ROYAUME DU MAROC

مكتب التكوين المهني وإنعاش الشغل

Office de la Formation Professionnelle et de la Promotion du Travail

Cryptographie

www.ofppt.info

OFPPT

DIRECTION **R**ECHERCHE ET **I**NGENIERIE DE **F**ORMATION

SECTEUR NTIC

Sommaire

1. Introduction

Depuis l'Egypte ancienne, l'homme a voulu pouvoir échanger des informations de façon **confidentielle**.
Il existe de nombreux domaines où ce besoin est vital :

— **militaire** (sur un champ de bataille ou bien pour protéger l'accès à l'arme atomique) ;
— **commercial** (protection de secrets industriels) ;
— **bancaire** (protection des informations liées à une transaction financière) ;
— de la **vie privée** (protection des relations entre les personnes) ;
— diplomatique (le fameux « téléphone rouge » entre Etats-Unis et Union soviétique) ;

2. Qu'est-ce que la cryptographie?

Le mot cryptographie est un terme générique désignant l'ensemble des techniques permettant de chiffrer des messages, c'est-à-dire permettant de les rendre inintelligibles sans une action spécifique. Le verbe crypter est parfois utilisé mais on lui préfèrera le verbe chiffrer.

La cryptologie est essentiellement basée sur l'arithmétique : Il s'agit dans le cas d'un texte de transformer les lettres qui composent le message en une succession de chiffres (sous forme de bits dans le cas de l'informatique car le fonctionnement des ordinateurs est basé sur le binaire), puis ensuite de faire des calculs sur ces chiffres pour :

- d'une part les modifier de telle façon à les rendre incompréhensibles. Le résultat de cette modification (le message chiffré) est appelé **cryptogramme** (en anglais *ciphertext*) par opposition au message initial, appelé *message en **clair*** (en anglais *plaintext*) ;
- faire en sorte que le destinataire saura les déchiffrer.

Le fait de coder un message de telle façon à le rendre secret s'appelle chiffrement. La méthode inverse, consistant à retrouver le message original, est appelée déchiffrement.

Le chiffrement se fait généralement à l'aide d'une *clef de chiffrement*, le déchiffrement nécessite quant à lui une *clef de déchiffrement*. On distingue généralement deux types de clefs :

- *Les clés symétriques*: il s'agit de clés utilisées pour le chiffrement ainsi que pour le déchiffrement. On parle alors de **chiffrement symétrique** ou de chiffrement à clé secrète.
- *Les clés asymétriques*: il s'agit de clés utilisées dans le cas du **chiffrement asymétrique** (aussi appelé *chiffrement à clé publique*). Dans ce cas, une clé différente est utilisée pour le chiffrement et pour le déchiffrement

On appelle *décryptement* (le terme de *décryptage* peut éventuellement être utilisé également) le fait d'essayer de *déchiffrer illégitimement* le message (que la clé de déchiffrement soit connue ou non de l'*attaquant*).

Lorsque la clef de déchiffrement n'est pas connue de l'attaquant on parle alors de **cryptanalyse** ou **cryptoanalyse** (on entend souvent aussi le terme plus familier de *cassage*).
La **cryptologie** est la science qui étudie les aspects scientifiques de ces techniques, c'est-à-dire qu'elle englobe la cryptographie et la cryptanalyse.

La cryptographie est traditionnellement utilisée pour dissimuler des messages aux yeux de certains utilisateurs. Cette utilisation a aujourd'hui un intérêt d'autant plus grand que les communications via internet circulent dans des infrastructures dont on ne peut garantir la fiabilité et la confidentialité. Désormais, la cryptographie sert non seulement à préserver la confidentialité des données mais aussi à garantir leur intégrité et leur authenticité

On appelle **cryptanalyse** la reconstruction d'un message chiffré en clair à l'aide de méthodes mathématiques. Ainsi, tout cryptosystème doit nécessairement être résistant aux méthodes de cryptanalyse. Lorsqu'une méthode de cryptanalyse permet de déchiffrer un message chiffré à l'aide d'un cryptosystème, on dit alors que l'algorithme de chiffrement a été « cassé ».
On distingue habituellement quatre méthodes de cryptanalyse :

- Une **attaque sur texte chiffré seulement** consiste à retrouver la clé de déchiffrement à partir d'un ou plusieurs textes chiffrés ;
- Une **attaque sur texte clair connu** consiste à retrouver la clé de déchiffrement à partir d'un ou plusieurs textes chiffrés, connaissant le texte en clair correspondant ;
- Une **attaque sur texte clair choisi** consiste à retrouver la clé de déchiffrement à partir d'un ou plusieurs textes chiffrés, l'attaquant ayant la possibilité de les générer à partir de textes en clair ;
- Une **attaque sur texte chiffré choisi** consiste à retrouver la clé de déchiffrement à partir d'un ou plusieurs textes chiffrés, l'attaquant ayant la possibilité de les générer à partir de textes en clair.

3. La notion de codage de l'information

Historiquement, l'utilisation d'alphabet a permis de coder chaque mot du langage à partir de mêmes symboles à la différence des idéogrammes chinois par exemple.

L'ajout d'un ordre sur ces lettres à permis de définir les premières méthodes «*mathématiques* » de chiffrement d'un message constitué de lettres (code César, ROT13…).

Ces chiffrements partent d'un message contenant des lettres vers un cryptogramme contenant également des lettres.

Ces méthodes se décomposent en deux grandes familles de chiffrement :

— **Par Substitution**
— **par transposition**.

D'autres formes de chiffrement ?
Il existe également d'autres formes comme le code morse ou bien les sémaphores dans la Marine. Ce sont des techniques de brouillage.

4. Chiffrement par substitution

Cette méthode correspond à substituer un caractère ou un groupe de caractères par un autre dans le texte à chiffrer.

Plusieurs types de **cryptosystèmes par substitution :**

— **monoalphabétique** (code César) consiste à remplacer chaque lettre du message par une autre lettre de l'alphabet ;

– **homophonique** permet de faire correspondre à chaque lettre du message en clair un ensemble possible d'autres caractères *c'est un peu similaire aux méthodes employées par les mordus de SMS* ;

– **polyalphabétique** (code Vigenère) consiste à utiliser une suite de chiffrement, monoalphabétique réutilisée périodiquement ;

— **polygrammes** consiste à substituer un groupe de caractères (polygramme) dans le message par un autre groupe de caractères.

4.1. Exemples : Chiffrement par substitution mono alphabétique

Le chiffrement de César

Ce code de chiffrement est un des plus anciens, dans la mesure où Jules César l'aurait utilisé. Le principe de codage repose sur l'ajout d'une valeur constante à l'ensemble des caractères du message, ou plus exactement à leur code ASCII (pour une version "informatique" de ce codage).

Il s'agit donc simplement de décaler l'ensemble des valeurs des caractères du message d'un certain nombre de positions, c'est-à-dire en quelque sorte de substituer chaque lettre par une autre. Par exemple, en décalant le message " **WNT** " de 3 positions, on obtient "*VMS*". Lorsque l'ajout de la valeur donne une lettre dépassant la lettre Z, il suffit de continuer en partant de A, ce qui revient à effectuer un *modulo 26*.

A titre d'exemple, dans le film *L'odyssée de l'espace*, l'ordinateur porte le nom de *HAL*. Ce surnom est en fait *IBM* décalé de 1 position vers le bas...

On appelle clé le caractère correspondant à la valeur que l'on ajoute au message pour effectuer le cryptage. Dans notre cas la clé est C, car c'est la 3ème lettre de l'alphabet.

Ce système de cryptage est certes simple à mettre en oeuvre, mais il a pour inconvénient d'être totalement symétrique, cela signifie qu'il suffit de faire une soustraction pour connaître le message initial. Une méthode primaire peut consister à une bête soustraction des nombres 1 à 26 pour voir si l'un de ces nombres donne un message compréhensible.

Une méthode plus évoluée consiste à calculer les fréquences d'apparition des lettres dans le message codé (cela est d'autant plus facile à faire que le message est long). Effectivement, selon la langue, certaines lettres reviennent plus couramment que d'autres (en français, par exemple, la lettre la plus utilisée est la lettre E), ainsi la lettre apparaissant le plus souvent dans un texte crypté par le chiffrage de César correspondra vraisemblablement à la lettre E, une simple soustraction donne alors la clé de cryptage...

Un autre exemple : le ROT13

Dans le cas spécifique du chiffrement de Jules César où la clé de cryptage est N (13ème lettre de l'alphabet), on appelle ce cryptage ROT13 (le nombre 13, la moitié de 26, a été choisi pour pouvoir chiffrer et déchiffrer facilement les messages textuels).

Le ROT13 (rotation de 13) est un code César qui permet quand on l'applique deux fois de retrouver le message original.

Il est souvent employé sur USENET (les news) pour masquer la solution d'une devinette ou pour parler aux initiés. *Les lecteurs de news l'intègrent en général*

4.2. Cryptanalyse du chiffrement par substitution

4.2.1. Cryptanalyse du chiffrement par substitution

Dans le cas de l'utilisation d'un code par substitution, la cryptanalyse ou déchiffrement se fait par l'utilisation de données **statistiques** :
En anglais, les caractères les plus fréquemment utilisés sont : e, t, o, a, n, i…
Les combinaisons de deux lettres (digrammes) les plus fréquentes sont : th, in, er, re, et an. Les combinaisons de trois lettres (trigrammes) : the, ing, and et ion.

4.2.2. Méthode empirique de cryptanalyse

Il suffit pour retrouver le texte en clair de :
— Rechercher les **caractères**, **digrammes** et **trigrammes** les plus fréquents du texte chiffré;
— Faire des **suppositions** en les associants à ceux les plus fréquents d'un texte en clair (dans la langue choisi).
Par exemple dans un texte crypté appartenant à une banque il est probable de trouver des mots tel que financier, montant, solde…

4.2.3. Comment finir la cryptanalyse ?

Si certains mots commencent à émerger du texte chiffré, alors il y a de **fortes probabilités** que le code de chiffrement soit découvert.

Un code par substitution **ne modifie pas** les **propriétés statistiques** des caractères, digrammes et trigrammes substitués.

Il conserve **l'ordre des caractères** du texte en clair, mais masque ces caractères.

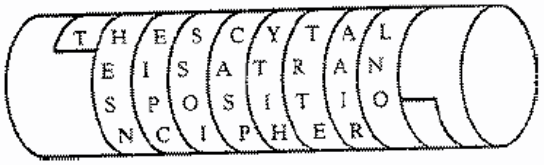
Table des fréquences d'apparition des lettres pour un texte français

Lettre	Fréquence %	Lettre	Fréquence %
A	9.42	N	7.15
B	1.02	O	5.14
C	2.64	P	2.86
D	3.39	Q	1.06
E	15.87	R	6.46
F	0.95	S	7.90
G	1.04	T	7.26
H	0.77	U	6.24
I	8.41	V	2.15
J	0.89	W	0.00
K	0.00	X	0.30
L	5.34	Y	0.24
M	3.24	Z	0.32

5. Chiffrement par transposition

Les méthodes de chiffrement par transposition consistent à réarranger les données à chiffrer de façon à les rendre incompréhensibles. Il s'agit par exemple de réordonner géométriquement les données pour les rendre visuellement inexploitables
Toutes les lettres du message sont présentes, mais dans un ordre différent. C'est un chiffrement de type *anagramme.* Il utilise le principe mathématique des **permutations** (par colonne par exemple)

La *scytale* spartiate (5ème siècle av. JC) :



LA TRANSPOSTION PERMET EN THEORIE D'AVOIR UN HAUT DEGRE DE SECURITE

L	R	S	S	I	P	M	E	H	R	D	O	U	A	D	R	E	C	I
A	A	P	I	O	E	E	N	E	I	A	I	N	U	E	E	S	U	T
T	N	O	T	N	R	T	T	O	E	V	R	H	T	G	D	E	R	E

LRSSIPMEHRDOUADRECIAAPIOEENEIAINUEESUTTNOTNRTTOEVRHTGDERE

Les méthodes de chiffrement par transposition consistent à réarranger les données à chiffrer de telle façon à les rendre incompréhensibles.
En général : réarranger géométriquement les données pour les rendre visuellement inexploitables.
Par exemple : "Ceci est un texte à chiffrer de la plus haute importance"
Ceci est un texte à chiffrer de la plus haute importance
Le texte est regroupé en tableau, suivant un nombre de colonnes donné.

Ceci est u
n texte à
chiffrer d Cncehre h atctiluaiefatn... Chaque colonne est ensuite copiée l'une après l'autre.
e la plus
haute impo
rtance

5.1. Cryptanalyse du chiffrement par tranposition

5.1.1. Cryptanalyse

— Déterminer si une substitution n'a pas été utilisée : une analyse statistique des caractères suffit à déterminer si les caractères ont été substitués (statistiques fréquentielles du texte identiques à celle d'un texte en clair).

— Si ce n'est pas le cas, il y a une forte probabilité pour qu'un chiffrement par transposition ait été employé.

— Ensuite, il faut faire une hypothèse sur le nombre de colonnes utilisées pour réaliser la transposition.

Les codes de transposition contrairement aux codes par substitution ne cachent pas les caractères, mais modifient l'ordre des caractères.

Histoire :
L'arrivée des ordinateurs a totalement démodé ces méthodes de chiffrement (on ne parle plus d'ailleurs de chiffrement car ces méthodes ne résiste pas au traitement informatique). La machine **Enigma** utilisée par les nazis a été « cassée » par Alan Turing, pionnier de l'informatique.
Il faut attendre les annés 60 pour voir les méthodes de chiffrement moderne basées sur l'usage de clés.

6. Comment renforcer la force des chiffrements ?

Combiner Substitution et Transposition

il est possible de faire subir aux caractères du « texte en clair » :

— une substitution ;
— plusieurs opérations de transposition.

Changer les paramètres de ces combinaisons très souvent

l'utilisation des paramètres de chaque opération doit être réduite au chiffrement de quelques messages avant d'être changés pour de nouveaux paramètres.

Combiner les paramètres

Les opérations sont connues, la séquence d'application des opérations est définie par la séquence des paramètres de chaque opération.
La combinaison des différents paramètres des différentes opérations permet de définir un secret.

Ce secret permet de réaliser le déchiffement et assure la sécurité du cryptogramme. Il est appelé clé de chiffrement.

Le but

rendre l'apparence du cryptogramme la plus « aléatoire » possible, c-à-d. éliminer les relations statistiques des caractères du cryptogramme pour éviter la cryptanalyse :

Transposition + Substitution = Diffusion

L'actualité ?

les chiffrements tels que **DES** (*Data Encryption System*) et **AES** (*Advanced Encryption System*) sont utilisés à l'heure actuelle.

7. Cryptographie moderne - Le cryptage à clé

7.1. Cryptographie moderne

Ce type de chiffrement repose sur l'utilisation :

— d'un algorithme public, connu de tous
— d'une clé.

Il correspond à la cryptographie moderne, par rapport aux codes par substitution et transposition. Auparavant, les algorithmes étaient simples mais utilisaient des clés longues.

Exemple : un XOR entre le message à transmettre et une clé de même taille suffit à le rendre indéchiffrable…technique du masque jetable

Maintenant, le but est d'utiliser des algorithmes sophistiqués et complexes associés à des clés courtes. Ces algorithmes représente des investissements à long terme, c-à-d. qu'ils sont employés pendant de nombreuses années jusqu'à ce qu'ils en puissent plus assurer le même niveau de sécurité.

Il existe deux sortes de cryptage :

— **à clé symétrique**
— **à clé asymétrique.**

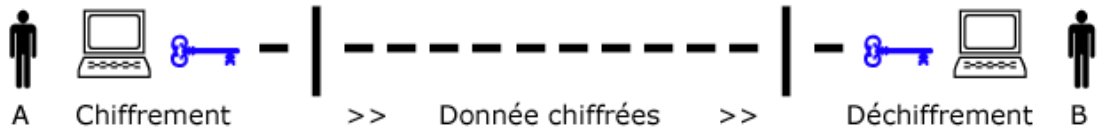
Hypothèse de base de la cryptanalyse :

Principe de Kerckhoff -- Auguste Kerckhoff, ``La cryptographie militaire'', février 1883 ; L'opposant connaît le système cryptographique et Toute la sécurité d'un système cryptographique doit reposer sur la clé, et pas sur le système lui-même

7.2. Chiffrement à clé symétrique

7.2.1. Principe

Le cryptage à clé symétrique (ou secrète) : La **même clé** doit être employée pour chiffrer ou déchiffrer le message;



Le chiffrement consiste alors à effectuer une opération entre la clé privée et les données à chiffrer. Le déchiffrement se fait à l'aide de cette **même clé secrète**.

Remarques

La qualité d'un crypto système symétrique se mesure par rapport :

— à des propriétés statistiques des textes chiffrés ;
— à la résistance aux classes **d'attaques connues**.

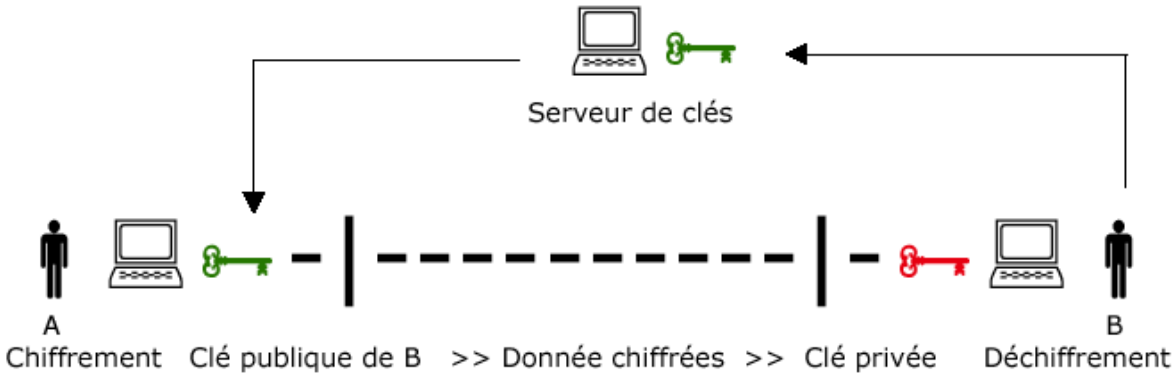
En pratique : tant qu'un crypto système symétrique n'a pas été cassé, il est bon, après il est mauvais !

7.3. Chiffrement à clé asymétrique

7.3.1. Principe

Il utilise :

— une **clé publique** connue de tous ;
— une **clé privée** connue seulement du destinataire du cryptogramme.

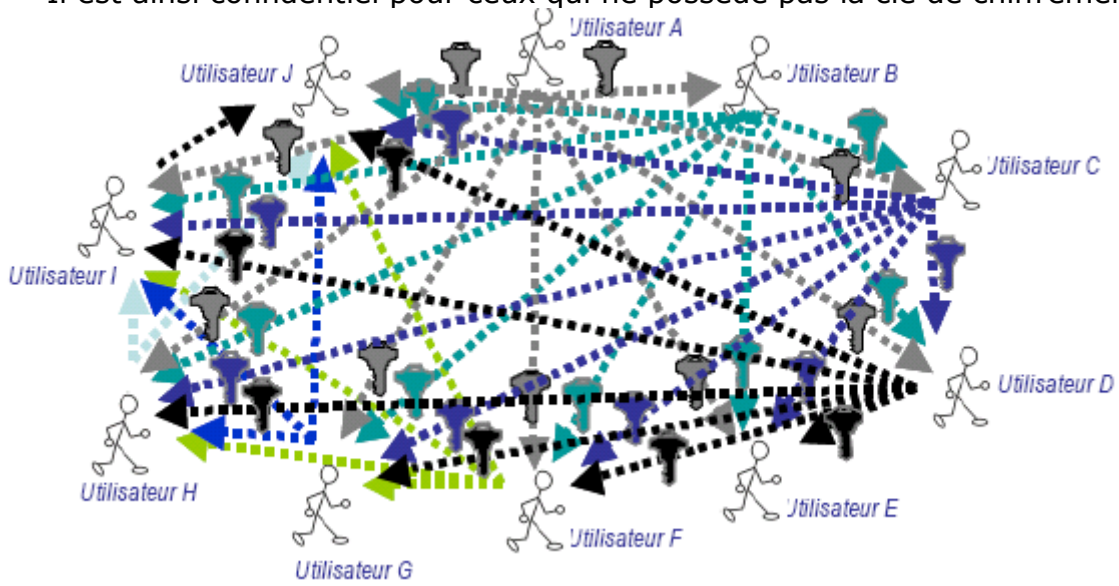


Ces chiffrements a « clé publique» ont été découvert par James Ellis (Angleterre) en 1969 et par Whitfield Diffie (Etats unis) en 1975.
L'idée de la conception de tels algorithmes revient à Diffie et Hellman en 1976.

7.4. Les limites de la cryptographie Symétrique

La multiplication des clés

Pour établir un canal de communication entre deux individus :
— Il faut qu'il soit chiffré avec une clé partagée entre les deux individus ;
— Il est ainsi confidentiel pour ceux qui ne possède pas la clé de chiffrement.



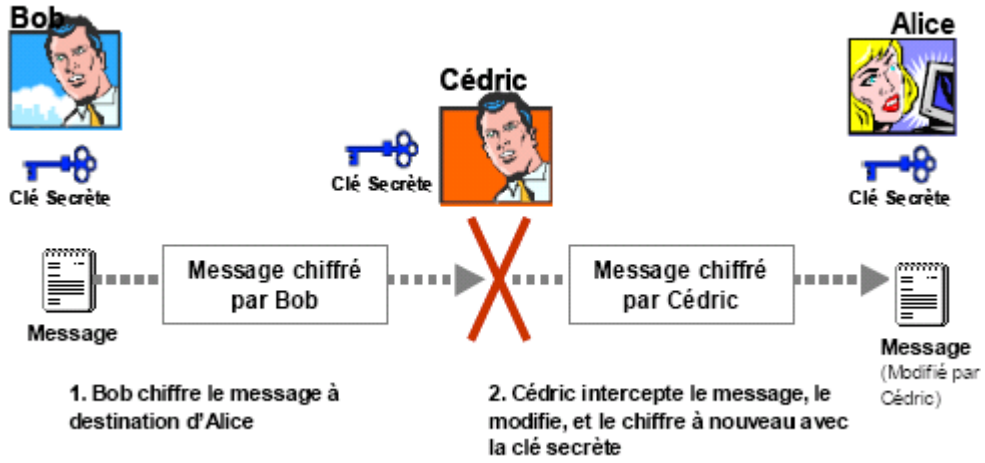
Pour que deux canaux de communications soient indépendants l'un de l'autre, c-à-d. qu'une personne accède à l'un mais pas à l'autre, il faut que ces deux canaux utilisent des clés différentes.

Il est possible qu'un des interlocuteurs connaissent plusieurs clés utilisés dans différents canaux le reliant à des utilisateurs différents.

Exemple : l'utilisateur D possède une clé pour chaque lien (avec J, I, H, G, F et E).
Problème : comment échanger toutes ces clés ?

Pas d'intégrité et d'identification de l'auteur

Si Alice, Bob et Cédric partage le même lien de communication alors ils partagent la même clé de chiffrement symétrique.



Chacun peut intercepter et modifer les messages qui s'échangent

7.5. Chiffrement asymétrique

Construction des clés

Les utilisateurs (A et B) choisissent une clé aléatoire dont ils sont seuls connaisseurs (il s'agit de la clé privée).

A partir de cette clé, ils déduisent chacun automatiquement par un algorithme la clé publique.

Les utilisateurs s'échangent cette clé publique au travers d'un canal non sécurisé.

Chiffrement d'un message

Lorsqu'un utilisateur désire envoyer un message à un autre utilisateur, il lui suffit de chiffrer le message à envoyer au moyen de la clé publique du destinataire (qu'il trouvera par exemple dans un serveur de clés tel qu'un annuaire ou bien en signature d'un courrier électroique).

Le destinataire sera en mesure de déchiffrer le message à l'aide de sa clé privée (qu'il est seul à connaître).

Rapports entre les clés

La recherche de la clé privée à partir de la clé publique revient à résoudre un problème mathématique notoirement très compliqué, c-à-d. demandant un grand nombre d'opérations et beaucoup de mémoire pour effectuer les calculs -> infaisable !
Par exemple dans RSA, l'algorithme le plus utilisé actuellement, la déduction de la clé privée à partir de la clé publique revient à résoudre un problème de factorisation de grand nombre que lequel travaille les
mathématiciens depuis plus de 2000 ans !

Le choix des clés doit être fait de la manière la plus imprédictible possible : éviter les mots du dictionnaire, nombres pseudo-aléatoires à germe de génération difficile à deviner, etc.

7.6. Prise en en compte de la notion d'échange par réseau

Echange par réseau

L'objectif de la cryptographie est de permettre à deux personnes, **Alice** et **Bob**, de communiquer au travers d'un canal peu sûr (téléphone, réseau informatique ou autre), sans qu'un opposant,**Oscar**, puisse comprendre ce qui est échangé.

Alice souhaite transmettre à **Bob** un ensemble de données (texte, nombres, …).

Alice transforme ces informations par un procédé de chiffrement en utilisant une clé prédéterminée, puis envoie le texte chiffré au travers du canal de communication.
Oscar, qui espionne peut-être le canal, ne peut reconstituer l'information, contrairement à **Bob** qui dispose de la clé pour déchiffrer le cryptogramme

7.7. Une approche théorique

7.7.1. Cryptage à clé symétrique

Ce cryptage repose sur la définition d'une formule mathématique de la forme :

Donnée chiffrées = Fonction (données, clé)

Avec une fonction inverse de la forme :

Données = Fonction_inverse (données_chiffrées, clé)

Dans cette méthode de chiffrement, on distingue deux types d'algorithmes :

— l'algorithme **par bloc** qui prend une longueur spécifiée de données comme entrée, et produit une longueur différente de données chiffrées (exemple : DES, AES…)

— l'algorithme en **flux continu** qui chiffre les données un bit à la fois (exemple : IDEA, CAST, RC4, SKIPjack…).

Avantages et inconvénients d'un cryptosystème à clé symétrique

Le principal inconvénient d'un cryptosystème à clefs secrètes provient de **l'échange** des **clés**.
Le chiffrement symétrique repose sur l'échange d'un secret (les clés).
Pour être totalement sûr : les chiffrements à clés secrètes doivent utiliser des clés d'une longueur au moins égale à celle du message à chiffrer (One Time Pad ou « Masque Jetable »)

En pratique : les clés ont une taille donnée, suffisante.

Lors d'échange entre plusieurs intervenants : une clé est partagée que par 2 interlocuteurs, donc pour N interlocuteurs il faut N*(N-1)/2 clés.

La plupart des codes utilisés

— sont **relativements rapides** ;
— peuvent s'appliquer à un **fort débit** de donnée à transmettre.

Il existe des processeurs spécialement conçu pour réaliser le chiffrement et le déchiffrement.
Principaux algorithmes utilisés :

— DES, *Data Encryption System* IBM 1977 ;
— IDEA, *International Data Encryption Algorithm* Lai et Massey 1990 ;
— Blowfish, Schneir 1994.

Problème d'assurer la sécurité des clés.
Problème de la **distribution des clés**, qui doit se faire par un canal qui doit être sûr. *La valise diplomatique dans le cas du téléphone rouge…*

7.8. Chiffrement asymétrique

Cryptage à clé asymétrique

Il repose sur la connaissance d'une fonction mathématique unidirectionnelle ("*one-way function*"), munie d'une porte arrière ("*one-way trapdoor function*").

Une fonction unidirectionnelle est une fonction y = f(x) telle que, si l'on connaît la valeur y, il est pratiquement impossible de calculer la valeur x (c'est-à-dire d'inverser la fonction f). On dit que cette fonction est munie d'une porte arrière s'il existe une fonction x = g(y, z) telle que, si
l'on connaît z, il est facile de calculer x à partir de y. Z est appelée trappe.

Exemple de scénario d'échange

Bob veux recevoir des messages codés **d'Alice**, il souhaite que ces messages soient indéchiffrables pour **Oscar** qui a accès à leurs échanges :
— **Bob** et **Alice** connaissent la fonction unidirectionnelle f ;
— **Bob** fournit à **Alice** sa "clé publique" c.
— f et c peuvent être connus de tout le monde : ils sont connus **d'Oscar**.
Alice chiffre le message M en utilisant l'algorithme f et la clé c ; ceci fournit un texte T chiffré ayant les apparences d'une suite de caractères choisis au hasard :
T = f(M, c).

Comme f est une fonction unidirectionnelle, Oscar est incapable de reconstituer le message même si il connaît l'algorithme f, la clé publique c et le texte T.

Bob, lui, possède la « clé privée » z qui est absolument secrète.
z ouvre la porte arrière de la fonction f et permet de déchiffrer le message en appliquant la fonction g au triplet (T, c, z) : M = g(T, c, z).
Bob peut lire le contenu du message envoyé par Alice !

Des clé et des cadenas

Alice :
— crée une clé aléatoire (la clé privée) ;
— puis fabrique un grand nombre de cadenas (clé publique) qu'elle met à disposition dans un casier accessible par tous (le casier joue le rôle de canal non sécurisé).

Bob :
— prend un cadenas (ouvert) ;
— ferme une valisette contenant le document qu'il souhaite envoyer ;
— envoie la valisette à Alice, propriétaire de la clé publique (le cadenas).
Cette dernière pourra ouvrir la valisette avec sa clé privée

Les contraintes pour un tel algorithme

Il faut trouver un couple de fonctions f (fonction unidirectionnelle) et g (fonction de porte arrière) : *C'est un problème mathématique difficile !*

Au départ, le système à clé publique n'a d'abord été qu'une idée dont la faisabilité restait à démontrer.
Des algorithmes ont été proposés par des mathématiciens .Un des premiers algorithmes proposé repose sur la factorisation du produit de deux grands nombres entiers. Cette factorisation demanderait un temps de calcul de plusieurs millions d'années.

Le problème est résolu !

Cet algorithme a été proposé par Rivest, Shamir et Adleman en 1977, ce qui a donné naissance à RSA. L'idée générale est la suivante :

— la clé publique c est le produit de deux grands nombres entiers;
— la clé privée z est l'un de ces deux nombres entiers;
— g comporte la factorisation de c.

Seul Bob, qui connaît z, peut factoriser c et donc déchiffrer le message chiffré.

Un dernier problème

Le système de chiffrement à clé publique est **universel** si chacun publie sa clé publique dans un **annuaire**.
Pour envoyer un message chiffré à Bob, il suffit de trouver sa clé publique dans l'annuaire et de s'en servir pour chiffrer le message avant de le lui envoyer (seul Bob pourra déchiffrer le message). Il faut bien sûr que **l'annuaire** soit **sûr**.

Oscar peut avoir substitué sa propre clé publique à celle de Bob afin de pouvoir lire les messages destinés à Bob. Il peut même les renvoyer à Bob une fois lu !

7.9. Quelques éléments de réflexion

La notion d'inverse

Ce que fait l'algorithme de chiffrement devra être défait plus tard lors du déchiffrement. En mathématique, l'idée de défaire est **l'inverse**.

Il existe des **fonctions inverses** et des **nombres inverses**.

Les fonctions inverses sont des paires d 'opérations : exemple la multiplication et la division sont des fonctions inverses, ce que l'une fait, l'autre le défait.

Exemple : 5 * 2 = 10, 10 / 2 = 5

Les nombres inverses sont des paires de nombres, ce qu'un nombre fait, l'autre le défait.

Exemple : 2 et ½ avec 5 * 2 = 10, et 10 * ½ = 5

Avec les nombres inverses, l'opération reste la même (ici, la multiplication).

La notion de nombre premier

Un nombre premier est simplement un nombre qui ne possède que deux facteurs, 1 et lui-même.
7 est premier car aucun nombre autre que 1 et 7 ne donne un résultat entier en divisant 7. Deux nombres sont premiers entre eux s'ils n'ont pas d'autre facteur que 1.
38 et 55 sont premiers entre eux, alors qu'aucun n'est premier : 38 = 2 * 19 *1 et 55 = 5 * 11 * 1
22 et 55 ne sont pas premiers entre eux, car 22 = 2 * 11 et 55 = 5 * 11

7.10. Idée de chiffrement à clé publique : le RSA

Euler modifié

On sait que $m^{(p-1)(q-1)+1} \bmod n = m$
Il est possible d'aller de m vers m par (p − 1)(q − 1) + 1, il ne suffit plus que de décomposer cette valeur
en deux sous valeurs :
— l'une permettant de passer de m à une valeur intermédiaire ;
— l'autre permettant de passer de cette valeur intermédaire vers m ;
Possibilité de chiffrement à clé publique !
e * d = (p − 1)(q − 1) + 1
Exemple : e * d = 41...mais 41 est premier !
Comment faire ?

utiliser l'arithmétique modulaire : trouver e * d tel que e * d = 1 mod { e * d – 1}
Principe de RSA
utiliser deux modules, l'un pour les clès et l'autre pour chiffrer.
pour les clés : (p – 1) (q – 1)
pour chiffrer p * q

8. Chiffrement asymétrique : présentation de RSA

Un algorithme simple

Soient :
— M le message en clair
— C le message encrypté
— (e,n) constitue la clé publique
— (d,n) constitue la clé privée
— n le produit de 2 nombres premiers
— ^ l'opération de mise à la puissance (a^b : a puissance b)
— mod l'opération de modulo (*reste de la division entière*)
Pour chiffrer un message M, on fait: C = M^e mod n
Pour déchiffrer: M = C^d mod n

Construction des clés

Pour créer une paire de clés, c'est très simple, mais il ne faut pas choisir n'importe comment e,d et n.
Le calcul de ces trois nombres est délicat.
— prendre deux nombres premiers p et q (de taille à peu près égale). Calculer n = pq.
— prendre un nombre e qui n'a aucun facteur en commun avec (p-1)(q-1).
— calculer d tel que e * d mod (p-1)(q-1) = 1
Le couple (e,n) constitue la clé publique. (d,n) est la clé privée.
La puissance du cryptage RSA est en effet basée sur la difficulté de factoriser un grand entier. C'est pour cela que l'on choisit des nombres premiers p et q d'environ 100 chiffres, pour rendre la factorisation hors de portée, même des meilleurs ordinateurs.

8.1.1. Exemple d'utilisation de RSA

Création de la paire de clés:

Soient deux nombres premiers au hasard: p = 29, q = 37, on calcule n = pq = 29 * 37 = 1073.
On doit choisir e au hasard tel que e n'ai aucun facteur en commun avec (p-1)(q-1):
(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 = 1079.
On a maintenant les clés :
— la clé publique est (e,n) = (71,1073) (=*clé de chiffrement*)
— la clé privée est (d,n) = (1079,1073) (=*clé de déchiffrement*)

Chiffrement du message 'HELLO'.

On prend le code ASCII de chaque caractère et on les met bout à bout:
m = 7269767679
Il faut découper le message en blocs qui comportent moins de chiffres que n.
n comporte 4 chiffres, on découpe notre message en blocs de 3 chiffres:
726 976 767 900 (on complète avec des zéros)
On chiffre chacun de ces blocs :
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 déchiffrer avec d:

436^1079 mod 1073 = 726
822^1079 mod 1073 = 976
825^1079 mod 1073 = 767
552^1079 mod 1073 = 900
C'est à dire la suite de chiffre 726976767900.
On retrouve notre message en clair 72 69 76 76 79 : 'HELLO' !

Propriété unique

L'algorithme a la propriété spéciale suivante (*utilisé* pour l'authentification):
chiffrement (déchiffrement (M)) = déchiffrement (chiffrement (M))
C'est-à-dire que l'utilisation de sa clé privée pour chiffrer un message M permet de construire un message M' qui peut être déchiffré par sa clé publique...ainsi il est possible de prouver que l'on dispose bien de la clé privée qui correspond à la clé publique !

Sécurité

La force du chiffrement dépend de la longueur de la clé utilisée.
Ce protocole a l'avantage d'utiliser des clés de longueur variable de 40 à 2 048 bits ;
Il faut actuellement utiliser une clé au minimum de 512 bits (Six laboratoires ont dû unir leurs moyens
pour casser en août 1999 une clé à 512 bits)

9. Le cryptage à clé symétrique - le DES

Un standard de chiffrement

Développé dans les années 70 par IBM, la méthode DES fut adoptée et rendue standard par le gouvernement des Etats Unis. Il devait répondre à l'époque aux critères suivants :

— avoir 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.

La méthode DES utilise des clés d'une taille de 56 bits ce qui la rend de nos jours facile à casser avec les nouvelles technologies de cryptanalyse. Mais elle est toujours utilisée pour des petites tâches tel que l'échange de clés de cryptage (technologie SSL).

La clé est sur 64bits dont 8 sont utilisés comme calcul de l'intégrité des 56 autres (parité).

Le DES est un standard utilisé depuis plus de 20 ans. Il a suscité de nombreuses critiques, des suspicions de vulnérabilité à l'attaque de son algorithme, mais n'a pas eu d'alternatives jusqu'à ces dernières années : modifié par la NSA, trafiqué par IBM, …

Principe de l'algorithme

C'est un algorithme à base de :
— décalage ;
— « ou exclusif » ;
— transposition/recopie (appelé expansion).
Ces opérations sont faciles à réaliser par un processeur.

Le chiffrement par DES est très rapide.

Certaines puces spécialisées chiffrent jusqu'à 1 Go de données par seconde ce qui est énorme : c'est plus que ce qu'est capable de lire un disque dur normal.

Principe de fonctionnement

L'algorithme utilise une clé de 56 bits. Décomposition du texte en clair en bloc

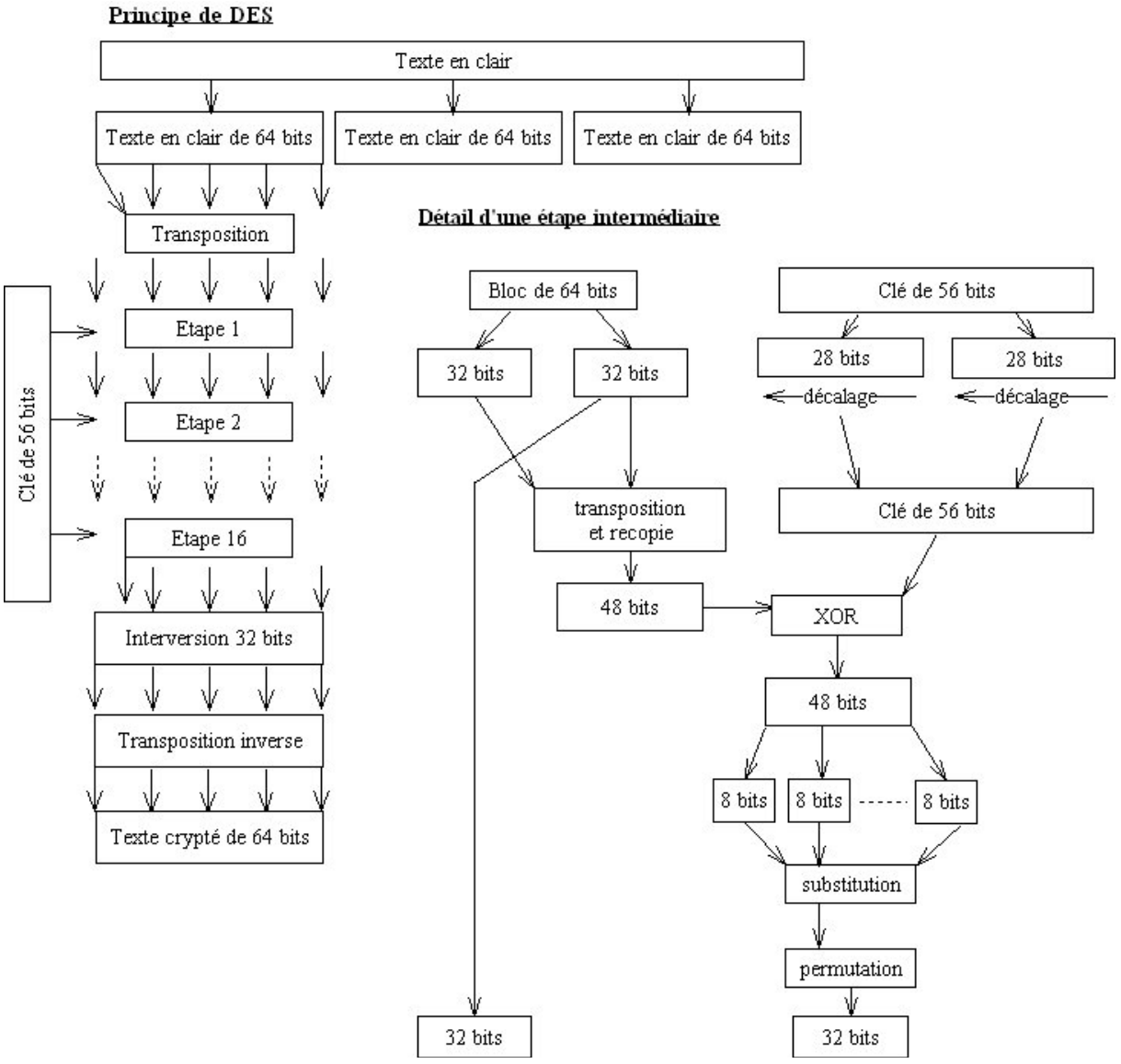
— le texte en clair est découpé en bloc de 64 bits qui seront chiffrés un par un ;
Utilisation en différentes étapes, éventuellement répétées (en tout 19 étapes) :
— la première étape **transpose** chaques blocs de 64 bits du texte en clair avec la clé de 56 bits ;
— 16 **étapes intermédiaires** ;
— l'avant dernière étape **intervertit** les 32 bits de droite et de gauche ;

— la dernière étape **transpose** chaques blocs de 64 bits du texte avec la clé de 56 bits (exactement à
l'inverse de la première étape).
Les 16 étapes intermédiaires sont identiques mais varient par différentes utilisations de la clé

Une étape intermédiaire

Elle consiste à couper le bloc de 64 bits en 2 blocs de 32 bits.
Le bloc de sortie de gauche sera une recopie du bloc de droite en entrée.
Le bloc de droite est utilisé pour calculer un nombre de 48 bits à l'aide de **règles** de **transposition** et de **recopie**. Ces règle sont **stockées** dans des **tables** et leur construction reste mystérieuse…le NSA y a participé !
La clé de 56 bits est divisée en 2 blocs de 28 bits, sur ces blocs de 28 bits un **décalage circulaire** est effectué vers la gauche d'un **nombre de position** dépendant de l'itération.

Un « ou exclusif » est calculé entre le nombre de 48 bits et la clé de 56 bits. Le résultat de ces « ou exclusifs » est découpé en blocs de 6 bits.



9.1.1. La cryptanalyse ?

Brute force : essayer toutes les clés possibles !
Le **nombre** de **clés** est **élevé** (2^56=7,2*10 16) et peut être facilement augmenté en changeant le nombre de bits pris en compte (soit exactement 72.057.595.037.927.936 clés différentes !).
Exemple : si une personne peut tester 1 million de clés par seconde
il lui faut **1000 ans** pour tout essayer !
La loi de Moore : énoncée par Gordon Moore en 70 :
« le nombre de transistors d'une puce doublerait tous les 18 mois à coût constant »
1975 : un ordinateur a besoin de 100 000 jours (300 ans) pour tester toutes les clés...
2000 : un ordinateur 100 000 fois plus puissant a besoin de 1 jour (un ordinateur à 200 K€) !
Challenge DES : proposé par la société RSA en janvier 1997
— cassage du DES en 96 jours ;
— février 98, cassage en 41 jours ;
— juillet 98, cassage en 56 heures sur une machine de moins de 60k€ ;
— janvier 99, cassage en moins de 24h !
Le DES a été cassé grâce aux méthodes de **cryptanalyse différentielle** et à la puissance coordonnées
des machines mises à disposition par un état par exemple.

Les évolutions

Si un algorithme est « *usé* » il est possible d'utiliser des **clés plus longues**.
Le TDES (*Triple DES*) a été créé pour pallier les limites du DES, par l'utilisation d'une chaîne de trois
chiffrements DES à l'aide de seulement deux clés différentes :
Chiffrement avec une clé C1-> **déchiffrement** avec une clé C2 -> **chiffement** avec la clé C1

L'avenir ?

Le DES et le TDES sont amenés à être remplacé par un nouvel algorithme : le **Rijndael** (du nom de ses inventeurs) qui a été sélectionné pour devenir AES.

10. Le cryptage à clé symétrique - le DES

Un standard de chiffrement
Développé dans les années 70 par IBM, la méthode DES fut adoptée et rendue standard par le gouvernement des Etats Unis. Il devait répondre à l'époque aux critères suivants :

— avoir 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.

La méthode DES utilise des clés d'une taille de 56 bits ce qui la rend de nos jours facile à casser avec les nouvelles technologies de cryptanalyse. Mais elle est toujours utilisée pour des petites tâches tel que l'échange de clés de cryptage (technologie SSL).
La clé est sur 64bits dont 8 sont utilisés comme calcul de l'intégrité des 56 autres (parité). Le DES est un standard utilisé depuis plus de 20 ans.
Il a suscité de nombreuses critiques, des suspicions de vulnérabilité à l'attaque de son algorithme, mais n'a pas eu d'alternatives jusqu'à ces dernières années : modifié par la NSA, trafiqué par IBM, …

Principe de l'algorithme

C'est un algorithme à base de :
— **décalage** ;
— « **ou exclusif** » ;
— **transposition**/**recopie** (appelé *expansion*).

Ces opérations sont **faciles** à réaliser par un processeur.

Le chiffrement par DES est très rapide.

Certaines puces spécialisées chiffrent jusqu'à 1 Go de données par seconde ce qui est énorme : *c'est*
plus que ce qu'est capable de lire un disque dur normal.

10.1. DES : l'algorithme

Principe de fonctionnement

L'algorithme utilise une clé de 56 bits. Décomposition du texte en clair en bloc : le texte en clair est découpé en bloc de 64 bits qui seront chiffrés un par un ;

Utilisation en différentes étapes, éventuellement répétées (en tout *19 étapes) :*

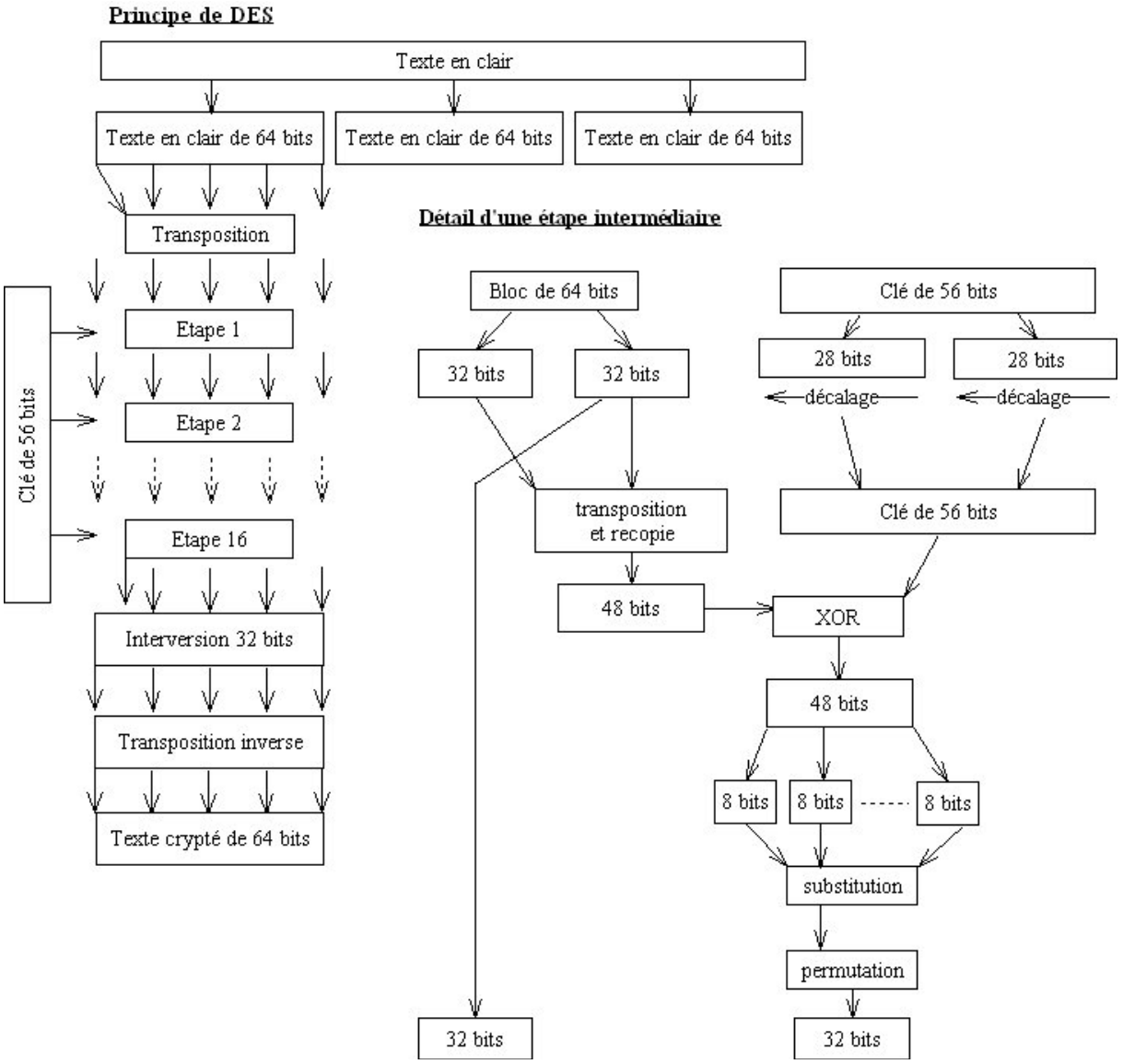
— la première étape transpose chaques blocs de 64 bits du texte en clair avec la clé de 56 bits ;
— 16 étapes intermédiaires ;
— l'avant dernière étape intervertit les 32 bits de droite et de gauche ;
— la dernière étape transpose chaques blocs de 64 bits du texte avec la clé de 56 bits (*exactement à l'inverse de la première étape*).

Les 16 étapes intermédiaires sont identiques mais varient par différentes utilisations de la clé

Une étape intermédiaire

Elle consiste à couper le bloc de 64 bits en 2 blocs de 32 bits.
Le bloc de sortie de gauche sera une recopie du bloc de droite en entrée.
Le bloc de droite est utilisé pour calculer un nombre de 48 bits à l'aide de règles de transposition et de recopie.
Ces règle sont stockées dans des tables et leur construction reste mystérieuse…le NSA y a participé !
La clé de 56 bits est divisée en 2 blocs de 28 bits, sur ces blocs de 28 bits un décalage circulaire est effectué vers la gauche d'un nombre de position dépendant de l'itération.

Un « ou exclusif » est calculé entre le nombre de 48 bits et la clé de 56 bits. Le résultat de ces « ou exclusifs » est découpé en blocs de 6 bits.



10.1.1. La cryptanalyse ?

Brute force : essayer toutes les clés possibles !

Le nombre de clés est élevé (2^56=7,2*10 16) et peut être facilement augmenté en changeant le nombre de bits pris en compte (soit exactement 72.057.595.037.927.936 clés différentes !).

Exemple : si une personne peut tester 1 million de clés par seconde
il lui faut 1000 ans pour tout essayer !

La loi de Moore : énoncée par Gordon Moore en 70 : *« le nombre de transistors d'une puce doublerait tous les 18 mois à coût constant »*
1975 : un ordinateur a besoin de 100 000 jours (300 ans) pour tester toutes les clés...
2000 : un ordinateur 100 000 fois plus puissant a besoin de 1 jour (un ordinateur à 200 K€) !

Challenge DES : proposé par la société RSA en janvier 1997

— cassage du DES en 96 jours ;
— février 98, cassage en 41 jours ;
— juillet 98, cassage en 56 heures sur une machine de moins de 60k€ ;
— janvier 99, cassage en moins de 24h !

Le DES a été cassé grâce aux méthodes de cryptanalyse différentielle et à la puissance coordonnées des machines mises à disposition par un état par exemple.

Les évolutions

Si un algorithme est « *usé* » il est possible d'utiliser des clés plus longues. Le TDES (*Triple DES*) a été créé pour pallier les limites du DES, par l'utilisation d'une chaîne de trois chiffrements DES à l'aide de seulement deux clés différentes : Chiffrement avec une clé C1-> déchiffrement avec une clé C2 -> chiffement avec la clé C1

L'avenir ?

Le DES et le TDES sont amenés à être remplacé par un nouvel algorithme : le Rijndael (du nom de ses inventeurs) qui a été sélectionné pour devenir AES.

10.2. Chiffrement à clé symétrique - Autres algorithmes

10.2.1. AES (Advanced Encryption Standard)

L'AES est un standard de cryptage symétrique destiné à remplacer le DES (Data Encryption Standard) qui est devenu trop faible au regard des attaques actuelles.

L'AES
— est un standard, libre d'utilisation, sans restriction d'usage ni brevet ;

— est un algorithme de chiffrement par blocs (comme le DES) ;
— supporte différentes combinaisons [longueur de clé]-[longueur de bloc] : 128-128, 192-128 et 256-128 bits

Le choix de cet algorithme répond à de nombreux critères tels que :
— la sécurité ou l'effort requis pour une éventuelle cryptanalyse ;
— la facilité de calcul : cela entraîne une grande rapidité de traitement ;
— les faibles besoins en ressources : mémoire très faibles ;
— la flexibilité d'implémentation : cela inclut une grande variété de plates-formes et d'applications ainsi que des tailles de clés et de blocs supplémentaires (il est possible d'implémenter l'AES aussi bien
sous forme logicielle que matérielle, câblé) ;
— la simplicité : le design de l'AES est relativement simple.

10.2.2. IDEA (International Data Encryption Algorithm)

— conçu dans les années 90 par deux chercheurs suisses (Lai et Massey) de l'ETH (Eidgenössische Technische Hochschule) de Zurich, IDEA (International Data Encryption Algorithm) ;
— utilise une clé de 128 bits ;
— résistera encore pendant quelques dizaines d'années aux attaques cryptanalytiques.

Aucune attaque existe contre l'IDEA.
IDEA est breveté aux Etats-Unis et dans de nombreux pays européens.
IDEA est gratuit tant que son utilisation reste non commerciale.

10.2.3. Blowfish

— développé par Bruce Schneier ;
— blowfish travaille par bloc de 64 bits en utilisant une clé variable pouvant aller jusqu'à 448 bits ;
— il n'existe aucun moyen de casser cet algorithme.
Blowfish est utilisé dans différents logiciels tel que NAUTILUS ou PGPFONE

10.2.4. RC4 (Rivest Cipher 4)

— algorithme de cryptage très rapide ;
— utilisé dans de multiples applications telles que les communications sécurisées pour crypter le trafic
transitant entre des interlocuteurs ;
— RC4 est basé sur l'utilisation de permutations aléatoires.
Le gouvernement des Etats-Unis autorise l'exportation du RC4 avec des clés de 40 bits.
Problème : un flux chiffré avec 2 clés identiques sera facilement cassable

10.3. Chiffrement à clé publique versus chiffrement à clé secrète

Remarques sur le chiffrement à clé publique : *L'utilisation de tels codes de chiffrement est coûteuse, ils ne peuvent pas être appliqué sur un grand débit de données à transmettre.*

Principaux algorithmes utilisés : RSA, Rivest, Shamir et Adelman 1978.
El Gamal 1981.

Remarques sur le chiffrement à clé privée
Difficulté du partage des clés, ainsi que la multiplication des clés quand plusieurs interlocuteurs sont en contact.

Dans un réseau de 5 personnes communicant entre elles il faut n(n-1)/2 clés, soient 10 clés différentes..

10.3.1. Comparaisons entre RSA et DES

RSA

— clé de 40 bits
— chiffrement matériel : 300 Kbits/sec
— chiffrement logiciel : 21,6 Kbits/sec
— Inconvénient majeur : un pirate substitue sa propre clé publique à celle du destinataire, il peut alors intercepter et décrypter le message pour le recoder ensuite avec la vraie clé publique et le renvoyer
sur le réseau. « L'attaque » ne sera pas décelée.
— usage : on ne les emploiera que pour transmettre des données courtes (de quelques octets) telles que les clés privées et les signatures électroniques.

DES

— clé de 56 bits
— chiffrement matériel : 300 Mbits/sec
— chiffrement logiciel : 2,1 Mbits/sec
— Inconvénient majeur : attaque « brute force » rendue possible par la puissance des machines.
— Usage : chiffrement rapide, adapté aux échanges de données de tous les protocoles de communication sécurisés.

10.4. Comparaison et combinaison

La sécurité offerte par le chiffrement à clé
La sécurité d'un code à clé est proportionnelle à la taille de la clé employée, c-à-d. plus la clé est longue plus il faut de calcul et donc de temps pour arriver à le casser.

Chiffrement par substitution : 26 lettres possibles associables, soit 26! (factorielle 26) soient 291 461 : 126 605 635 584 000 000 possibilités ! mais l'analyse fréquentielle...

Le chiffrement à clé : il protège des analyses fréquentielles ; Attaque « brute force » : essayer toutes les clé possibles pour déchiffrer le message chiffré, donc plus la clé est longue (nombre de bits) plus il y a de clé à essayer (2 fois plus de clé à essayer pour chaque bit ajouté !).

La force de la sécurité est à mettre en rapport avec le type de données à sécuriser :
– une transaction bancaire doit être sécurisée pendant quelques minutes
– un document secret d'état doit pouvoir être protégé plus de 50 ans par exemple.

La vitesse

Il existe un décalage de puissance de calcul pour le chiffrement/déchiffrement des codes à clé secrète (algorithme de cryptage symétrique de type DES) et à clé publique (algorithme de cryptage asymétrique de type RSA).

Code à clé secrète : applicable à un débit de données supérieur. C'est pourquoi seule l'utilisation de code à clé secrète est «réaliste» pour sécuriser une transaction entre deux utilisateurs sur Internet.
Résolution du problème de l'échange des clés secrètes :
utilisation d'une méthode hybride combinant à la fois chiffrement symétrique et asymétrique

10.5. Le chiffrement par bloc

Le chiffrement par bloc est la manière choisie pour chiffrer le message décomposé en bloc, c-à-d. dans quel ordre et après quel transformation chaque bloc va être chiffré. On parlera de mode d'opérations.

Quatres modes définis

Quatre modes sont définis dans FIPS 81, Federal Information Processing Standards Publication 81, (2 décembre 1980) et aussi dans la norme ANSI X3.106-1983.
— Electronic Code Book (ECB) ;
— Cipher Block Chaining (CBC) ;
— Cipher FeedBack (CFB) ;
— Output FeedBack (OFB).
ECB : Electronic Codebook

Mode d'opération normal : il applique l'algorithme au texte clair en transformant normalement chaque bloc de texte clair.

T[n] = nième bloc de texte en clair. Chiffrement : C[n] = E(T[n])
C[n] = nième bloc de texte chiffré. Déchiffrement : T[n] = D(C[n])
E(m) = fonction de chiffrement du bloc m. T et C sont d'une longueur fixe.

D(m) = fonction de déchiffrement du bloc m.

Problèmes :

— si on utilise deux fois le même texte clair et la même clé de chiffrement, le résultat du chiffrement sera identique.

— il faut un nombre suffisant d'octets de texte en clair (huit octets pour le DES par exemple) avant de commencer.

10.5.1. CBC : Cipher Block Chaining

C'est un des modes les plus populaires. Il apporte une solution au premier problème du mode ECB :

— avant d'être chiffré, l'opération binaire « XOR » est appliquée entre le bloc actuel de texte en clair et le bloc précédent de texte chiffré ;

— pour le tout premier bloc, un bloc de contenu aléatoire est généré et utilisé, appelé « vecteur d'initialisation » (initialization vector, ou IV).

Ce premier bloc est envoyé tel quel avec le message chiffré.

T[n] = nième bloc de texte en clair.
E(m) = fonction de chiffrement du bloc m.
C[n] = nième bloc de texte chiffré.
D(m) = fonction de déchiffrement du bloc m.
VI = vecteur d'initialisation

Chiffrement :

C[0] = E(T[0] xor VI)
C[n] = E(T[n] xor C[n-1]) , si (n > 0)

Déchiffrement :

T[0] = D(C[n]) xor VI
T[n] = D[C[n]) xor C[n-1] , si (n > 0) T et C sont d'une longueur fixe

10.5.2. OFB : Output Feedback

Le mode OFB est une solution aux deux problèmes relatifs au mode ECB.
Au départ un vecteur d'initialisation est généré. Ce bloc est chiffré à plusieurs reprises et chacun des résultats est utilisé successivement dans l'application de l'opération XOR avec un bloc de texte en clair.
Le vecteur d'initialisation est envoyé tel quel avec le message chiffré.

T[n] = nième bloc de texte en clair.
I[n] = nième bloc temporaire
C[n] = nième bloc de texte chiffré.
R[n] = nième bloc temporaire second
E(m) = fonction de chiffrement et de déchiffrement du bloc m
VI = vecteur d'initialisation

Chiffrement :

I[0] = VI

I[n] = R[n-1] , si (n > 0)
R[n] = E(I[n])
C[n] = T[n] xor R[n]

Déchiffrement :

I[0] = VI
I[n] = R[n-1] , si (n > 0)
R[n] = E(I[n])
T[n] = C[n] ^ R[n] T et C sont d'une longueur fixe

Problèmes :

— le texte en clair est seulement soumis à un XOR. Si le texte clair est connu, un tout autre texte en clair peut être substitué en inversant les bits du texte chiffré de la même manière qu'inverser les bits du texte clair (bit-flipping attack).

– il existe une petite possibilité qu'une clé et un vecteur d'initialisation soient choisis tels que les blocs successifs générés puissent se répéter sur une courte boucle.
Le mode OFB est souvent utilisé comme générateur de nombre aléatoire.

11. Le chiffrement par flux

11.1.1. Définition

Les algorithmes de chiffrement par flux peuvent être vu comme des algorithmes de chiffrement par bloc où le bloc a une dimension unitaire (1 bit, 1 octet…) ou relativement petite. Ils sont appelés stream ciphers.

Avantages :

— la méthode de chiffrement peut être changée à chaque symbole du texte clair ;
— ils sont extrêmement rapides ;
— ils ne propagent pas les erreurs (diffusion) dans un environnement où les erreurs sont fréquentes ;
— ils sont utilisables lorsque l'information ne peut être traitée qu'avec de petites quantités de symboles à la fois (par exemple si l'équipement n'a pas de mémoire physique ou une mémoire tampon très limitée).

Fonctionnement :

Ils appliquent de simples transformations selon un keystream utilisé.
Le keystream est une séquence de bits utilisée en tant que clé qui est générée aléatoirement par un algorithme (keystream generator).

Propriétés :

Avec un keystream choisi aléatoirement et utilisé qu'une seule fois, le texte chiffré est très sécurisé.
La génération du keystream peut être :

— indépendante du texte en clair et du texte chiffré, appelée chiffrement de flux synchrone (synchronous stream cipher) ;
— dépendante (self-synchronizing stream cipher).

Les chiffrements de flux les plus répandus sont synchrones Algorithmes les plus connus :
LFSR (Linear Feedback Shift Register), rapide mais vulnérable à l'heure actuelle.
RC4, inventé par Ron Rivest en 87 (société RSA), utilisé dans le protocole SSL et Oracle Secure SQL.
SEAL (Software-optimized Encryption Algorithm), Don Coppersmith et Phillip Rogaway en 93 (IBM),plus rapide que RC4.

11.1.2. Echange sécurisé

L'utilisation d'algorithme de chiffrement à clé symétrique n'est pas réaliste d'un point de vue de la puissance de calcul nécessaire.
Cette **puissance augmente** en **même temps** qu'il est nécessaire d'améliorer la sécurité de ces algorithmes (augmentation de la taille des clé) : le **décalage reste** ! Il existe alors soit à trouver un moyen de partager secrétement une même clé secrète ou bien à combiner les deux :
l'échange de la clé secrète d'un algorithme de chiffrement symétrique est « protégé » par un algorithme de chiffrement asymétrique.

Avantages :

— la clé secréte est chiffrée et échangée ;
— après l'échange on bascule le chiffrement en utilisant un algorithme symétrique plus rapide ;
— on démarre l'échange avec l'utilisation d'un algorithme asymétrique qui possède l'avantage d'offrir un
moyen d'identifier les interlocuteurs.
L'algorithme RSA a la propriété chiffrement(déchiffrement(M)) = déchiffrement(chiffrement(M)).

Échange sécurisé d'information

Cet échange se déroule en 2 phases :
— échange sécurisé d'une clé secrète pour la session, appelée également «clé de session»
— échange sécurisé des messages à l'aide d'un algorithme à clé secrète.
Une phase d'authentification des interlocuteurs peut être ajoutées au début

11.2. Clé de session

C'est un compromis entre le chiffrement symétrique et asymétrique permettant de combiner les deux techniques. Il existe deux méthodes :

Première possibilité :

— générer aléatoirement une clé de taille raisonnable utilisée pour un algorithme de cryptage symétrique;
— chiffrer cette clé à l'aide d'un algorithme de cryptage à clé publique (à l'aide de la clé publique du destinataire) ;
Cela impose que l'un des interlocuteurs possède la clé publique de l'autre (pas toujours facile de s'assurer que la clé publique appartient bien à la bonne personne).
Seconde possibilité :

— construire une clé de session à l'aide de la méthode d'échange des clés de Diffie-Hellman.
— les interlocuteurs n'ont pas besoin de partager une clé publique avant de commencer leur communication chiffrée !
Cette méthode est extrémement employée pour initier un canal de transmission sécurisée avant tout échange.

Les deux interlocuteurs disposent ensuite :
— d'une clé commune qu'ils sont seuls à connaître
— de la possibilité de communiquer en chiffrant leur données à l'aide d'un algorithme de chiffrement symétrique rapide.

11.2.1. La méthode d'échange des clés de Diffie-Hellman

Alice et Bob se mettent en accord sur deux grands nombres premiers n et g avec (n-1)/2 premier et quelques conditions sur g. Ces nombres sont publics.

Alice prend le nombre n et Bob le nombre g.
Alice choisit un nombre de 512 bits secret x, Bob fait de même avec y.
Alice envoie à Bob un message contenant le nombre n, le nombre g et le résultat de (g^x mod n)

Bob envoie à Alice le résultat de (g^y mod n)
Alice et Bob calculent (g^y mod n)^x et (g^x mod n)^y
A et B partagent maintenant la même clé secrète g^xy mod n.

Si Oscar, l'intrus capture g et n, il ne peut pas calculer x et y, car il n'existe pas de méthode humainement utilisable pour calculer x à partir de g^x mod n !

Problème : Oscar peut s'insérer entre Alice et Bob et proposé sa valeur z en lieu et place de x pour Bob et de y pour Alice :

Alice --> n, g, g^x mod n --> Oscar –> n, g, g^z mod n –> Bob
<– g^z mod n <-- <– g^y mod n <–
Conclusion : il faut une phase **préliminaire d'authentification** !

12. L'authentification

L'authentification est suivie par l'autorisation

L'autorisation définit les ressources, services et informations que la personne identifiée peut utiliser, consulter ou mettre à jour, exemple : son courrier électronique, des fichiers sur un serveur FTP…

L'approche traditionnelle

Combinaison d'une identification et d'un mot de passe (code secret personnel).
Le mot de passe doit posséder certaines caractéristiques : non trivial, difficile à deviner, régulièrement modifié, secret…
Des outils logiciel ou hardware de génération de mots de passe existent, mais les mots de passe générés sont difficiles à retenir !

L'approche évoluée, la notion de challenge/réponse

Alice envoie à Bob un message aléatoire (challenge)

Chiffement à clé secrète :

– Alice et Bob partage une même clé secrète ;
– Bob renvoie à Alice le message chiffré à l'aide de la clé secrète (réponse) ;
— Alice peut déchiffrer le message chiffré avec la clé secrète...C'est Bob !

Chiffrement à clé publique :

– Bob renvoie à Alice le message chiffré à l'aide de sa clé privée (réponse) ;
– exploitation de la propriété chiffrement(déchiffrement(M)) = déchiffrement(chiffrement(M)) ;
— Alice peut déchiffrer ce message chiffré à l'aide de la clé publique de Bob… c'est donc Bob !

Problème : cette méthode permet de faire des attaques sur la clé privée de Bob en soumettant des messages aléatoires bien choisi.
Solution : calculer un «résumé» du message aléatoire initial, un "digest", et l'utiliser à la place du message aléatoire lors du chiffrement. L'obtention de ce «résumé» se fait à l'aide d'une fonction de hachage

12.1. Fonction de hachage

Une fonction de hachage est une fonction permettant d'obtenir un résumé d'un texte, c-à-d. une suite de caractères assez courte représentant le texte qu'il résume. La fonction de hachage doit :

— être telle qu'elle associe un et un seul résumé à un texte en clair (cela signifie que la moindre modification du document entraine la modification de son résumé), c-à-d. « sans collision ».
— être une fonction à sens unique (one-way function) afin qu'il soit impossible de retrouver le message original à partir du résumé.

y = F(x), mais il est impossible de retrouver x à partir de y !

Propriétés

une fonction de hachage "H" transforme une entrée de données d'une dimension variable "m" et donne comme résultat une sortie de données inférieure et fixe "h" (h = H(m)).

— l'entrée peut être de dimension variable ;
— la sortie doit être de dimension fixe ;
— H(m) doit être relativement facile à calculer ;
— H(m) doit être une fonction à sens unique ;
— H(m) doit être « sans collision ».

Utilisation - Authentification et intégrité

Les algorithmes de hachage sont utilisés :

— dans la génération des signatures numériques, dans ce cas, le résultat "h" est appelé "empreinte" ;
— pour la vérification si un document a été modifié (le changement d'une partie du document change son empreinte) ;
— pour la construction du MAC, Message Authentication Code, ou code d'authentification de message, il permet de joindre l'empreinte du message chiffré avec une clé secrète ce qui protège contre toute modification du message (si l'intrus modifie le message et son empreinte, il est incapable de chiffrée celle-ci pour la remplacer dans le message).

12.1.1. Principaux algorithmes

Il existe différents algorithmes réalisant de traitement :

— **MD2**, **MD4** et **MD5** (MD signifiant Message Digest), développé par Ron Rivest (société RSA Security), créant une empreinte digitale de 128 bits pour MD5. Il est courant de voir des documents en téléchargement sur Internet accompagnés d'un fichier MD5, il
s'agit du résumé du document permettant de vérifier l'intégrité de ce dernier

— **SHA** (pour Secure Hash Algorithm, pouvant être traduit par Algorithme de hachage sécurisé), développé par le NIST en 1995. il crée des empreintes d'une longueur de 160 bits. C'est un standard SHA0 et SHA1 (devenu le standard SHS)
— **RACE** Integrity Primitives Evaluation Message Digest, développé par Hans Dobbertin, Antoon Bosselaers et Bart Preneel ;
— **RIPEMD**-128 et RIPEMD-160, créé entre 88 et 92 ;
— **Tiger**, développé par Ross Anderson et Eli Biham, plus rapide que MD5 (132Mb/s contre 37Mb/s sur une même machine, optimisé pour processeur 64bit).

12.2. La signature électronique

Le scellement ou sceau ou signature électronique Il est possible de :
— joindre à un document sa signature obtenue à l'aide d'une fonction de hachage en la chiffrant à l'aide de sa clé privée.

Le document peut être identifié comme provenant de la personne (Authentification) et cela assure également la non-répudiation (utilisation de la clé privée).
Il est possible de déchiffrer cette signature à l'aide de la clé publique de la personne. Cette signature permet de contrôler l'intégrité du document.
La confidentialité est assurer par un chiffrement du document.
Il est optionnel car cela nécessite du temps (utilisation d'un chiffrement à clé publique)

Fonctionnement

1. L'expéditeur calcule l'empreinte de son texte en clair à l'aide d'une fonction de hachage ;
2. L'expéditeur chiffre l'empreinte avec sa clé privée ;
Le chiffrement du document est optionnel si la confidentialité n'est pas nécessaire.
3. L'expéditeur chiffre le texte en clair et l'empreinte chiffrée à l'aide de la clé publique du destinataire.
4. L'expéditeur envoie le document chiffré au destinataire ;
5. Le destinataire déchiffre le document avec sa clé privée ;
6. Le destinataire déchiffre l'empreinte avec la clé publique de l'expéditeur (authentification) ;
7. Le destinataire calcule l'empreinte du texte clair à l'aide de la même fonction de hachage que l'expéditeur ;
8. Le destinataire compare les deux empreintes.
Deux empreintes identiques impliquent que le texte en clair n'a pas été modifié (intégrité).
Le standard américain est le DSS (Digital Signature Standard), qui spécifie trois algorithmes : le DSA
(Digital Signature Algorithm), RSA et ECDSA (Elliptic Curves Digital Signature Algorithm).

12.3. La signature électronique et la notion de certificat

Le problème de la diffusion des clés publiques

Le problème est de s'assurer que la clé que l'on récupére provient bien de la personne concernée : rien ne garantit que la clé est bien celle de l'utilisateur à qui elle est associée.
Un pirate peut remplacer la clé publique présente dans un annuaire par sa clé publique.
Ainsi, il peut déchiffrer tous les messages ayant été chiffrés avec cette clé.
Il peut même ensuite renvoyer à son véritable destinataire le message (modifié ou non) en chiffrant avec la clé originale pour ne pas être démasqué !

Notion de certificat

Un certificat permet d'associer une clé publique à une entité (une personne, une machine, ...) afin
d'en assurer la validité.
Le certificat est la carte d'identité de la clé publique, délivré par un organisme appelé autorité de certification.
Ces certificats sont émis et signé par une tierce partie, l'autorité de certification ou CA (Certificate Authority).
L'autorité de certification est chargée de
– délivrer les certificats ;
– d'assigner une date de validité aux certificats (équivalent à la date limite de péremption des produits
alimentaires) ;

– révoquer éventuellement des certificats avant cette date en cas de compromission de la clé (ou du propriétaire).

13. SSL

13.1.1. Introduction

SSL (**Secure Sockets Layers**, que l'on pourrait traduire par *couche de sockets sécurisée*) est un procédé de sécurisation des transactions effectuées via Internet. Le standard SSL a été mis au point par *Netscape*, en collaboration avec *Mastercard*, *Bank of America*, *MCI* et *Silicon Graphics*. Il repose sur un procédé de cryptographie par clef publique afin de garantir la sécurité de la transmission de données sur internet. Son principe consiste à établir un canal de communication sécurisé (chiffré) entre deux machines (un client et un serveur) après une étape d'authentification.

Le système SSL est indépendant du protocole utilisé, ce qui signifie qu'il peut aussi bien sécuriser des transactions faites sur le Web par le protocole HTTP que des connexions via le protocole FTP, POP ou IMAP. En effet, SSL agit telle une couche supplémentaire, permettant d'assurer la sécurité des données, située entre la couche application et la couche transport (protocole TCP par exemple).

De cette manière, SSL est transparent pour l'utilisateur (entendez par là qu'il peut ignorer qu'il utilise SSL). Par exemple un utilisateur utilisant un navigateur internet pour se connecter à un site de commerce électronique sécurisé par SSL enverra des données chiffrées sans aucune manipulation nécessaire de sa part.

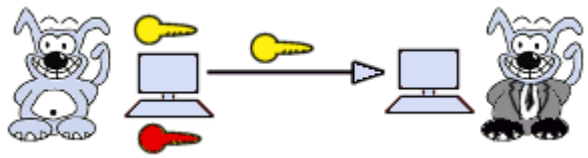
La quasi intégralité des navigateurs supporte désormais le protocole SSL. *Netscape Navigator* affiche par exemple un cadenas verrouillé pour indiquer la connexion à un site sécurisé par SSL et un cadenas ouvert dans le cas contraire, tandis que *Microsoft Internet Explorer* affiche un cadenas uniquement lors de la connexion à un site sécurisé par SSL.

13.1.2. Fonctionnement de SSL 2.0

La sécurisation des transactions par SSL 2.0 est basée sur un échange de clés entre client et serveur. La transaction sécurisée par SSL se fait selon le modèle suivant :

Dans un premier temps, le client se connecte au site marchand sécurisé par SSL et lui demande de s'authentifier. Le client envoie également la liste des cryptosystèmes qu'il supporte, triée par ordre décroissant selon la longueur des clés.

Le serveur a réception de la requête envoie un certificat au client, contenant la clé publique du serveur, signée par une autorité de certification (CA), ainsi que le nom du cryptosystème le plus haut dans la liste avec lequel il est compatible (la longueur de la clé de chiffrement - 40 bits ou 128 bits - sera celle du cryptosystème commun ayant la plus grande taille de clé).



Le client vérifie la validité du certificat (donc l'authenticité du marchand), puis crée une clé secrète aléatoire (plus exactement un bloc prétenduement aléatoire),

chiffre cette clé à l'aide de la clé publique du serveur, puis lui envoie le résultat (la clé de session).
Le serveur est en mesure de déchiffrer la clé de session avec sa clé privée. Ainsi, les deux entités sont en possession d'une clé commune dont ils sont seuls connaisseurs. Le reste des transactions peut se faire à l'aide de clé de session, garantissant l'intégrité et la confidentialité des données échangées.

13.1.3. SSL 3.0

SSL 3.0 vise à authentifier le serveur vis-à-vis du client et éventuellement le client vis-à-vis du serveur.

14. La PKI

14.1.1. Introduction à la notion de certificat

Les algorithmes de chiffrement asymétrique sont basés sur le partage entre les différents utilisateurs d'une clé publique. Généralement le partage de cette clé se fait au travers d'un annuaire électronique (généralement au format LDAP) ou bien d'un site web.
Toutefois ce mode de partage a une grande lacune : rien ne garantit que la clé est bien celle de l'utilisateur a qui elle est associée. En effet un pirate peut corrompre la clé publique présente dans l'annuaire en la remplaçant par sa clé publique. Ainsi, le pirate sera en mesure de déchiffrer tous les messages ayant été chiffrés avec la clé présente dans l'annuaire.
Ainsi un certificat permet d'associer une clé publique à une entité (une personne, une machine, ...) afin d'en assurer la validité. Le certificat est en quelque sorte la carte d'identité de la clé publique, délivré par un organisme appelé autorité de certification (souvent notée CA pour Certification Authority).
L'autorité de certification est chargée de délivrer les certificats, de leur assigner une date de validité (équivalent à la date limite de péremption des produits alimentaires), ainsi que de révoquer éventuellement des certificats avant cette date en cas de compromission de la clé (ou du propriétaire).

14.1.2. Structure d'un certificat ?

Les certificats sont des petits fichiers divisés en deux parties :

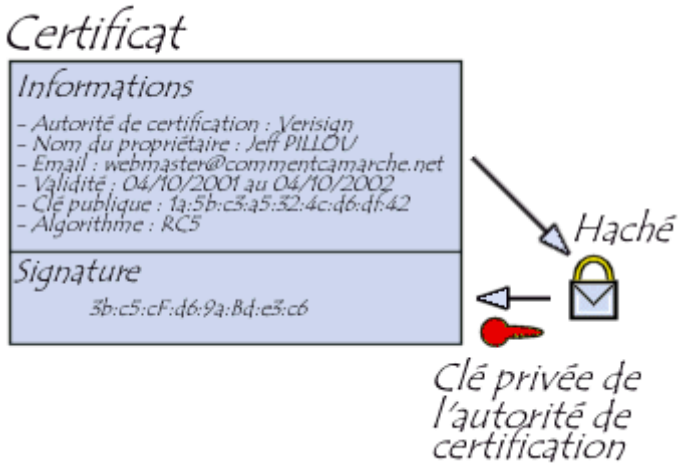
- La partie contenant les informations
- La partie contenant la signature de l'autorité de certification

La structure des certificats est normalisée par le standard X.509 de l'UIT (plus exactement X.509v3), qui définit les informations contenues dans le certificat :

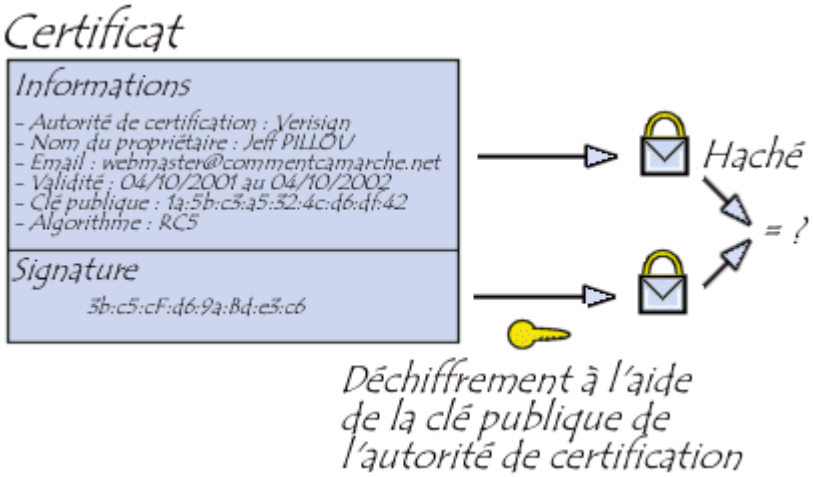
- La version de X.509 à laquelle le certificat correspond ;

- Le numéro de série du certificat ;
- L'algorithme de chiffrement utilisé pour signer le certificat ;
- Le nom (DN, pour Distinguished Name) de l'autorité de certification émettrice ;
- La date de début de validité du certificat ;
- La date de fin de validité du certificat ;
- L'objet de l'utilisation de la clé publique ;
- La clé publique du propriétaire du certificat ;
- La signature de l'émetteur du certificat (thumbprint).

L'ensemble de ces informations (informations + clé publique du demandeur) est signé par l'autorité de certification, cela signifie qu'une fonction de hachage crée une empreinte de ces informations, puis ce condensé est chiffré à l'aide de la clé privée de l'autorité de certification; la clé publique ayant été préalablement largement diffusée afin de permettre aux utilisateurs de vérifier la signature avec la clé publique de l'autorité de certification.



Lorsqu'un utilisateur désire communiquer avec une autre personne, il lui suffit de se procurer le certificat du destinataire. Ce certificat contient le nom du destinataire, ainsi que sa clé publique et est signé par l'autorité de certification. Il est donc possible de vérifier la validité du message en appliquant d'une part la fonction de hachage aux informations contenues dans le certificat, en déchiffrant d'autre part la signature de l'autorité de certification avec la clé publique de cette dernière et en comparant ces deux résultats.



14.1.3. Signatures de certificats

On distingue différents types de certificats selon le niveau de signature :

- Les certificats auto-signés sont des certificats à usage interne. Signés par un serveur local, ce type de certificat permet de garantir la confidentialité des échanges au sein d'une organisation, par exemple pour le besoin d'un intranet. Il est ainsi possible d'effectuer une authentification des utilisateurs grâce à des certificats auto-signés.
- Les certificats signés par un organisme de certification sont nécessaires lorsqu'il s'agit d'assurer la sécurité des échanges avec des utilisateurs anonymes, par exemple dans le cas d'un site web sécurisé accessible au grand public. Le certificateur tiers permet d'assurer à l'utilisateur que le certificat appartient bien à l'organisation à laquelle il est déclaré appartenir.

14.1.4. Types d'usages

Les certificats servent principalement dans trois types de contextes :
Le certificat client, stocké sur le poste de travail de l'utilisateur ou embarqué dans un conteneur tel qu'une carte à puce, permet d'identifier un utilisateur et de lui associer des droits. Dans la plupart des scénarios il est transmis au serveur lors d'une connexion, qui affecte des droits en fonction de l'accréditation de l'utilisateur. Il s'agit d'une véritable carte d'identité numérique utilisant une paire de clé asymétrique d'une longueur de 512 à 1024 bits.
Le certificat serveur installé sur un serveur web permet d'assurer le lien entre le service et le propriétaire du service. Dans le cas d'un site web, il permet de garantir que l'URL et en particulier le domaine de la page web appartiennent bien à telle ou telle entreprise. Par ailleurs il permet de sécuriser les transactions avec les utilisateurs grâce au protocole SSL.
Le certificat VPN est un type de certificat installé dans les équipement réseaux, permettant de chiffrer les flux de communication de bout en bout entre deux points (par exemple deux sites d'une entreprise). Dans ce type de scénario, les utilisateurs possèdent un certificat client, les serveurs mettent en oeuvre un certificat serveur et les équipements de communication utilisent un certificat particulier (généralement un certificat IPSec.

14.1.5. Le but de PKI

Le but de la PKI est de présenter l'autorité de certification, à savoir une entité humaine (une personne, un groupe, un service, une entreprise ou une autre association) autorisée par une société à émettre des certificats à l'attention de ses utilisateurs informatiques.
Une autorité de certification fonctionne comme un service de contrôle des passeports du gouvernement d'un pays.
L'autorité de certification **crée** des **certificats** et les signe de façon numérique à l'aide d'une clé privée qui lui appartient.
Elle gère une **liste** de **révocation** qui permet d'invalider des certificats déja diffusés.

Vérification d'un certificat

On peut vérifier la signature numérique de l'AC émettrice du certificat, à l'aide de la clé publique de l'AC.
Si c'est le cas alors il garantit l'intégrité du contenu du certificat (la clé publique et l'identité du détenteur du certificat).
L'utilisation que fait le titulaire du certificat ne concerne plus la PKI, mais les diverses applications qui sont compatibles.

Confiance dans le PKI

L'ensemble des personnes et des services doivent faire confiance à la PKI :
- signature des courriers,
- chiffrement,
- authentification sur des applications maison…

Ils doivent également savoir déchiffrer un certificat et être capable de contacter l´Autorité de Certification afin de vérifier la validité du certificat auprès de la liste de révocation. La PKI n'est qu'une simple couche destinée à faciliter la gestion des identités numériques à grande échelle. Elle est totalement indépendante des applications éventuelles qui utilisent ces identités.

L'émission de certificat n'est pas le seul service de l'IGC

— vérifie l'identité du titulaire lors de l'émission de certificat
— publie le certificat,
— assure le renouvellement,
— révoque les certificats invalidés,
assure parfois le recouvrement de la clef privée.

La révocation

Accepteriez vous d'utiliser une carte bancaire si vous ne pouviez par y faire opposition, même en cas de vol ?
Les CRL : la liste des certificats révoqués, liste signée par la CA
Mal implémenté dans les navigateurs
Pas encore de CRL incrémentale.
Alternative : OCSP
La révocation est une limite théorique au modèle des PKIs.

Les composants

On distingue différents composants dans une IGC :
— Autorité de certification AC (certificat authority)
— Autorité d'enregistrement AE (registry authority)
— Interface utilisateur (Enrolment Entity)

14.2. Les différentes autorités

L'autorité de certification :C'est une organisation qui délivre des certificats à une population. Il existe des
– autorités privées (intranet d'une entreprise),
– organisationnelles (CRU, CNRS),
– corporative (notaires),
– commerciales (Thawte, Verisign, …),
– très commerciales (Microsoft),

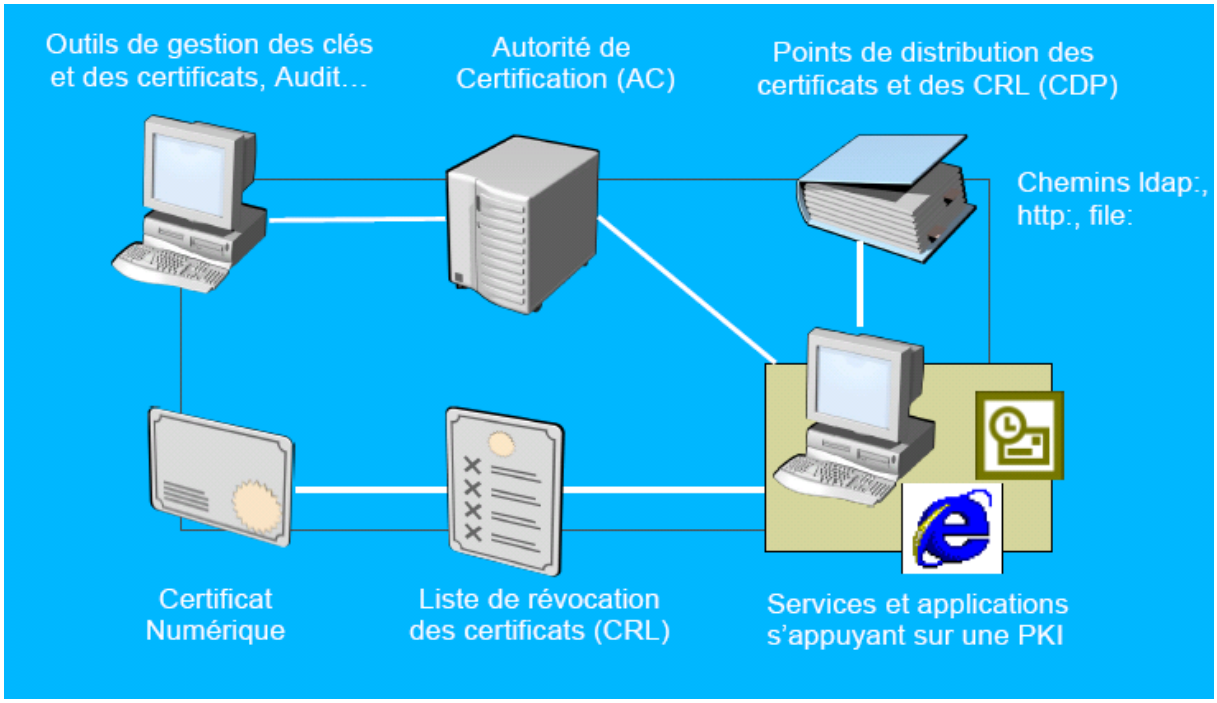
— institutionnelles, etc

Les tâches de l'AC

– Protège la clé privée de la AC (bunker informatique)
– Vérifie les demandes de certificats (Certificat Signing Request) provenant des AE
– Génère les certificats et les publie
— Génère les listes de certificats révoqués (Certificat Revocation List)

L'autorité d'enregistrement

Vérifie l'identité des demandeurs de certificats et les éléments de la demande.
Exemple :
– *L'email présent dans le DN est-il l'email canonique ?*
– *Le demandeur a-t-il le droit de disposer d'un certificat de signature ?*
Transmet les demandes valides par un canal sûr à l'AC (demandes signées par l'opérateur de la AC)
Recueille et vérifie les demandes de révocation



Les composants du PKI