \\ a_{2,j} \\ a_{3,j} \end{bmatrix} \quad 0 \leq j \leq 3$$

[https://en.wikipedia.org/wiki/Advanced\\_Encryption\\_Standard](https://en.wikipedia.org/wiki/Advanced_Encryption_Standard)

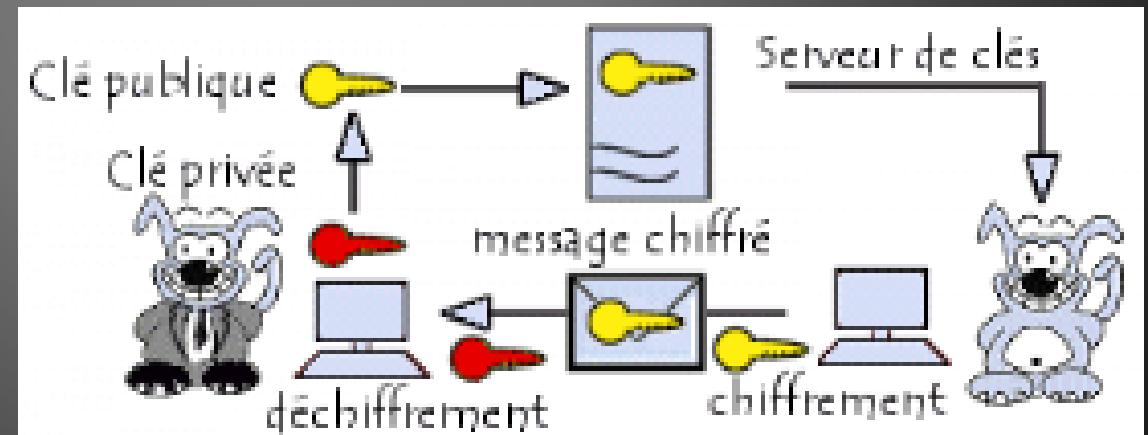
# Algorithme AES

- Clé de ronde ?



[http://www.ibiblio.org/pub/Linux/docs/LDP/linuxfocus/Francais/Archives/lf-2002\\_05-0243.html](http://www.ibiblio.org/pub/Linux/docs/LDP/linuxfocus/Francais/Archives/lf-2002_05-0243.html)

# Algorithmes à clé publique



# Algorithmes à clé publique

- Introduction
- Algorithme RSA
- Comparaison
- Systèmes hybrides
- [Algorithme fondé avec les courbes elliptiques]  
*Elliptic Curve Digital Signature Algorithm* (ECDSA)  
1992 - NIST

# Introduction

- **A clé secrète**

- Premiers trop vite cassables
- Secret de la clé...

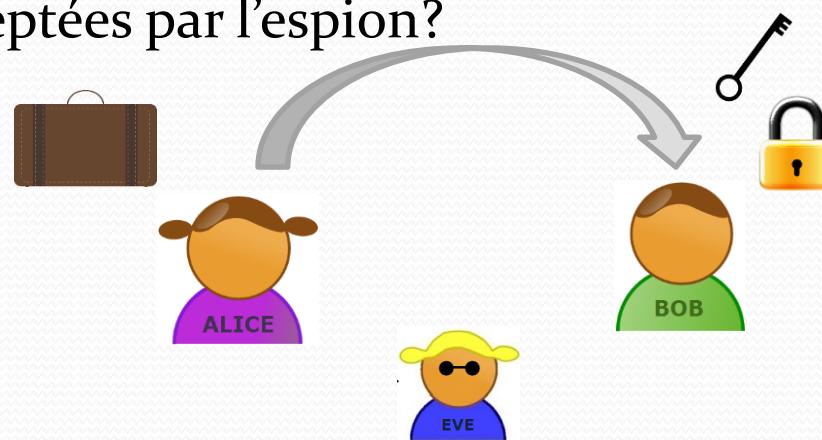
- **Réflexion quant à d'autres stratégies**

- Deux clés différentes
  - Clé de chiffrement  $\neq$  clé de déchiffrement
- Déchiffrement n'est pas l'inverse du chiffrement

# Introduction

## • Préambule

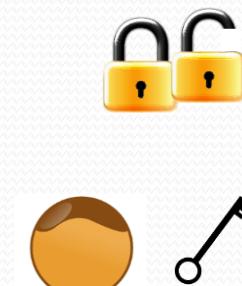
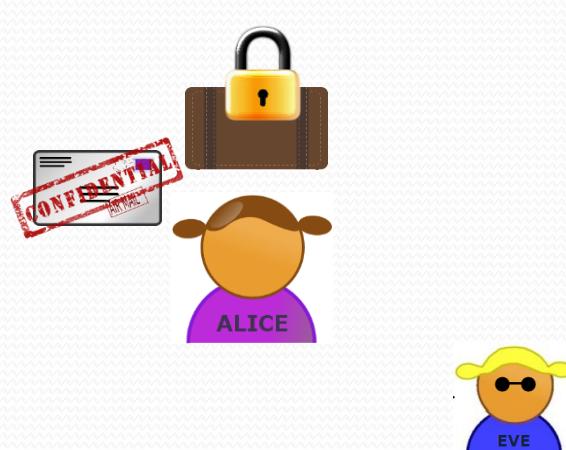
- Alice veut envoyer une lettre confidentielle à Bob
- Le facteur (espion) veut savoir ce qu'il est écrit dans la lettre
- Alice possède une valise bien solide
- Bob a en sa possession une clé et son cadenas
- Comment Alice et Bob peuvent procéder pour être certain que les informations contenues dans la lettre ne seront pas interceptées par l'espion?



# Introduction

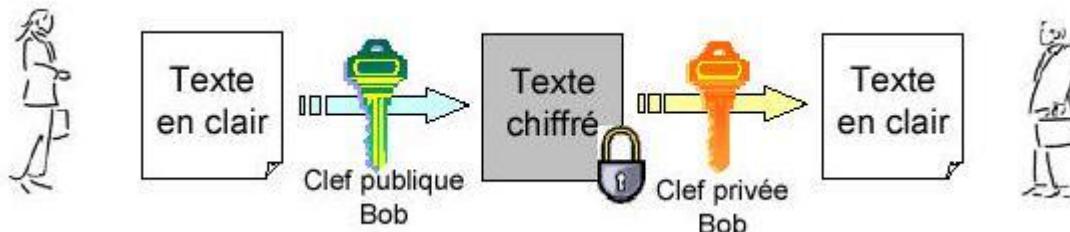
## • Préambule

- Bob envoie son cadenas sans clé en position ouverte
- Alice met la lettre dans la valise et la ferme à l'aide du cadenas
- Alice envoie la valise fermée à Bob ; l'espion n'ayant pas la clé, il ne sait pas l'ouvrir
- Bob reçoit la valise et ouvre le cadenas à l'aide de sa clé



# Algorithmes à clé publique

- **Clés**



- **Sécurité calculatoire**

- Durée du processus de calcul nécessaire à casser la clé doit être supérieure à la durée de vie de l'information à protéger

- **Gestion des clés**

- N personnes : N clés publiques (moins de clés à gérer pour l'ensemble, chacun gérant les clés privées)

- **Vitesse de calcul**

- Moins rapide que les algorithmes à clé privée

# Algorithme RSA

- Rivest – Shamir – Adleman – 1977 – MIT
- Basé sur la
  - Factorisation de grands nombres
  - Fonction à sens unique avec trappe : fonction puissance
- Taille des clés, conseillée en 2018
  - 2048 bits (617 chiffres décimaux)

# Algorithme RSA

## ● Protocole

- Création de clés
  - Bob crée aléatoirement 2 nbres premiers p et q >>>
  - Il calcule  $n = p * q$  et  $\Phi(n) = (p-1)*(q-1)$
  - Il engendre e, premier avec  $\Phi$   
(algorithme d'Euclide étendu)
  - Il calcule d, inverse de e au sens modulo de  $\Phi$ 
    - tel que  $d = e^{-1} \% \Phi$ ,  $e*d = 1 \% (p-1)(q-1)$
- On peut montrer que
  - n, d sont premiers entre eux
  - d est unique
- Bob rend publique (n,e) et garde d privée
  - p et q devenus inutiles sont écartés

# Algorithme RSA

## • Chiffrement

- Alice veut envoyer X à Bob :
  - X est transformé en un entier  $< n$  ou en plusieurs blocs numériques  $< n$
  - Données binaires
    - Taille d'un bloc = la plus grande puissance de 2  $< n$
  - Elle chiffre X ou un bloc numérique avec
    - $(e, n)$ , clé publique de Bob
    - Et la fonction puissance :

$$X^e \% n \ (= Y)$$

- Elle envoie Y à Bob

# Algorithme RSA

- **Déchiffrement**
  - Bob reçoit et déchiffre
    - Il prend sa clé privée  $d$  et  $n$
    - Il utilise la formule :  $Y^d \% n$
    - Il retrouve ainsi  $X$ !
- **Simplicité de l'algorithme mais coûteux en ressources**
  - Exemple : pour une carte à puce :
    - Une itération AES ou DES en 1 à 5 millisecondes (voir nanosecondes avec un accélérateur matériel)
    - Un déchiffrement RSA (considérant les optimisations matérielles et logicielles) en 100-300 millisecondes

# Algorithme RSA

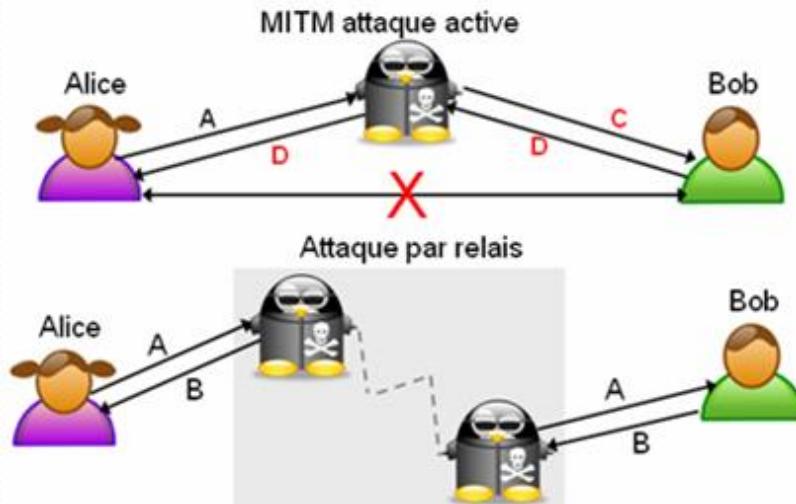
- Exemple

- Choix de  $p = 5$  et  $q = 11$ ;  $n$  vaut donc 55
  - $\Phi = 40$ ; choix de  $e$  premier avec  $40 : 7$
  - On calcule  $d = 23$  inverse de  $e = 7 \% 40$

Chiffrement						
Clair		Clé publique : 7		Chiffré		
		$P^7$	$P^7 \text{ mod } 55$			
I	9	4782969	4	70368744177664	9	I
U	21	1801088541	21	2576580875108218291929075869661	21	U
P	16	268435456	36	623673825204293256669089197883129856	16	P
M	13	62748517	7	27368747340080916343	13	M
I	9	4782969	4	70368744177664	9	I
A	1	1	1	1	1	A
G	7	823543	28	1925904380037276068854119113162752	7	G
E	5	78125	25	142108547152020037174224853515625	5	E
					$C^{23}$	$C^{23} \text{ mod } 55$
				Chiffré	Clé privée : 23	Déchiffré
				Déchiffrement		

# Man in the Middle Attack (MITM)

- Vulnérabilité des algorithmes à clé publique



[https://commons.wikimedia.org/wiki/File:MITM\\_Relay\\_Attack.png](https://commons.wikimedia.org/wiki/File:MITM_Relay_Attack.png)

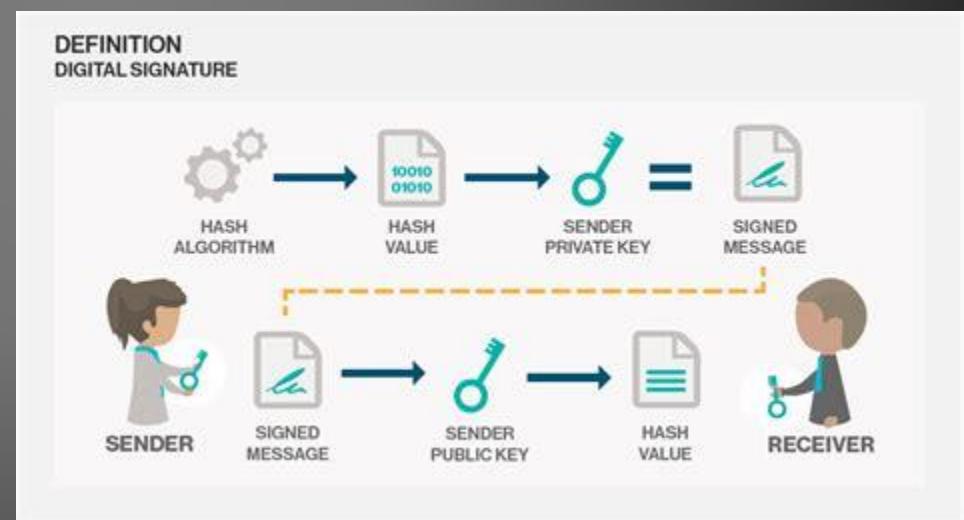
# Man in the Middle Attack (MITM)

- **Vulnérabilité des algorithmes à clé publique**
  - L'espion peut modifier les échanges entre Alice et Bob !
  - Bob envoie sa clé publique à Alice. L'espion l'intercepte et renvoie à Alice sa propre clé publique en se faisant passer pour Bob
  - Alice veut envoyer un message à Bob, elle utilise donc la clé publique de l'espion!
  - Alice chiffre le message avec la mauvaise clé et l'envoie à celui qu'elle croit être Bob
  - L'espion intercepte le message, le déchiffre avec sa clé privée et peut lire le message
  - Puis elle chiffre à nouveau le message avec la clé publique de Bob, après l'avoir éventuellement modifié
  - Bob déchiffre son message avec sa clé privée, et ne se doute de rien !

# Comparaison

- **Gestion du nombre de clés**
  - Algo à clé secrète : plus de clés à gérer
  - Algo à clé publique : moins de clés
- **Sécurité**
  - Défaut : sécurité du partage de la clé privée en algos symétriques
- **Temps d'exécution**
  - Programmation logicielle
    - Vitesse d'exécution des algos à clé secrète plus rapide
  - Implémentation matérielle
    - Souvent des algorithmes à clé secrète

# Signature numérique



# Contenu

- Introduction
- Fonctions de hachage
- Principes
- Algorithme MD5  
*Encore utilisé lors de téléchargements de fichiers.*
- Algorithmes SHA\_256 ou SHA\_512
- SHA\_3 (384 bits) : Keccak

# Introduction

- **Apports**

- Le destinataire d'un message peut vérifier l'identité affichée par l'émetteur
- L'émetteur ne pourra renier la paternité du contenu du message envoyé et signé
- Le destinataire ne peut créer lui-même un message et faire croire qu'il a été émis par un tiers

- **Droit**

- Signature juridiquement acceptable
  - Cryptage avec la clé privée de l'émetteur

# Introduction

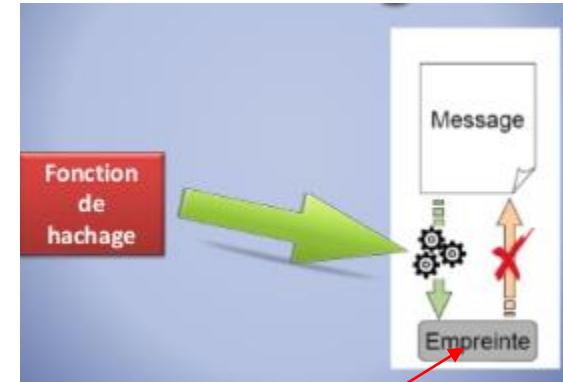
- Concepts

- Signature

- Cryptage d'une forme condensée
    - Avantage de la condensation
      - Temps d'exécution diminué
    - Condenser ne suffit pas!
      - Espion pourrait modifier document et donc, forme condensée
      - Donc, cryptage!

- Forme condensée ou empreinte

- Taille fixe quelle que soit la méthode employée pour condenser (ou hacher) *sauf pour les algorithmes de nouvelle génération*
    - Impossible d'inverser en un temps record



# Introduction

- Qualités de la fonction de hachage
  - Grande dispersion
    - Petit écart entre deux documents → empreintes très distinctes
  - Absence de collision
    - Deux documents ne peuvent donner la même empreinte
  - Inversion “impossible”

```
MD5("Wikipedia, l'encyclopédie libre et gratuite") =  
d6aa97d33d459ea3670056e737c99a3d
```

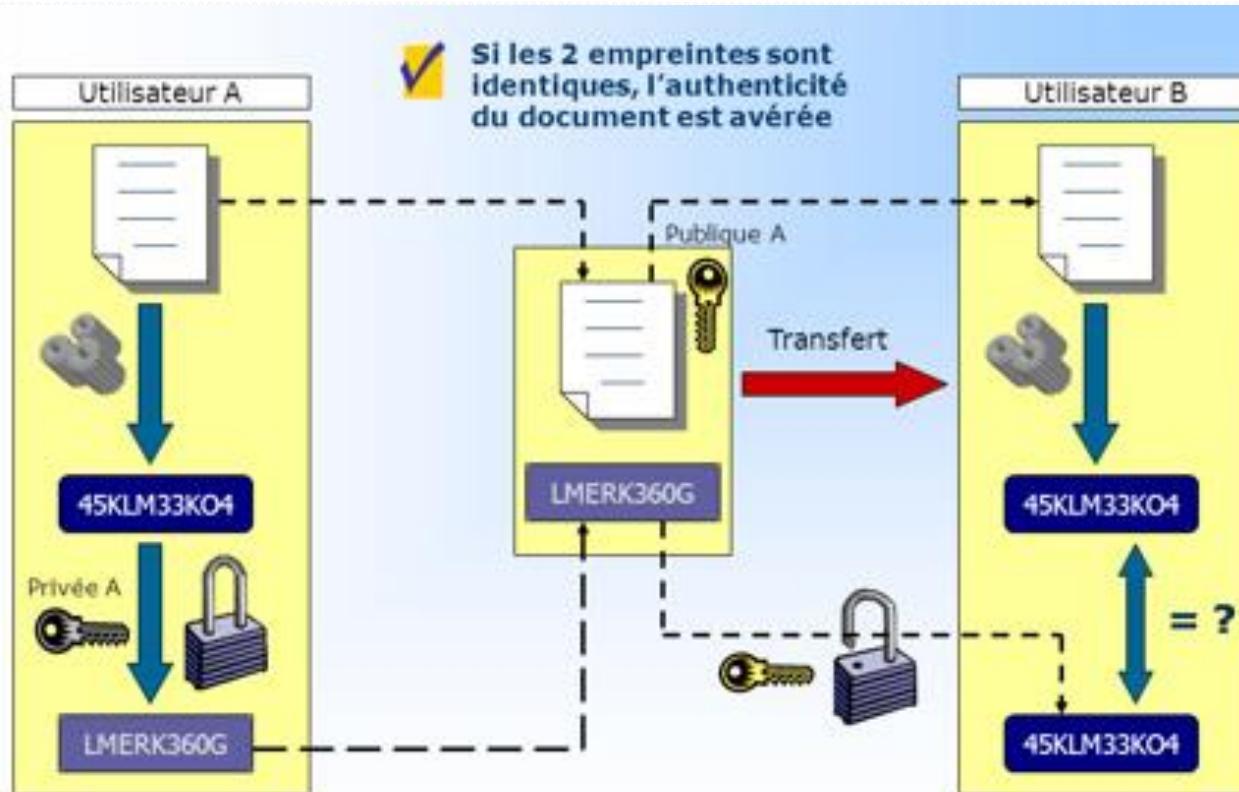
En modifiant un caractère, cette empreinte change radicalement :

```
MD5("Wikipedia, l'encyclopédie libre et gratuitE") =  
5da8aa7126701c9840f99f8e9fa54976
```

<https://www.slideshare.net/mohammededd/les-signatures-numriques>

# Introduction

- Mécanisme



# Fonctions de hachage

- Fonctions de hachage
  - MD5 (Message Digest version 5)
    - En 1991 par Rivest
    - Empreinte : 128 bits
    - Texte en clair : blocs de 512 bits
    - Déconseillé!
  - SHA\_1 (Secure Hash Algorithm)
    - Empreinte : 160 bits
    - Blocs de 512 bits
  - SHA\_2 : 256, 384, 512 bits comme empreinte, recommandé par NIST
  - RIPEMD\_160 (Ripe Message Digest)
    - Empreinte : 160 bits mais ressources plus grandes
- En pratique
  - Longueur  $\geq$  160 bits

# Principes

- Schéma pour MD5 / SHA

message	100.....0	Longueur du message % $2^{64}$
---------	-----------	--------------------------------

- Ajout d'un padding composé de 1 suivi de 0
- Longueur du message sur 64 bits
- Ensemble multiple de 512 bits
- Découpe de l'ensemble en blocs de 512 bits chacun
- Ces blocs vont subir des transformations au cours de rondes pour, en final, obtenir une empreinte de
  - 128 bits pour MD5
  - 256 bits pour SHA\_256
  - 512 bits pour SHA\_512

# MD5

- Apparu en 1991
- Collisions
  - Se sont présentées en 2004
- Principe

```
*  
A = 1re partie empreinte = 0X01234567  
B = 2de partie empreinte = 0X89ABCDE  
C = 3e partie empreinte = 0XFEDCBA98  
D = 4e partie empreinte = 0X76543210  
do par bloc de 512 bits  
do 4 times //ronde  
do 16 times (nombre de sous-blocs de 32 bits dans 512 bits)  
    [opération]  
    [ ]  
    [ ]  
1re partie empreinte += A  
2de partie empreinte += B  
3e partie empreinte += C  
4e partie empreinte += D
```

# MD5

- **Opération?**

- On prend 3 constantes parmi A, B, C et D
- On y applique une fonction différente suivant la ronde, ce qui donne un résultat res
- $\text{res} = \text{la } 4^{\text{e}} \text{ constante non choisie} \oplus \text{res}$
- $\text{res} = \text{res} \oplus \text{sous-bloc de 32 bits composant le bloc traité}$
- $\text{res} = \text{res} \oplus \text{pow}(2,32) * \text{abs}(\sin(\text{numéro ronde}))$
- $\text{res} = \text{res} \ll \text{variable}$
- $\text{res} = \text{res} \oplus \text{n'importe quelle constante parmi A, B, C ou D}$
- res remplace la constante non choisie
- **Empreinte finale constituée des 4 parties dont il est question : 128 bits**

# SHA

- **SHA\_1**
  - Publiée en 1995
  - Attaquable en 2005
  - Depuis 2010, Google et Microsoft commencent à l'interdire
- **SHA\_2**
  - Couvre SHA\_256 et SHA\_512 (et SHA\_384 et SHA\_224)
- **SHA\_3**
  - Crée en 2015
  - Empreinte variable!

# SHA\_256

- Mécanisme pour un bloc
  - Chaque bloc va être modifié par une fonction de compression
  - Fonction de compression
    - Deux entrées
      - Bloc à compresser : 512 bits
      - Résultat du bloc précédent passé par la fonction de compression : 256 bits
    - Une sortie
      - Bloc transformé : 256 bits
    - 64 rondes
    - 8 buffers au lieu de 4

# Certificat numérique



# Contenu

- Objectif
- Structure
- Validation
- Infrastructure à clés publiques
- Synthèse

# Introduction

- **Objectif**
  - Assurer que la clé publique appartient à la personne adéquate
- **Infrastructures**
  - Associer une entité et sa clé publique
  - Désignées par :
    - IGC « Infrastructure de Gestion de Clés »
    - PKI « Public Key Infrastructure »
  - Fonctions principales d'une PKI
    - Génération de couple unique de clés (privée et publique)
    - Crédit et gestion de certificats numériques
    - Diffusion des clés publiques aux ressources qui la solliciteraient
    - Certification des clés publiques

# Introduction

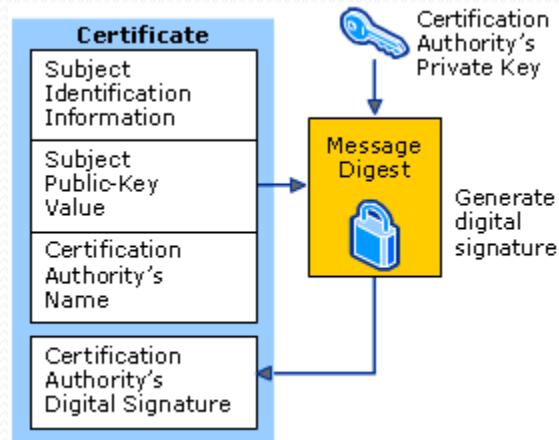
## ● But

- Attester de l'identité numérique du détenteur de la clé publique
- Délivré pour un organisme habilité, le certificat doit être :
  - **infalsifiable** : une signature y sera apposée, cryptée pour empêcher toute modification
  - **nominatif** : il est délivré à une entité (comme la carte d'identité est délivrée à une personne et une seule)
  - **certifié** : présence d'un « tampon » de l'autorité qui l'a délivré

# Introduction

## • Contenu

- Informations d'identité, notamment :
  - Coordonnées de l'entité dont on atteste la validité de la clé
  - Dates de début et de fin de validité
  - Nom de l'autorité de certification (CA : Certificate Authority)
- Signature chiffrée de l'autorité de certification
  - Pour permettre de vérifier que le certificat est bien délivré par l'autorité de certification



[https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2003/cc776447\(v=ws.10\)](https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2003/cc776447(v=ws.10))

# Introduction

- **Principaux types de certificat**

- Certificat de messagerie : crypter et signer ses e-mails
- Authentification I(nternet)P(rotocol)Sec(urity) pour un accès distant par V(irtual) P(rivate) N(etwork)
- Authentification Internet pour les pages Web sécurisées
- Cryptage des données avec E(ncrypting) F(ile) S(ystem)
- Signature de logiciel

# Structure

- **Informations**

- Version
  - Format du certificat
- Numéro de série
  - Identifier le certificat parmi tous les certificats générés par une même autorité de certification
- Algorithme de signature
  - Schéma de cryptographie asymétrique qui garantit l'intégrité et l'authenticité d'un message
    - Certificat : document signé par l'autorité de certification et indiquant à ce titre quel algorithme de signature a été utilisé pour créer cette signature
- Émetteur
  - Entité émettrice du certificat (autorité de certification)

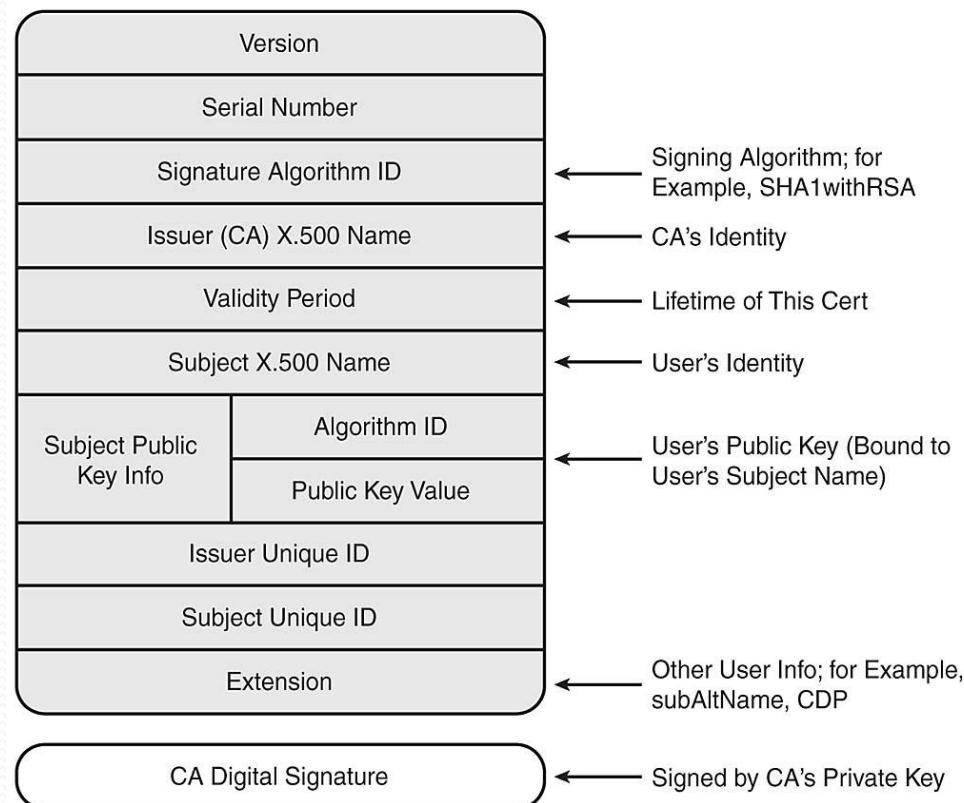
# Structure

- **Informations**

- Durée de validité du certificat
- Objet
  - Informations personnelles pour identifier l'entité détentrice du certificat sans risque de la confondre avec une autre
- Clé publique
  - Clé que partage la plateforme détentrice du certificat avec ses interlocuteurs pour échanger en toute sécurité avec eux

# Structure

- Exemple



# Validation

- **Réception du certificat**

- Validation : le client doit
  - obtenir la clé publique de l'organisme de certification
  - déchiffrer la signature (dernier champ du certificat) à l'aide cette clé publique
  - calculer le hachage du certificat
  - comparer l'empreinte calculée et celui se trouvant dans la signature
  - vérifier que la période de validité du certificat est correcte

- **MITM**

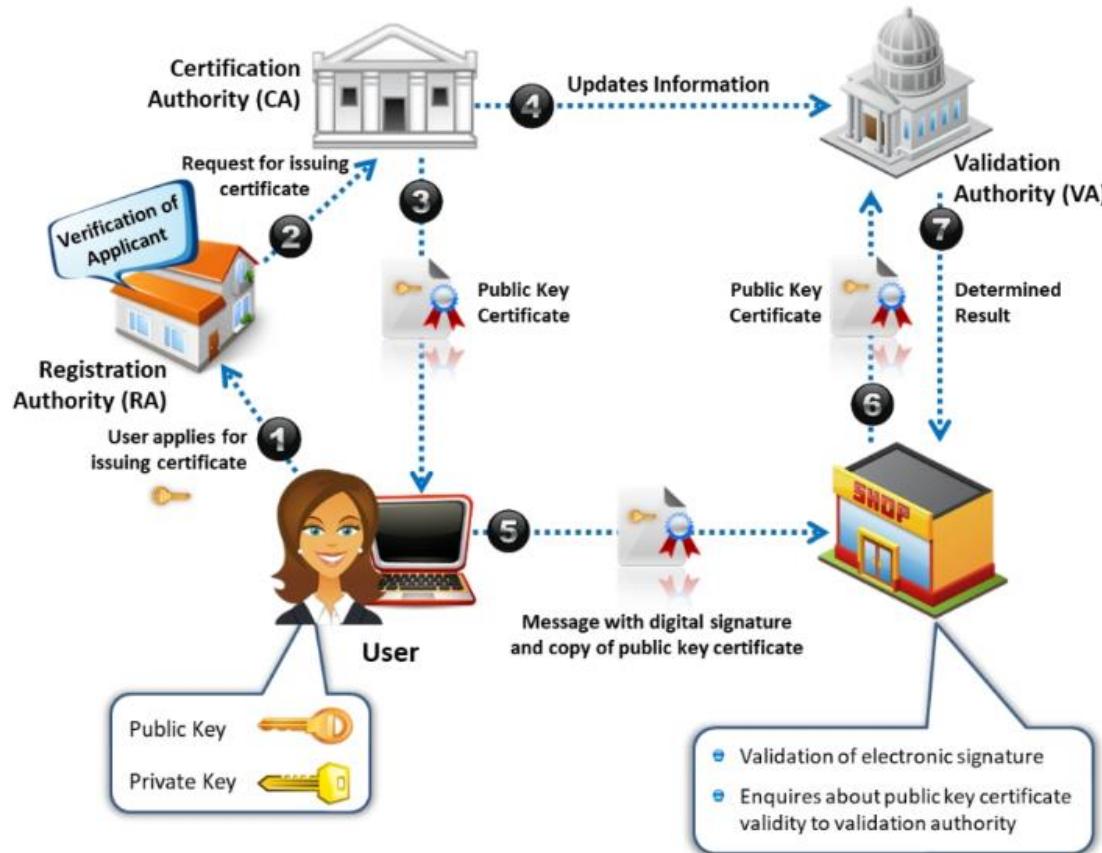
- L'espion ne pourra pas faire passer sa clé pour une autre puisqu'il devrait signer le certificat avec la clé privée de l'organisme de certification

# Infrastructure à clés publiques

- **Composants d'une PKI**
  - Certificate Authority (CA)
    - Émet et révoque des certificats
  - Registration Authority (RA)
    - Agit comme vérificateur d'identité pour le CA
  - Validation Authority (VA)
    - Détient les certificats (avec leur clé publique)
  - End User :
    - Demande, utilise, et gère des certificats
  - Digital Certificates
    - Établit l'identification d'une personne lors de transactions en ligne

# Infrastructure à clés publiques

## • Suivi



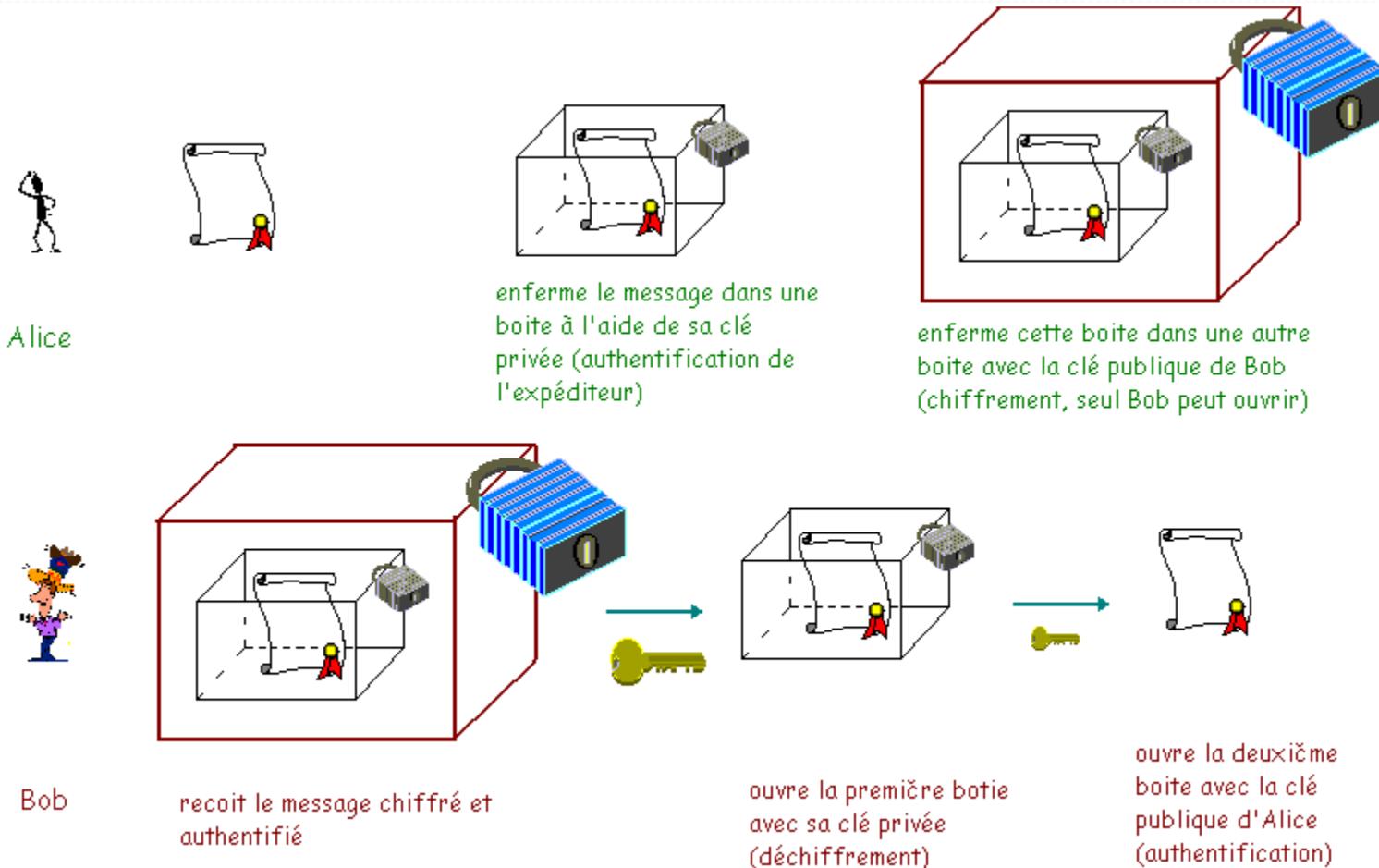
1. Le sujet (utilisateur, société ou système) demande un certificat auprès de l'autorité d'enregistrement (RA).
2. L'autorité d'enregistrement vérifie son identité et demande à l'autorité de certification de lui attribuer un certificat de clé publique.
3. L'autorité de certification émet le certificat avec la clé publique du sujet.
4. Ensuite il envoie les informations mises à jour à l'autorité de validation.
5. Lorsqu'un utilisateur effectue une transaction, il signe le message avec le certificat de clé publique.
6. Le destinataire vérifie l'authenticité de l'utilisateur en s'informant de la validité du certificat avec le VA.
7. Le VA compare le certificat de clé publique de l'utilisateur avec celui qu'il détient et détermine le résultat (qu'il soit valide ou non).

# Infrastructure à clés publiques

- **Autorité de certification privée**

- Une entreprise peut être sa propre autorité de certification
- Utilisé en Intranet mais d'aucune utilité en dehors
- La machine dédiée à l'attribution et à la gestion des certificats doit être sécurisée (isolée physiquement la plupart du temps)
- En général, une arborescence est créée pour former une hiérarchie de confiance. Le certificat racine étant auto-signé.
- Si la clé racine est compromise, tous les serveurs le sont.

# Synthèse



# Synthèse

- **Alice**

- crée un condensé du message (hachage)
- chiffre le condensé avec sa clé secrète (signature)
- envoie
  - sa signature
  - le message crypté (clé publique de Bob)
  - et le certificat (clé publique Alice)

- **Bob**

- Possède, via un programme de navigation ou tout autre logiciel de communication, la clé publique du CA
- vérifie la signature
  - regarde la signature officielle apposée sur le certificat
  - déchiffre la signature avec la clé publique d'Alice (condensé du message)
  - déchiffre le message crypté, applique le hachage (condensé)
  - compare les deux condensés... Non altéré +Alice?

# Stéganographie



Extraction



<https://en.wikipedia.org/wiki/Steganography#Printed>

# Stéganographie

- Introduction
- Procédé
- Utilisation...

# Introduction

- **Historique**

- Vième siècle avant JC : Grecs : rasage des cheveux
- Tablettes de cire (bois gravé recouvert de cire)
- Textes invisibles à l'œil et lisibles à la lumière ou aux réactifs chimiques
- Dissimulation du message dans le texte lui-même
- Micro-films placés dans des timbres ou couvertures de magazines

- **Informatiquement**

- Glisser des bits discrets dans des lecteurs mp3, images, textes ou programmes

# Procédé

- **Messages dans des programmes exécutables**
  - Ajout dans le programme d' une quantité de code jamais atteinte
  - Exemple : après l'accolade finale ou dans des commentaires
- **Messages dans des parties libres de disques durs**
  - Message crypté dans la partie libre sans écrire dans l'index où se trouve la taille d'allocations des fichiers
  - Connaître l'adresse permet de le retrouver
- **Messages dans des fichiers multimédias**
  - Image = ensemble de pixels
  - Pixel = ensemble de 3 nombres codés sur 8 bits (RGB) où
    - R : intensité du rouge (0 à 255 nuances),
    - G : vert
    - B : bleu
  - Possibilité de  $(2^8)^3 = 16\ 777\ 216$  couleurs différentes

# Procédé

- **Exemple simple**

- Image initiale :

R<sub>1</sub> 01001110

R<sub>2</sub> 01110011

G<sub>1</sub> 01101111

G<sub>2</sub> 01110110

B<sub>1</sub> 11111111

B<sub>2</sub> 10101010

- Les 2 bits de droite vont être modifiés (de 3 ou moins) → peu de changement dans l'intensité
  - Soit M 101100001101 à placer
    - par groupe de 2, on les place à droite des éléments des 2 pixels
    - R<sub>1</sub> 01001110, G<sub>1</sub> 01101111, B<sub>1</sub> 11111100  
R<sub>2</sub> 01110000, G<sub>2</sub> 01110111, B<sub>2</sub> 10101001

# Procédé

- Possibilité de cacher de informations de taille élevée
    - Dans une image 200\*200, on peut cacher 200\*200\*6 bits (240000 bits ou 30.000 caractères)
  - Plus perfectionné
    - Les 2 bits de droite ne sont plus remplacés par les 2 bits du message mais par 2 bits droite  $\oplus$  2 bits message
    - M chiffré par TDES ou AES ,  
C caché dans l'image
- moins détectable :  
les bits de poids le plus faible étant devenus aléatoires

Le temps à l'espion de trouver la clé...

# Utilisation...

- **Un mode de propagation des malwares pour éviter la détection des antivirus**
  - Ordinateurs infectés grâce à des fichiers, à priori, innocents
  - Exemple : virus SyncCrypt (août 2017) qui récupère une image jpg dans laquelle se trouve un fichier ZIP à la fin de l'image
    - Ce fichier ZIP contient les composants nécessaires à l'infection de l'ordinateur.
- **Le Watermarking**
  - Utilisé dans l'industrie actuelle.
  - Cacher un copyright au sein d'une œuvre protégée
    - Si litige de droits d'auteurs, le watermark sera montré pour prouver l'originalité de l'oeuvre



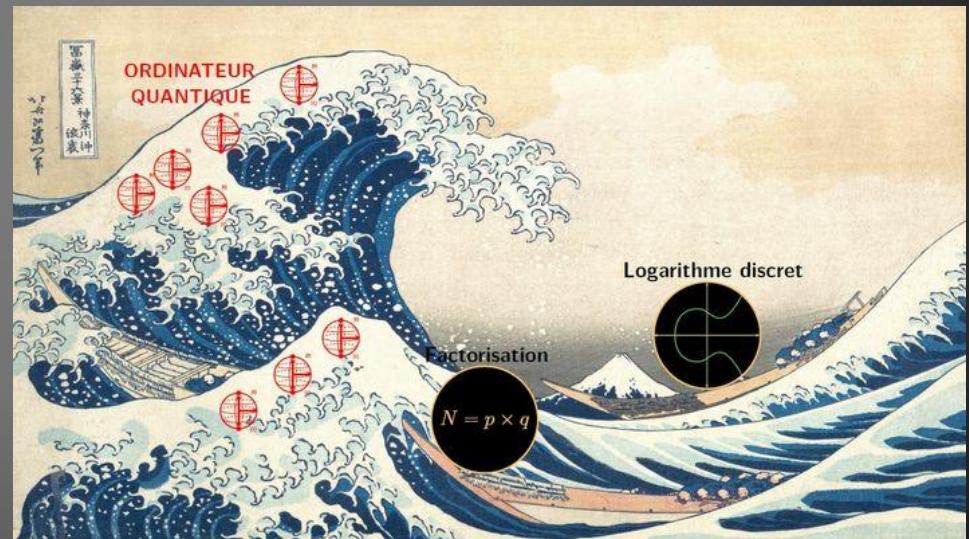
Visible Watermarking



Invisible Watermarking

<https://www.slideshare.net/ankushkr007/digital-watermarking-15450118>

# Cryptographie quantique



<https://www.franceculture.fr/recherche?q=cryptographie+quantique>

# Cryptographie quantique

- Introduction
- Protocole BB84
- Avenir

# Introduction

## ● Sécurité ???

- Sécurité des algorithmes à clé publique pour échanger des clés secrètes assurée jusque ... Les algorithmes mathématiques difficiles à inverser le seront peut-être un jour !
- Dans les années 1990, P. Shor a écrit un algorithme qui permettra de factoriser très rapidement en nombres premiers quand les ordinateurs quantiques seront au point (RSA KO dès lors)
- Création d'un ordinateur quantique « digne de ce nom » : Google, IBM, Microsoft, D-Wave ...

## **Les ordinateurs quantiques lancent un défi à la cryptographie**

L'Institut américain des standards a lancé un concours pour renforcer la sécurité des protocoles d'échanges électroniques en prévision de l'apparition des ordinateurs quantiques.

[http://www.lemonde.fr/sciences/article/2017/07/04/les-ordinateurs-quantiques-lancent-un-defi-a-la-cryptographie\\_5155407\\_1650684.html](http://www.lemonde.fr/sciences/article/2017/07/04/les-ordinateurs-quantiques-lancent-un-defi-a-la-cryptographie_5155407_1650684.html)

# Introduction

- **Ensemble de protocoles**
  - permettant de distribuer une clé de chiffrement entre 2 entités distantes pour assurer la sécurité de la transmission grâce aux lois de la physique quantique et de la théorie de l'information
- **Sécurité des communications**
  - Construction de clés secrètes de chiffrement en direct au moment du besoin
    - via les fibres optiques à l'issue des communications
    - via la communication avec des satellites
    - par la mesure des oscillations de la plus petite unité quantique de la lumière : le photon (oscillation : polarisation)
  - Sécurité assurée
    - par un des principes de la physique quantique
    - "Toute mesure de la polarisation d'un photon (toute écoute de la part de l'espion) entraîne des erreurs sur la mesure suivante"

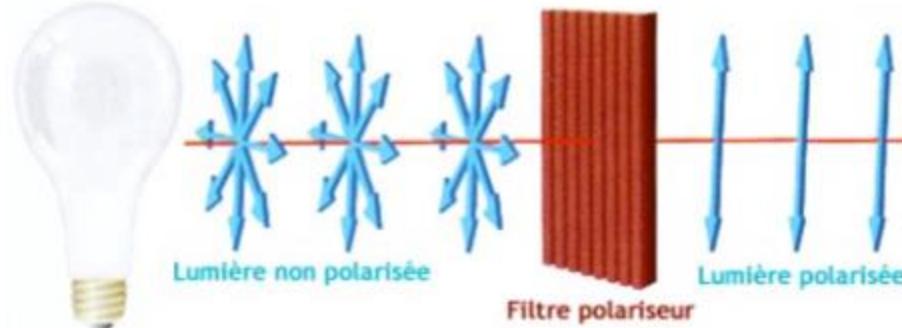
# Introduction

- **Echange d'informations**

- au moyen d'impulsions lumineuses voyageant sur des réseaux de fibres optiques
  - pour chaque bit, une impulsion est émise et transmise via la fibre à un récepteur qui la détecte et la transforme en signal électrique
- Cryptographie quantique
  - Les impulsions sont constituées d'un unique photon (quantité d'énergie minuscule) ou paire de photons.
  - Échange de clé : Q(uantum) K(ey) D(istribution)
- Plusieurs protocoles BB84, E91, MSZ96, DPS02, SARG04, KMBo9

# Introduction

- Un peu de physique



<https://docplayer.fr/917299-La-cryptographie-quantique.html>

# Introduction

- Détection des polarisations
  - Filtre polarisant suivi d'un détecteur de photons
  - Filtre orienté à  $0^\circ$ 
    - Photon à  $0^\circ$  polarisé
      - Il traverse le filtre
      - Il est enregistré par le détecteur
    - Photon à  $90^\circ$  polarisé
      - Il est stoppé
      - Le détecteur n'enregistre rien

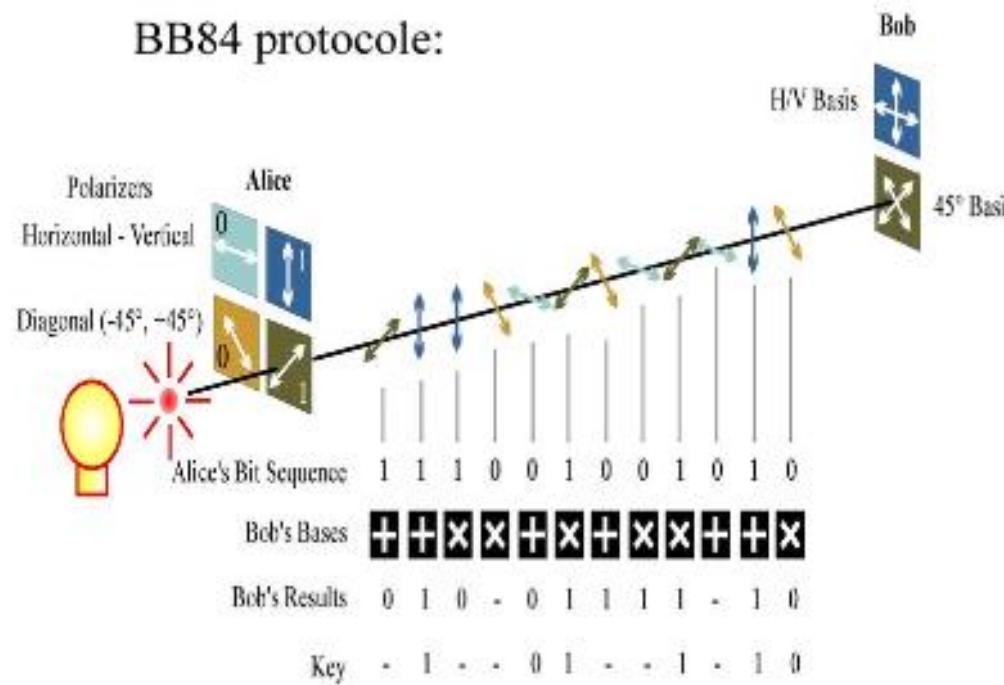
# Introduction

- **Détection des polarisations**

- Filtre orienté à  $0^\circ$ 
  - Photon diagonalement ( $45^\circ$  ou  $135^\circ$ ) polarisé
    - Une fois sur deux, il traverse le filtre
    - Une fois sur deux, il est stoppé
    - Si on peut distinguer entre une polarisation à  $0^\circ$  et à  $90^\circ$ , il est impossible de distinguer en même temps entre une polarisation à  $45^\circ$  et à  $135^\circ$
- Filtre orienté à  $45^\circ$  : il laisse passer les photons polarisés à  $45^\circ$ , stoppe ceux polarisés à  $135^\circ$ , et se comporte aléatoirement avec ceux à  $0^\circ$  et  $90^\circ$ !

# Protocole BB84

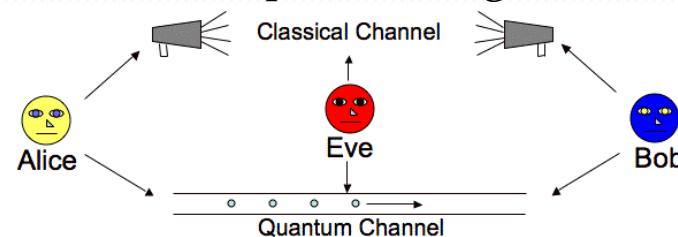
## ● Schéma



<http://docplayer.fr/2703021-Communication-quantique-jongler-avec-des-paires-de-photons-dans-des-fibres-optiques.html>

# Protocole BB84

- Protocole créé par Bennett Ch. et Brassard G. en 1984
  - Utilisation de 2 canaux
    - Canal quantique
      - Mesure des photons polarisés
    - Canal classique publique
      - Pour discuter et comparer les signaux envoyés sur le canal quantique



<https://www.cse.wustl.edu/~jain/cse571-07/ftp/quantum/>

- Espion sur le canal quantique
  - Il pourra mesurer correctement
    - soit la vitesse du photon et non l'état
    - soit l'état du photon et non la vitesse

# Protocole BB84

- Polarisation avec 4 valeurs :
  - $0^\circ$  et  $90^\circ$  : polarisation rectiligne
  - $45^\circ$  et  $135^\circ$  : polarisation diagonale



- Convention dans l'exemple
  - $0^\circ$  et  $135^\circ$  : bit = 0
  - $90^\circ$  et  $45^\circ$  : bit = 1

# Protocole BB84

- **Etapes**

- Alice encode sa suite de bits en sélectionnant de manière aléatoire la base verticale/horizontale ou la base diagonale/anti-diagonale sans révéler ses choix à personnes.  
Les photons sont ensuite transmis à Bob via le canal quantique.
- Bob reçoit les photons et enregistre les résultats en choisissant de manière aléatoire une des deux bases d'analyse.
- Alice communique à Bob via un canal publique (par communication classique) ses choix de bases (non la valeur binaire associée à chaque photon!)
- Bob compare ses choix de base avec ceux d'Alice.  
Il identifie le sous ensemble de bits correspondants aux cas où ils ont tous les deux choisis la même base.
- Bob communique ensuite à Alice via le canal publique les positions correspondantes dans la séquence, les autres bits sont alors éliminés.

# Protocole BB84

- Exemple

Quantum transmission & detection	ALICE sends photons							
	ALICE's random bits	0	1	0	1	1	1	0
	BOB's detection events							
	BOB's detected bit values	1	1	0	1	1	1	0
Public discussion (i.e., sifting)	BOB tells ALICE the basis choices he made							
	Rect	Diag	Diag	Rect	Diag	Diag	Diag	Diag
	ALICE tells BOB which bits to keep		✓		✓		✓	✓
	ALICE and BOB's shared sifted key	-	1	-	1	-	1	0

<http://physique.unice.fr/sem6/2014-2015/PagesWeb/PT/Tomographie/?page=bb84>

# Protocole BB84

## • Construction de la clé

- Elimination des cas où les détecteurs n'ont pas enregistré de photons (efficacité : pas 100%)
- Ecoute espion ?
  - Ils choisissent au hasard un sous-ensemble de données de polarisation et le comparent publiquement.
  - Si la comparaison montre l'écoute, ils éliminent tout et recommencent.
  - Sinon, leur clé sera celle choisie hormis le sous-ensemble.

# Avenir

Jusqu'à présent, il n'était possible d'encoder qu'un bit par photon. Les chercheurs ont découvert qu'ils pouvaient encoder deux bits par photon en ajustant le temps de libération des photons et en exploitant des détecteurs de photos à très haute vitesse pour suivre ces changements.

[Extrait de [https://www.silicon.fr/cybersecurite-un-pas-de-geant-franchi-dans-la-cryptographie-quantique-191661.html?inf\\_by=5beee880671db8c5708b51b7](https://www.silicon.fr/cybersecurite-un-pas-de-geant-franchi-dans-la-cryptographie-quantique-191661.html?inf_by=5beee880671db8c5708b51b7), 2017]

Fruit du travail des chercheurs de Toshiba Research Europe à Cambridge (au Royaume-Uni), ce dispositif atteint une vitesse de distribution de 13,7 mégabits par seconde, soit 7 fois la vitesse du premier prototype de ce dispositif, présenté en 2016, et qui affichait une vitesse de 1,9 Mbit/s.

[Extrait de [https://www.silicon.fr/innovation-toshiba-chiffrement-quantique-197635.html?inf\\_by=5bec6942671db84d338b4c7e](https://www.silicon.fr/innovation-toshiba-chiffrement-quantique-197635.html?inf_by=5bec6942671db84d338b4c7e), 2018]

# Avenir

## La Chine, leader des communications quantiques

Grâce au satellite Micius, des chercheurs chinois ont réussi à téléporter des informations entre l'espace et la Terre. Une première qui assoit leur domination sur ces échanges futuristes hautement sécurisés

<https://www.letemps.ch/sciences/chine-leader-communications-quantiques>

Le satellite chinois de communication quantique "Micius" a permis aux scientifiques de réaliser la distribution de clé quantique entre la Chine et l'Autriche, jetant une base pour la construction d'un réseau mondial de communication quantique.

<http://french.peopledaily.com.cn/n3/2018/0123/c31357-9418654.html>

Et dans les langages

# Librairies

- En Java
  - API JCA/JCE permettent d'utiliser des technologies de cryptographie
  - API JSSE permet d'utiliser le réseau au travers des protocoles sécurisés SSL ou TLS
  - API JAAS propose un service pour gérer l'authentification et les autorisations d'un utilisateur
- En .Net
  - Namespace System.Security.Cryptography
    - Services de chiffrement, comprenant l'encodage et le décodage sécurisé des données, le hachage, la génération aléatoire de nombres et l'authentification de messages

# Librairies

- En JavaScript
  - [cryptojs.altervista.org](http://cryptojs.altervista.org)
  - <https://www.nccgroup.trust/us/about-us/newsroom-and-events/blog/2011/august/javascript-cryptography-considered-harmful/>

# Quelques pas en cryptanalyse

# Cryptanalyse

- Introduction
- Types d'attaque
- Algorithmes les plus connus

# Introduction

- **Objectifs**

- Décrypter un message sans avoir la clé
- Conséquence : l'algorithme testé est-il robuste?

- **Attaque**

- Processus pour comprendre un message (ou le détériorer)
- Passive ou active

# Cryptanalyse : Attaques

- **Attaque basique : recherche exhaustive de la clé**
  - Essayer toutes les clés possibles jusqu'à trouver... technique très lourde
- **Analyse des fréquences de lettres ou de groupes de lettres**
- **Technique plus puissante : technique du mot probable**
  - Supposer qu'une suite de lettres du texte chiffré correspond à un mot qu'on devine

# Cryptanalyse : Attaques

- **Attaque à texte chiffré seulement (*Ciphertext-only*)**
  - Dispose du texte chiffré de plusieurs messages, tous chiffrés avec le même algorithme
  - Objectif
    - Retrouver le plus grand nombre de messages clairs en faisant des hypothèses
    - Ou mieux retrouver la(les) clé(s) utilisée(s)
- **Attaque à texte clair connu (*Known plaintext*)**
  - Dispose des textes chiffrés avec la même clé et les textes clairs correspondants
  - Objectif
    - Retrouver la(les) clé(s) utilisée(s) pour chiffrer
    - Retrouver l'algorithme qui permet de déchiffrer d'autres messages chiffrés avec ces mêmes clés

# Cryptanalyse : Attaques

- **Attaque à texte clair choisi (*Chosen plain text*)**
  - Dispose de textes en clair, peut créer des versions chiffrées avec l'algorithme
  - Attaque plus efficace que l'attaque à texte clair connu
    - Choix parmi des textes en clair spécifiques qui donneront plus d'informations sur la clé
- **Attaque à texte chiffré connu (*Chosen cipher text*)**
  - Choix entre différents textes chiffrés à déchiffrer.  
Le cryptanalyste demande la version en clair de certains pour mener l'attaque.
  - Objectif :
    - Retrouver la clé

# Cryptanalyse : Algorithmes

- Cryptanalyse différentielle
  - Par Biham & Shamir à la fin des années 1980
  - Technique du texte clair choisi
    - Étude sur la manière dont les différences entre les données en entrée affectent les différences de leurs sorties
    - Ensemble des techniques
      - permettant de retracer les différences à travers le réseau des transformations
      - en conséquence, découvrant où l'algorithme montre un comportement prédictible
      - exploitant ainsi ces propriétés afin de retrouver la clé secrète

# Cryptanalyse : Algorithmes

- Cryptanalyse linéaire
  - Par Hatsui en 1993
  - Processus du texte clair connu
    - Retrouver (de l'information sur) la clé
    - Trouver des relations linéaires de dépendance via les probabilités entre les bits d'entrée et de sortie
    - Une relation linéaire ne peut être vraie pour tous les messages sinon le protocole a une faiblesse!
- Cryptanalyse différentielle linéaire
  - Introduite par M. Hellman et S. K. Langford en 1994
  - Combinaison des deux autres