Introduction à la cryptographie

Table des matières

Histoire de cryptographie

Carré de Polybe

Principe

C'est une matrice qui fait correspondre des caractères à une paire de chiffres (les coordonnées du caractère dans le tableau).

Exemple : « H » correspond à « 23 »

	1	2	3	4	5
1	A	B	C	D	E
2	F	G	H	I,J	K
3	L	M	N	O	P
4	Q	R	S	T	U
5	V	W	X	Y	Z

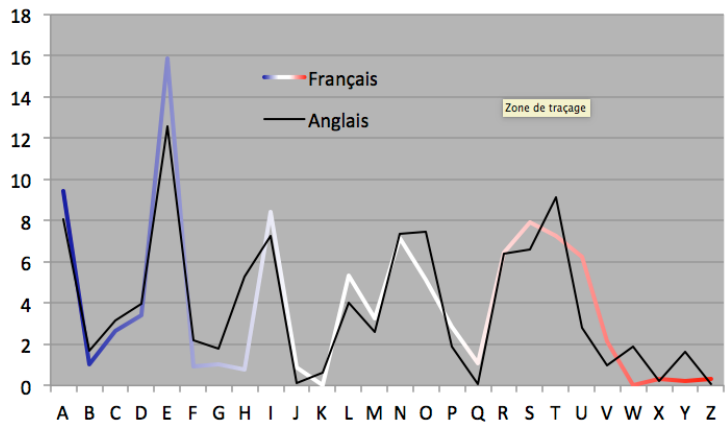
Problèmes/Solutions

1. Dès que l'on connaît la matrice de correspondance on peut tout convertir. Il n'y a plus de secret.
   ⇨ En introduisant un code dans la matrice, cette dernière change : la connaissance du principe de chiffrement ne suffit plus pour décoder les messages, il faut en outre connaître le code.

2. Si l'on a la correspondance d'un grand texte dont on connaît la langue, on peut facilement trouver les lettres en réalisant une analyse de fréquence.
   ⇨ Ce problème n'est pas évitable avec le carré de Polybe sauf si l'on peut changer le code à chaque échange.

Analyse de fréquence

L'analyse de fréquence part du constat que dans une langue les symboles ne sont pas tous utilisés autant. La répartition statistique de l'usage de ces symboles peut être mesurée ; il suffit ensuite de faire correspondre la fréquence d'apparition des caractères encodés avec celle des lettres associées.



Introduction aux protocoles de sécurité

Pour améliorer le carré de Polybe et rendre l'analyse de fréquence plus difficile, il est possible de changer de matrice à chaque échange.

Pour cela, l'émetteur part d'un code convenu avec son destinataire, et choisit le prochain code convenu, qu'il place dans son message.

Le destinataire peut déchiffrer le premier message car il connaît le premier code convenu, et ce faisant connaître le prochain code convenu. Il n'a donc plus qu'à répondre de la même manière, en utilisant cette fois le second code convenu et en précisant dans sa réponse le troisième.

Ainsi de suite.

L'analyse de fréquence n'est possible que sur un échange qui, s'il n'a que peu de caractères, n'en permet pas de pertinente (l'échantillon de texte doit être assez gros pour que l'analyse de fréquence, basée sur des statistiques d'occurrence d'apparition de lettres, soit efficace).

Autres possibilités :

- Changer le code plusieurs dois au sein d'un même message, de manière à maintenir le nombre de caractères utilisant le même code toujours trop faible pour une analyse en fréquence. Par exemple : « Bonjour, comment vas-tu nc LISA moi ça va bien » la chaine « nc » signifiant que le mot suivant constitue le nouveau code à utiliser pour la suite du message ; ici « LISA », pour décoder « moi ça va bien ».
- Introduire des mots sans sens avec des caractères qui modifient les statistiques.

Chiffre de César

Principe

Il s'agit d'un « **codage alphabétique** ». Comme pour le carré de Polybe, il s'agit d'une autre forme de substitution.

Le principe consiste à associer à chaque lettre de l'alphabet une autre lettre de l'alphabet, par permutation circulaire de n caractères.

Par exemple, avec n=1 : A→B, B→C, C→D, etc. « HAL » → « IBM » (cf. 2001 l'odyssée de l'espace).

Problème

Une lettre donnée correspond toujours à la même autre lettre. Donc l'analyse de fréquence est possible, toujours dans la mesure où le nombre de caractères du message est suffisant.

Chiffre de Vigenère

Principe

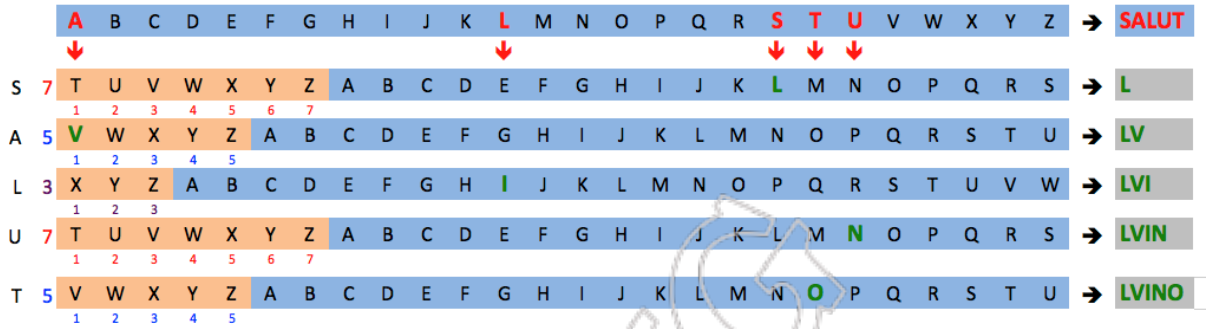
Il s'agit d'un « **codage poly-alphabétique** ».

Le principe consiste à réaliser un chiffre de César avec un nouveau décalage pour chaque lettre. L'ensemble des décalages constitue le code secret partagé entre les interlocuteurs. Cet ensemble est

généralement fini et assez petit donc lorsqu'on a épuisé tous les décalages de l'ensemble on recommence au premier.

Par exemple :

Si le code est 123, on fera un décalage de 1 pour trouver la correspondance de la première lettre du message, puis un décalage de 2 pour connaître celle de la seconde lettre, puis un décalage de 3 caractères pour connaître la correspondance du troisième. Et on recommence au avec un décalage de une lettre pour le quatrième caractère, puis 2 lettres etc.



Problème

L'analyse de fréquence est aussi possible sur toutes les sous-chaines utilisant le même décalage, donc le même chiffre de César.

Néanmoins, elle est un peu plus compliquée à mettre en œuvre. En effet, l'attaquant ne connaissant pas la longueur du code, il va devoir tester plusieurs possibilités jusqu'à trouver la bonne.

Chaque sous-chaine ne contient qu'une fraction des caractères de la chaine initiale (son nombre divisé par le nombre de caractères du code).

Illustration : si le codage est réalisé avec deux décalages, le 1er caractère sera codé avec le 1er décalage, le 2nd avec le 2nd, le 3ème avec le 1er à nouveau, le 4ème avec le 2nd, etc. On voit bien que : un caractère sur deux sera codé avec le 1er décalage, et l'autre moitié de la chaine avec le 2nd décalage. Et si on utilisait 3 décalages, ce serait un sur trois avec chacun des décalages.

Plus le code est long, moins les sous-chaînes le seront et plus il faudra faire d'essais pour trouver le bon nombre de décalages. Donc plus le code est long, plus le décodage est complexe et sa probabilité de pertinence faible.

A la limite, si le code fait la même longueur que le message, l'analyse en fréquence ne devrait plus fonctionner du tout.

RSA

Système de chiffrement à clé **asymétrique**, basé sur le fait qu'il est très difficile de décomposer le produit de deux nombre premiers de grande taille.

Un message chiffré avec le produit des deux nombres premiers peut facilement être décodé lorsqu'on connaît ces deux facteurs.

En revanche, connaissant le produit de ces deux facteurs il est aujourd'hui impossible de trouver ces deux facteurs premiers permettant le décodage.

Le MIT a déposé un brevet en 1983, tombé dans le domaine public (un peu avant l'expiration du brevet : 06/09/2000), le 6 septembre 2000.

DES (1977), AES (2001)

Data Encryption Standard (National Bureau of Standards à Federal Information Processing Standards)

Advanced Encryption Standard (NIST)

Il s'agit d'algorithmes de chiffrement symétriques.

PGP (P.Zimmerman)

Premier à mettre à disposition du grand public un logiciel de chiffrement simple à utiliser, en 1991.

⇒ Il a subit **3 ans d'enquête criminelle** par les douanes américaines (il était interdit d'exporter des logiciels de cryptographie)

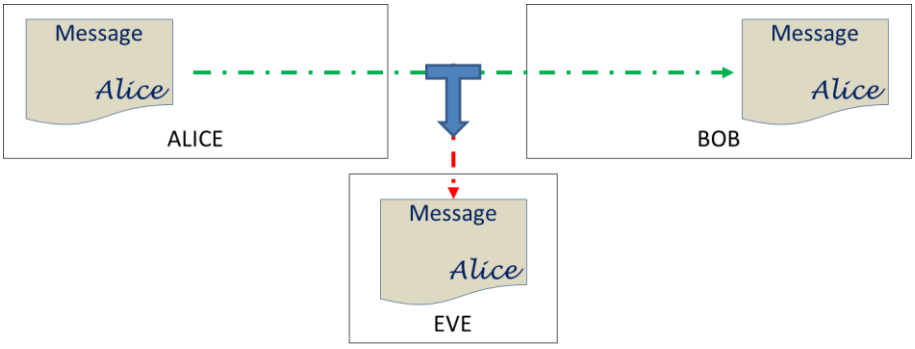
Extrait de « Why I wrote PGP » :

« (…)
What if everyone believed that law-abiding citizens should use postcards for their mail? If a nonconformist tried to assert his privacy by using an envelope for his mail, it would draw suspicion. Perhaps the authorities would open his mail to see what he's hiding.
(…)
If we do nothing, new technologies will give the government new automatic surveillance capabilities that Stalin could never have dreamed of. The only way to hold the line on privacy in the information age is strong cryptography.
(…)
When use of strong cryptography becomes popular, it's harder for the government to criminalize it. Therefore, using PGP is good for preserving democracy. If privacy is outlawed, only outlaws will have privacy.
(…)
PGP empowers people to take their privacy into their own hands. There has been a growing social need for it. That's why I wrote it. »

Principes de cryptographie

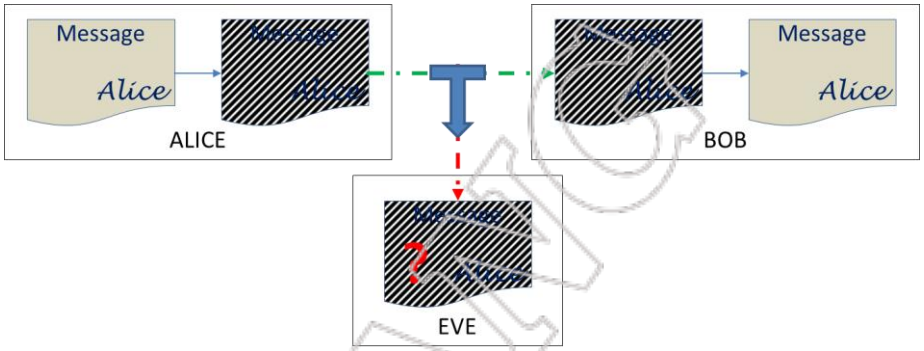
Pourquoi faire de la cryptographie ?

Pour assurer la confidentialité et l'intégrité des messages.



INTERCEPTION

Avec de la cryptographie :



DECRYPTAGE NÉCESSAIRE…

Principes de Kherkhoffs

(Source Wikipedia)

Auguste Kerckhoffs von Nieuwenhoff (19/01/1835 – 09/08/1903) est un cryptologue militaire néerlandais. Son essai La Cryptographie militaire (1883) constitue une référence de la cryptographie du XIXe siècle. À l'époque, l'une des préoccupations des cryptographes était de mettre en place un réseau de télégraphie sécurisé.

Son œuvre présente ce qu'on appelle aujourd'hui le principe de Kerckhoffs en cryptographie stratégique.

Kerckhoffs énonce les règles que doit respecter un système cryptographique pour assurer une communication confidentielle :

- Le système doit être matériellement, sinon mathématiquement, indéchiffrable ;
- Il faut qu'il n'exige pas le secret, et qu'il puisse sans inconvénient tomber entre les mains de l'ennemi ;

- La clef doit pouvoir en être communiquée et retenue sans le secours de notes écrites, et être changée ou modifiée au gré des correspondants ;
- Il faut qu'il soit applicable à la correspondance télégraphique ;
- Il faut qu'il soit portatif, et que son maniement ou son fonctionnement n'exige pas le concours de plusieurs personnes ;
- 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.

En résumé

La sécurité d'un Système de chiffrement doit tenir dans le secret de la clé et non dans celui de l'algorithme

Pourquoi

Ce principe répond à trois points principalement :

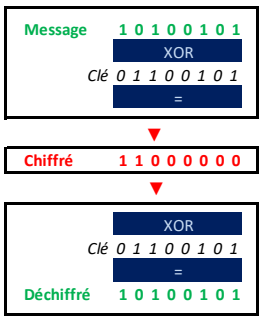
- **Faible évolutivité des algorithmes** : une fois implémenté et distribué, il est complexe de revenir en arrière sur un algorithme. Des implémentations matérielles sont parfois intégrées dans les ordinateurs grand public. Le reverse engineering est possible et le maintien du secret difficile.
- **Besoin de partage** : pour que le Système de chiffrement fonctionne, l'algorithme doit être partagé. Ce partage est incompatible avec le secret.
- **Evaluation et fiabilisation** : les spécialistes aux intérêts indépendants de l'algorithme (et entre eux) vont pouvoir analyser l'algorithme à la recherche d'éventuels défauts de conception.

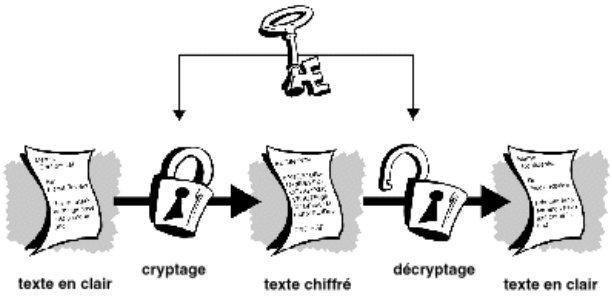
➔ **Soyez méfiants vis-à-vis des algorithmes secrets, et privilégiez les algorithmes ouverts**

Types de chiffrements

Chiffrement symétrique

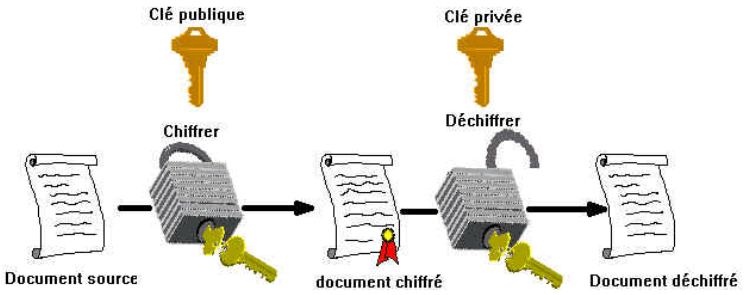
Ce type de chiffrement consiste à utiliser la même clé pour chiffrer et déchiffrer le message.





Chiffrement asymétrique

Contrairement au précédent, ce type de chiffrement consiste à chiffrer et déchiffrer avec deux clés complémentaires. La première, (publique) ne servant qu'à chiffrer, et la seconde (privée) qu'à déchiffrer.

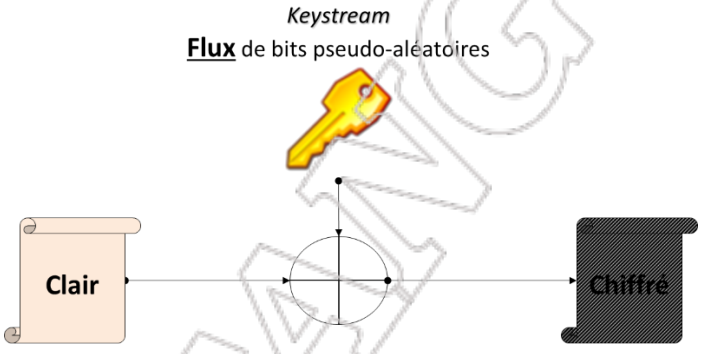


Famille de chiffrement

Chiffrement par flot

Ce chiffrement consiste à transformer bit à bit un message par une opération (par exemple XOR) à partir d'une suite pseudo aléatoire de bits, générée au fil de l'eau.

Un algorithme produit la suite pseudo aléatoire à partir d'une clé symétrique.



Chiffrement par blocs

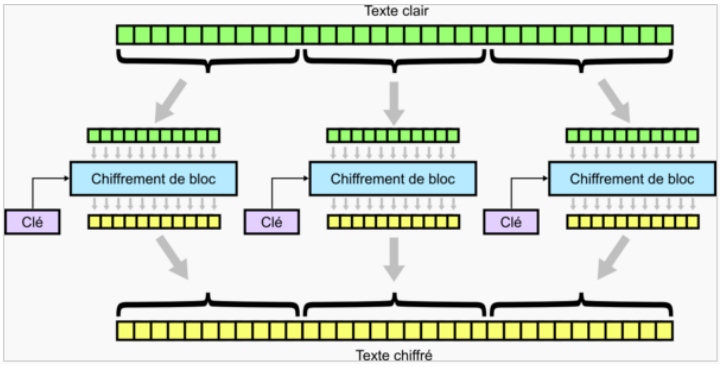
Ce chiffrement consiste à transformer des blocs d'octets par un algorithme cryptographique, à partir d'une **clé symétrique**.

En général, les blocs font 8 ou 16 octets.

Modes d'opération

ECB (Electronic Code Book)

Principe



Problème

- Tous les blocs sont chiffrés avec la même clé : deux blocs identiques auront le même résultat chiffré. Ses caractéristiques statistiques sont relativement inchangées (cf. exemple page suivante).
- Les blocs font tous la même taille : si le dernier bloc contient un padding connu (clair connu), alors il est possible d'extraire une partie de la clé.

Le schéma ci-dessous montre l'effet du chiffrement d'une image en mode ECB : le chiffrement n'a fait que décaler l'image comme un filtre. Cette dernière garde ses caractéristiques d'ensemble et reste reconnaissable à l'œil. Ces motifs « *reconnaissables* » permettent de monter des attaques.

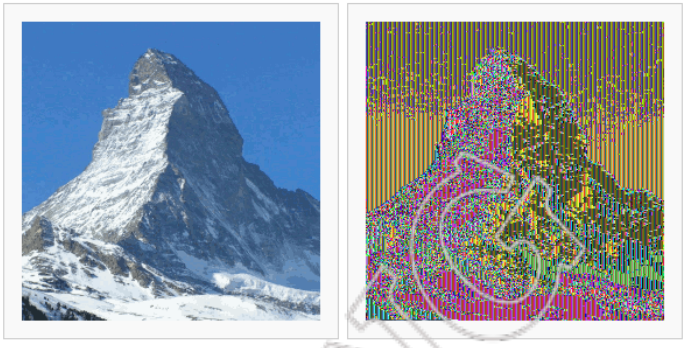


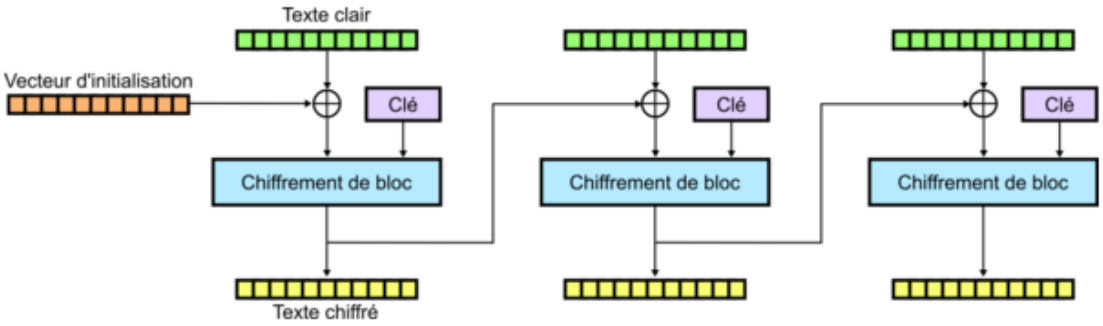
Image originale — Chiffrement en mode ECB

Le mode d'opération ECB est très déconseillé

CBC (Cipher block chaining)

Principe

Il s'agit d'un mode de **chiffrement par BLOCS** : le message à chiffrer est aussi tronçonné en blocs de même taille et on complète le dernier bloc si besoin en faisant du « padding ». Mais cette fois-ci le résultat de chiffrement du dernier bloc sert de diffusion du message en clair pour le chiffrement du bloc suivant. C'est la même clé qui est utilisée pour le chiffrement de tous les blocs.



Problèmes

- Une erreur à un instant t se répercute sur tout le reste du message.
- Aucune parallélisation des chiffrements possible
- Ce mode ne permet pas de déchiffrer sélectivement une partie seulement du message.

Image originale | Chiffrement avec un mode sûr (autre que ECB)

Le mode d'opération CBC est conseillé. Mais il faut en connaître ses limitations.

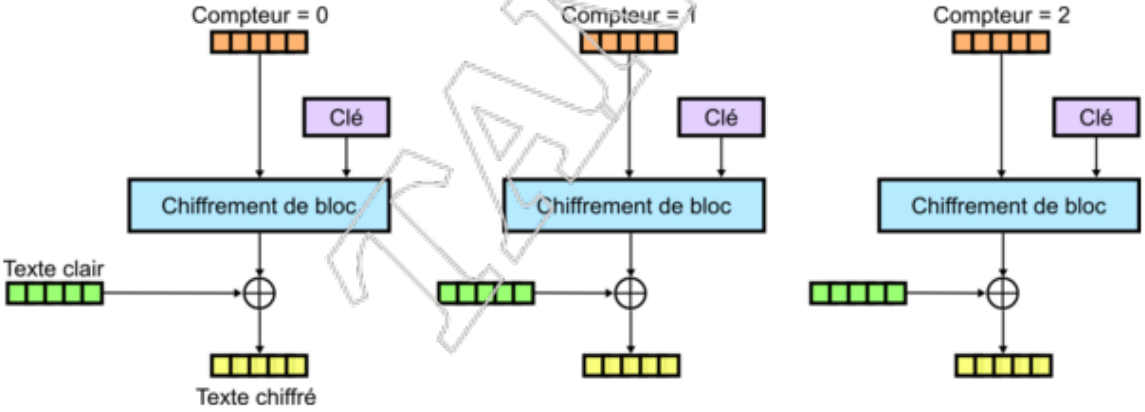
CTR (CounTeR)

Principe

On utilise des chiffrements par BLOCS pour faire du chiffrement par FLOT.

Pour chaque bloc du message :

- Un compteur (dont la valeur dépend du bloc) est chiffré avec une clé (qui ne change pas)
- La valeur chiffrée du compteur est ensuite mixée (XOR) avec le texte en clair pour obtenir le bloc chiffré
- Les blocs sont traités indépendamment les uns des autres.



Avantages

- Chaque bloc peut être décodé indépendamment des autres
- L'opération de mixage (XOR) est réalisée avec une valeur qui change pour chaque bloc : les attaques par analyse de fréquence sont peu probables.
- On peut considérer qu'il s'agit d'un chiffrement par FLOT, le keystream étant la suite de compteurs chiffrés avec la clé. Le keystream est calculable à l'avance : cela rend les opérations de chiffrement/déchiffrement très rapides.

Le mode d'opération CTR est conseillé.

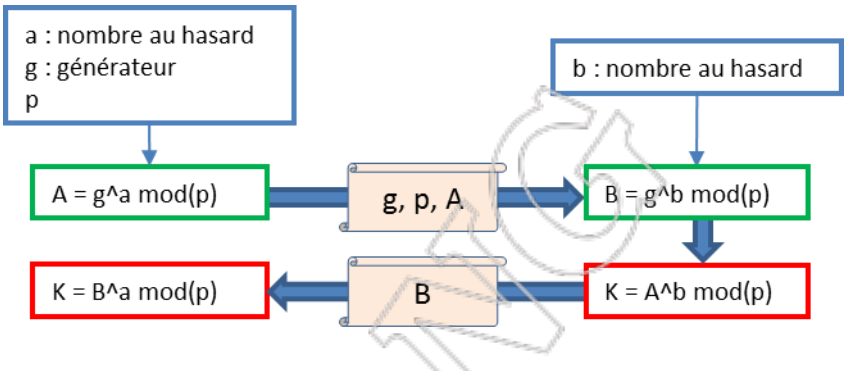
Echange de clés

Diffie-Hellman

C'est un algorithme d'échanges de clés basé sur le problème mathématique des *logarithmes discrets*.

Le gros avantage de ce système réside dans le fait qu'il est possible d'établir une communication chiffrée avec un interlocuteur sans avoir à partager un secret. Ce secret est en effet le résultat d'un calcul que chacun des interlocuteurs peut réaliser à partir d'un élément connu de lui seul (un nombre aléatoire choisi par ses soins).

Il est à ce jour réputé impossible de trouver la clé de chiffrement à partir des seuls éléments échangés entre les deux interlocuteurs.

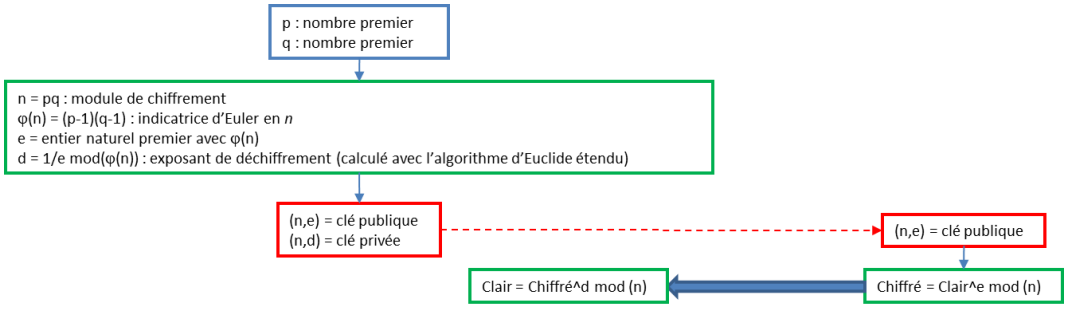


K = A^b mod(p)
= (g^a mod(p))^b mod(p)
= g^(ab) mod(p)
= (g^b mod(p))^a mod(p)
K = B^a mod(p)

Attention : ce protocole permet d'obtenir un secret commun sans prérequis particulier. En revanche, il ne permet pas de garantir l'identité de l'interlocuteur avec qui on échange. N'importe quel « homme du milieu » peut répondre à la place de l'interlocuteur légitime et se faire passer pour lui pendant la phase de génération de la clé. Ce protocole doit donc généralement être associé à un processus d'authentification.

RSA

C'est un algorithme d'échanges de clés basé sur le problème mathématique du *petit théorème de Fermat*.



Partage de clés symétriques par Distribution

L'échange se fait de pair à pair.

On appelle cela le ***web of trust***. On utilise généralement un outil de chiffrement asymétrique pour chiffrer la clé symétrique ; comme PGP, puis on se la transmet par mail (par exemple).

Partage de clés symétriques par Infrastructure de gestion de clés

La gestion des clés est alors réalisée par une infrastructure de gestion de clés (ou PKI).

Fonctions à sens unique

Principe général

Les fonctions à sens unique permettent de transformer des données sans possibilité de revenir à l'antécédent à partir de l'image. Généralement, les données sources sont plus volumineuses que le résultat de la fonction ; qu'on appelle une empreinte.

En cryptographie, on **utilise** ces fonctions pour :

- Générer des signatures
- Garantir d'authenticité d'un message
- Dériver des clés
- Crypter des mots de passe

En cryptographie, une fonction à sens unique doit avoir les **propriétés** suivantes :

- Réduire au maximum les collisions
- Changer au maximum la sortie pour la moindre variation en entrée
- Etre facile à calculer
- En connaissant la fonction et le résultat, rendre très difficile la recherche de sources produisant le même résultat

L'attaque la plus classique consiste à trouver les collisions, c'est-à-dire l'ensemble des entrées provoquant le même résultat.

En effet, grâce à cette technique il devient possible de faire passer un message modifier pour authentique, et par exemples :

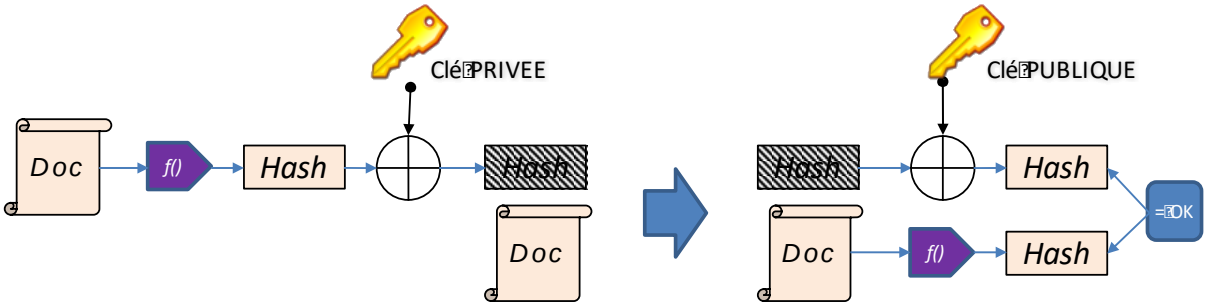
- Utiliser un faux mot de passe, qui fonctionne puisque c'est son hash que l'on compare avec celui du vrai mot de passe
- Usurper une identité en falsifiant un document administratif
- …

MAC/HMAC/CBC-MAC (signature à clé symétrique)

Ces fonctions permettent de générer un hash à partir d'un algorithme de chiffrement par blocs.

Signature numérique RSA/Elgamal/DSA

Pour signer un document, la méthode courante consiste à calculer le hash du document puis à le chiffrer avec la clé privée du signataire.

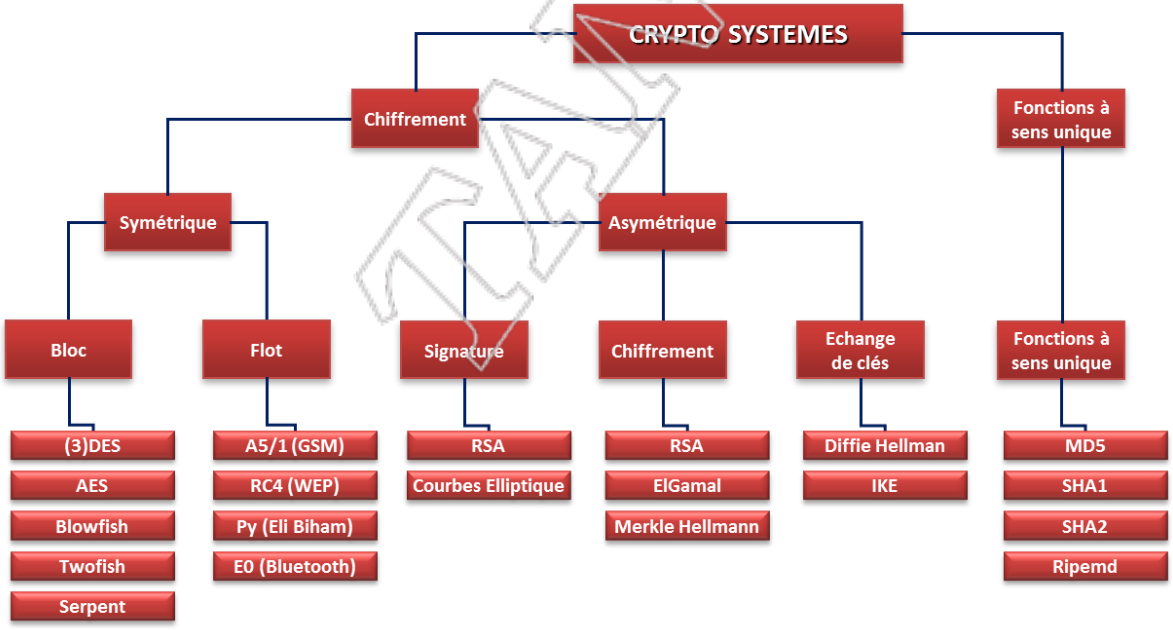


Les destinataires peuvent alors :

- Déchiffrer le hash avec la clé publique du signataire
- Le comparer avec le hash calculé du document reçu
- S'il est le même, c'est que le document est le même que celui émis.

<u>Attention :</u> la signature garantit 'une' intégrité, mais pas la provenance du document. L'original, s'il a été intercepté par un pirate, peut être signé par lui. Il faut donc garantir en plus que la personne qui utilise la clé privée est celle que l'on croit ; c'est le rôle des PKI (voir plus loin) !

Synoptique des cryptosystèmes



Protocoles de sécurité

Une suite cryptographique se compose d'un large choix d'implémentations possibles pour les fonctions suivantes :

Initialisation du contexte cryptographique :

Négociation de suite cryptographique

Accord mutuel entre le client et le serveur sur la meilleure suite cryptographique applicable

Echange de clés

DH (accord) ou RSA (chiffrement)

Authentification

Algorithme asymétrique (signature RSA ou DSA)

Application du contexte à la communication :

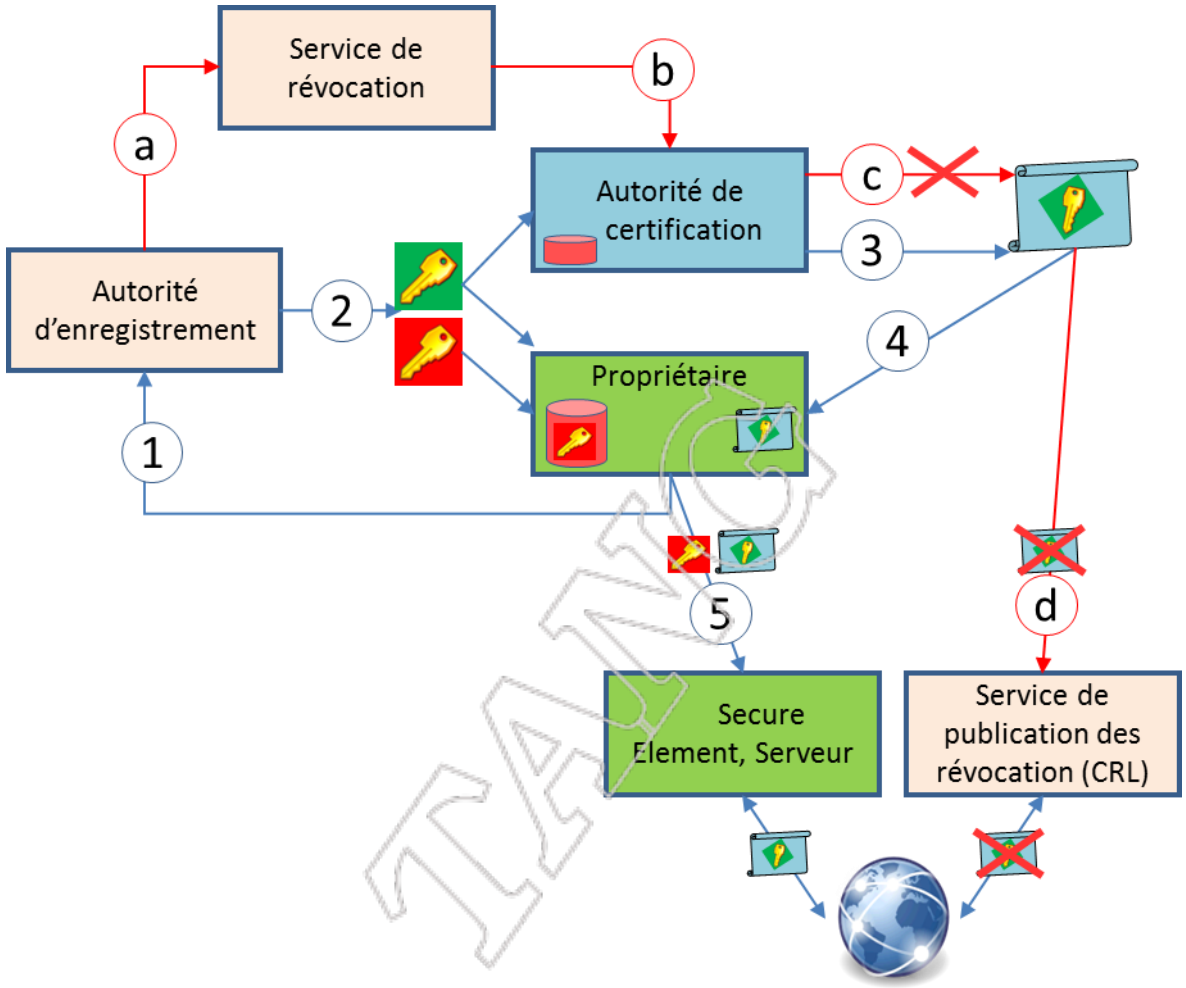
Protection des paquets

- chiffrement symétrique des paquets réseau (AES, 3DES)
- Intégrité des paquets réseau (HMAC-SHA1, CBC-MAC)

PKI et Certificats

La PKI

Schéma de fonctionnement



Description

L'**autorité d'enregistrement** effectue les vérifications d'identité sur le futur propriétaire du certificat.

L'**autorité de certification** peut s'appuyer sur ces vérifications pour signer en toute confiance le certificat contenant la clé publique du propriétaire, garantissant donc que la clé publique présentée lui appartient bien.

L'autorité de certification subit elle aussi le même parcours : sa clé publique est aussi validée, par une autorité supérieure.

Ainsi de suite, jusqu'à une **autorité Racine**. Le certificat de ces autorités racines est ancré dans le *magasin* des navigateurs Internet.

La **révocation** d'une clé publique (donc d'un certificat) est essentielle, pour pouvoir réagir en cas de compromission. Un processus similaire de révocation des certificats est donc prévu.

Remarques sur la révocation

Les éléments de biométrie constituent une clé non révocable. Qui peut changer son ADN ? Ses empreintes ? C'est aussi la clé que vous disséminez la plus autour de vous. Que penser des systèmes d'authentification à base d'empreintes qui fleurissent sur les PC et smartphones ?...

A lire :

- Débat lien fort – lien faible CNI Numérique.

- Fiabilité de la biométrie : erreurs sur les empreintes, erreurs sur les analyses ADN

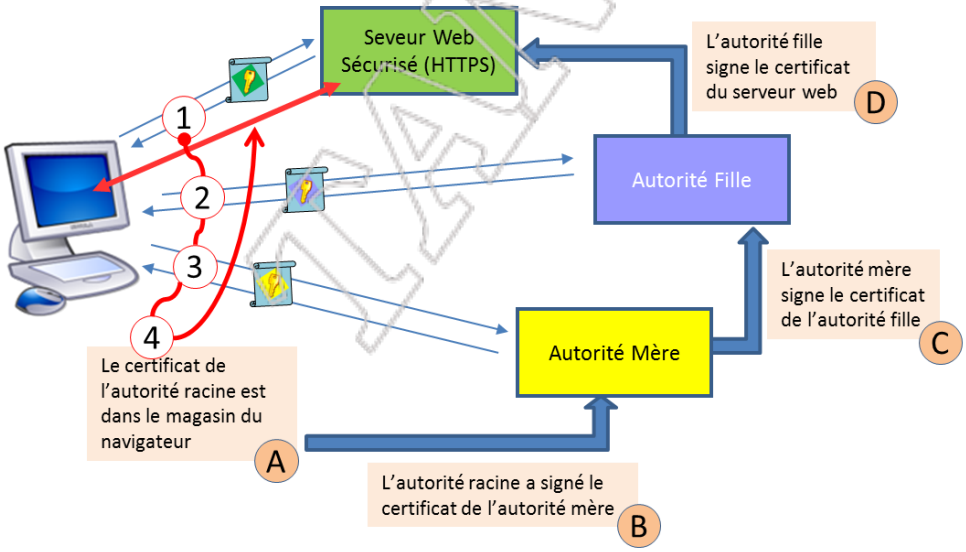
Les certificats

Principe

Les certificats sont utilisés pour récupérer la clé publique d'un interlocuteur en étant sûr qu'il s'agit bien de la sienne. Ils sont donc eux-aussi signés par une autorité supérieure, jusqu'à l'autorité racine, dont le certificat est introduit par les éditeurs de navigateurs dans leur magasin de certificats.

Dans l'exemple ci-dessous, on parle d'une authentification unilatérale (serveur web / HTTPS) ; mais on peut aussi faire de l'authentification réciproque.

Schéma de fonctionnement



Si on n'arrive pas jusqu'au 4 ; alors le navigateur lève une alerte pour proposer à l'utilisateur de mettre la plus haute autorité de la chaine de certificats dans le magasin du navigateur.

Attention : cela constitue un risque puisqu'ensuite le propriétaire de ce certificat et tous ceux qu'il approuve pour en émettre à son nom seront légitimes pour le navigateur.

Risques liés aux certificats

Ajout d'un faux certificat dans le magasin du navigateur.

Exemples :

- Un pirate peut signer un certificat de son faux site web pour faire du phishing,
- Une entreprise peut faire du man in the middle au niveau de ses proxys
…

Points d'attention :

➔ Un certificat valide permet à son propriétaire de signer toute une chaine de certificats qui seront aussi considérés valides : avoir un faux certificat dans le magasin de son navigateur est donc très grave, puisqu'on pourra vous faire croire qu'une signature (par exemple sur un site web) est légitime et le lien sécurisé.

➔ L'installation d'un nouveau certificat lève des alertes : soyez vigilants !

La technique la plus simple est le social engineering : envoi d'un lien vers un site qui propose un certificat auto-signé (ou dépendant d'une autorité auto-signée), dont on accepte l'installation dans le navigateur pensant le site légitime.

➔ Une entreprise ou un éditeur peuvent installer de faux certificats pour faire du *man in the middle* au niveau d'un proxy HTTPS. C'est la méthode la plus classique pour faire du DPI (Deep Packet Inspection).

Contenu d'un certificat

Les certificats contiennent beaucoup de champs. Voici les principaux :

Version : version du X509 utilisée par le certificat
Numéro de série : Numéro de série du certificat
Algorithme de signature: identifiant du type de signature
Emetteur : Distinguished Name (DN) de l'AC émettrice
Valide à partir de: date de début de validité de certificat
Valide jusqu'à : date de fin de validité de certificat
Objet: Distinguished Name (DN) du détenteur de la clef publique
Clé publique : infos sur la clef publique du certificat
Contraintes : extensions optionnelles
Utilisation : objet d'utilisation de la clé
Algorithme de signature des certificats : alogrithme de signature
Valeur : signature numérique de l'AC sur l'ensemble des champs précédents.

Présentation du RGS annexes B1 et B2

B1 Mécanismes cryptographiques

Cryptographie symétrique
- Taille de clés
- Chiffrements (par bloc et par flot)
- Authentification et intégrité de messages

Cryptographie asymétrique
- Problèmes mathématiques asymétriques
- Chiffrement asymétrique
- Signature asymétrique
- Authentification d'entité
- Etablissement de clés

Fonctions de hachage

Génération d'aléa cryptographique
- Architecture d'un générateur d'aléa
- Générateur physique d'aléa
- Retraitement algorithmique

Gestion de clés
- Clés secrètes symétriques
- Bi-clés asymétriques

B2 Gestion des clés cryptographiques

Gestion des clés cryptographiques
- Définition et concepts
- Objectifs de sécurité minimaux

Topologie des architectures de gestion de clés
- Cycle de vie des clés cryptographiques
- Architectures fonctionnelles des systèmes utilisateurs
- Exemples illustratifs

Règles et recommandations
- Règles et recommandations générales
- Demande de clés
- Génération de clés (locale, centralisée, clé de signature)
- Affectation d'une clé
- Introduction d'une clé (acheminement, injection)
- Utilisation d'une clé (diffusion, utilisation applicative)
- Fin de vie d'une clé
- Renouvellement d'une clé
- Recouvrement d'une clé