Université Cheikh Anta DIOP de Dakar

Faculté des Sciences et Techniques

Département Mathématiques et Informatique

Laboratoire d'Algèbre de Géométrique Algèbrique et Applications

L. A. C. G. A. A.

TITRE :

Sécurité Informatique et Cryptographie

présenté par

Dr Demba SOW

Année universitaire 2013 / 2014

Table des matières

Chapitre 1

SECURITE INFORMATIQUE

1.1 INTRODUCTION

Qu'est ce qu'un système d'information ?

Système d'information

- Organisation des activités consistant à acquérir, stocker, transformer, diffuser, exploiter, gérer ... les informations ;
- Un des moyens techniques pour faire fonctionner un système d'information est d'utiliser un **Système informatique**.

La sécurité des systèmes informatiques

La sécurité informatique c'est l'ensemble des moyens mis en oeuvre pour réduire la vulnérabilité d'un système contre les menaces accidentelles ou intentionnelles.

- Les systèmes informatiques sont au coeur des systèmes d'information.
- Ils sont devenus la cible de ceux qui convoitent l'information.
- Assurer la sécurité de l'information implique d'assurer la sécurité des systèmes informatiques.

1.2 PRINCIPES DE LA SÉCURITÉ

1.2.1 Objectifs de la sécurité informatique

Les exigences fondamentales en sécurité informatique se caractérisent ce à quoi s'attendent les utilisateurs de systèmes informatiques en regard de la sécurité :

1. **Intégrité :** demande que l'information sur le système ne puisse être modifiée que par les personnes autorisées.

2. **Confidentialité :** demande que l'information sur le système ne puisse être lue que par les personnes autorisées.

3. **Disponibilité :** demande que l'information sur le système soit disponible aux personnes autorisées.

4. **Authentification :** demande que seules les personnes autorisées aient accès aux ressources.

5. **Non répudiation :** permettant de garantir qu'une transaction ne peut être niée.

1.2.2 Aspects couverts

- **intégrité des informations** (pas de modification ni destruction)
- **confidentialité** (pas de divulgation à des tiers non autorisés)
- **authentification des interlocuteurs** (signature)
- **respect de la vie privée** (informatique et liberté).

♦ **Du point de vue de la sécurité informatique, une menace est une violation potentielle de la sécurité.**

♦ **Cette menace peut-être accidentelle, intentionnelle (attaque), active ou passive.**

La sécurité doit être abordée dans un contexte global et notamment prendre en compte les aspects suivants :

► **La sécurité physique**, soit la sécurité au niveau des infrastructures matérielles : salles sécurisées, lieux ouverts au public, espaces communs de l'entreprise, postes de travail des personnels.

► **La sécurité personnelle :** la sensibilisation des utilisateurs aux problèmes de sécurité.

► **La sécurité logique :** c'est-à-dire la sécurité au niveau des données, notamment les données de l'entreprise, les applications ou encore les systèmes d'exploitation.

► **La sécurité des communications :** technologies réseau, serveurs de l'entreprise, réseaux d'accès, etc.

► **La sécurité procédurale :** sert de tampon entre les commandes juridiques et les livraisons techniques.

1.2.3 Vulnérabilités

- Les **vulnérabilités** : ce sont les failles de sécurité dans un ou plusieurs systèmes.
- Une **vulnérabilité** peut se définir comme une faiblesse ou une faille dans les procédures de sécurité, les contrôles administratifs, les contrôles internes d'un système, qui pourrait être exploitée pour obtenir un accès non autorisé à un Système d'Information (SI), à un de ses services ou à des informations.
- Tout système vu dans sa globalité présente des vulnérabilités, qui peuvent être exploitables ou non.

1.2.4 Evolution des risques

Définition : RISQUE

- Un **risque** est un danger, un inconvénient plus ou moins probable auquel on est exposé dans un système d'information.
- Il est généralement admis que le **risque est une fonction** de la **menace**, des **vulnérabilités** et des **contre-mesures** (ensemble de mesures adoptées pour contrer les menaces et les failles).

CAUSES DE L'EVOLUTION DES RISQUES :

- **Croissance de l'Internet**
- **Croissance des attaques**
- **Failles des technologies**
- **Failles des configurations**
- **Failles des politiques de sécurité**
- **Changement de profil des pirates**

1.2.5 Défauts de sécurité

Les défauts de sécurité d'un système d'information les plus souvent constatés sont :

- ▷ Traces inexploitées.
- ▷ Authentification faible.
- ▷ Mises à jours non effectuées.
- ▷ Procédures de sécurité obsolètes.
- ▷ Télémaintenance sans contrôle fort.
- ▷ Services inutiles conservés (Netbios ...).

- ▷ Mots de passe inexistants ou par défaut.
- ▷ Installation des logiciels et matériels par défaut.
- ▷ Pas de séparation des flux opérationnels des flux d'administration des systèmes.
- ▷ Eléments et outils de test laissés en place dans les configurations en production.

1.3 FAILLES DE SÉCURITÉ

1.3.1 Failles de sécurité sur Internet

- En entreprise, c'est le réseau local qui est connecté à Internet. Il est donc indispensable de contrôler les communications entre le réseau interne et l'extérieur.
- De plus une formation du personnel est indispensable (règles de sécurité, déontologie, attention aux participations aux forums qui sont archivées ...).
- **Voici quelques exemples de failles de sécurité dur internet :**

▷ ***IP spoofing*** :
- ◦ Usurpation d'adresse IP, on fait croire que la requête provient d'une machine autorisée.
- ◦ Une bonne configuration du routeur d'entrée permet d'éviter qu'une machine extérieure puisse se faire passer pour une machine interne.

▷ ***DNS spoofing*** :
- ◦ Pousse un serveur de DNS à accepter l'intrus.
- ◦ **Solution :** séparer le DNS du LAN de celui de l'espace public.

▷ ***Flooding*** : Raid massif de connexions non terminées.

▷ ***Smurf*** : Saturation de la bande passante.

▷ ***Hoax (rumeur)*** :
- ◦ Un "**hoax** " est une rumeur que l'on transmet par mail.
- ◦ Ces rumeurs colportent souvent des problèmes de sécurité soit disant découverts par des services officiels ou célèbres ...
- ◦ Elles peuvent causer un véritable préjudice à certaines sociétés et de toute façon encombrent le réseau.
- ◦ Avant de retransmettre un tel message il est prudent de vérifier son authenticité.

▷ ***Hacker*** et ***cracker*** : **HACKER et CRACKER PAS UNE GRANDE DIFFERENCE ! ! !**
- ◦ **Hackers :** informaticiens généralement discrets, anti-autoritaristes et motivés par la curiosité (principalement des adolescents de sexe masculin) et qui s'introduisent à distance dans les systèmes informatiques et en piratant les systèmes téléphoniques, généralement à l'aide d'outils écrit par d'autres et trouvés sur Internet.
- ◦ **Cracker :** pirate informatique qui déjoue les protections de logiciels ou de réseaux

et cherche à détruire les programmes avec des virus.

- Les vrais **hackers** appellent ces gens des "**crackers**" et ne veulent rien avoir à faire avec eux et pensent que les crackers sont des gens paresseux, irresponsables et pas très brillants ...

1.3.2 Principales attaques

▷ ***Virus*** :

- Le virus est un exécutable qui va exécuter des opérations plus ou moins destructrices sur votre machine.
- Les virus existent depuis que l'informatique est née et se propageaient initialement par disquettes de jeux ou logiciels divers ...

▷ ***Déni de service (DoS)*** :

- Le but d'une telle attaque n'est pas de dérober des informations sur une machine distante, mais de paralyser un service ou un réseau complet.
- Les utilisateurs ne peuvent plus alors accéder aux ressources.

▷ ***Écoute du réseau (sniffer)*** :

- Il existe des logiciels qui, à l'image des analyseurs de réseau, permettent d'intercepter certaines informations qui transitent sur un réseau local, en retranscrivant les trames dans un format plus lisible (Network packet sniffing).

▷ ***Intrusion*** :

- L'intrusion dans un système informatique a généralement pour but la réalisation d'une menace et est donc une attaque.
- Les conséquences peuvent être catastrophiques : vol, fraude, incident diplomatique, chantage ...

▷ ***Cheval de Troie*** :

- L'image retenue de la mythologie est parlante ; le pirate, après avoir accédé à votre système ou en utilisant votre crédulité, installe un logiciel qui va, à votre insu, lui transmettre par Internet les informations de vos disques durs.
- Un tel logiciel, aussi appelé ***troyen*** ou ***trojan***, peut aussi être utilisé pour générer de nouvelles attaques sur d'autres serveurs en passant par le votre.

▷ ***"Social engeneering"*** :

- En utilisant les moyens usuels (téléphone, email ...) et en usurpant une identité, un pirate cherche à obtenir des renseignements confidentiels auprès du personnel de l'entreprise en vue d'une intrusion future.
- Seule une formation du personnel permet de se protéger de cette attaque.

1.3.3 Espionnage

▷ ***L'homme du milieu*** :

- Lorsqu'un pirate, prenant le contrôle d'un équipement du réseau, se place au milieu d'une communication il peut écouter ou modifier celle-ci.
- On parle alors de **"l'homme du milieu" (man in the middle).**

▷ ***Espiogiciels*** :

- Ces logiciels espions sont aussi appelés "**spyware**".
- Ils ne posent pas, à priori, de problème de sécurité mais plutôt celui du respect de la vie privée.

▷ ***Un "cookies"*** :

- C'est une chaîne de caractère qu'un serveur dépose sur votre disque dur, via votre navigateur, afin normalement d'accélérer ou d'autoriser votre prochaine visite.

1.4 PROTECTIONS

1.4.1 Formation des utilisateurs

On considère généralement que la majorité des problèmes de sécurité sont situés entre la **chaise** et le **clavier** ...!;

▷ **Discrétion** :

- La sensibilisation des utilisateurs à la faible sécurité des outils de communication et à l'importance de la non divulgation d'informations par ces moyens est indispensable.
- En effet il est souvent trop facile d'obtenir des mots de passe par téléphone ou par e-mail en se faisant passer pour un membre important de la société.

▷ **Virus** :

- Eviter d'ouvrir les messages tels que : "**VOUS AVEZ GAGNE**", "**I LOVE YOU**" ou similaires ...!
- L'information régulière du personnel est nécessaire, attention toutefois aux rumeurs (hoax).

▷ **Charte** :

- L'intérêt principal d'une charte d'entreprise est d'obliger les employés à lire et signer un document précisant leurs droits et devoirs et par la même de leur faire prendre conscience de leur responsabilité individuelle.

1.4.2 Poste de Travail

▷ **POSTE DE TRAVAIL** :

- Le poste de travail reste un maillon faible de la sécurité.
- Les fabricants cherchent à améliorer la sécurité du PC en lui dotant d'une puce dédiée à la sécurité.
- Cette puce se chargera de vérifier l'intégrité du BIOS, du chargement de l'OS, de sauvegarder les clés et certificats (PKI) et connaîtra les protocoles de cryptage (RSA, DES, ...).

1.4.3 Antivirus

▷ **ANTIVIRUS** :

- Principale cause de désagrément en entreprise, les virus peuvent être combattus à plusieurs niveaux.
- La plupart des antivirus sont basés sur l'analyse de signature des fichiers, la base des signatures doit donc être très régulièrement mise à jour sur le site de l'éditeur (des procédures automatiques sont généralement possibles).

1.4.4 Pare-Feu

▷ **PARE-FEU (fire wall) ou GARDE BARRIÈRE** :

- C'est une machine dédiée au routage entre LAN et Internet. Le trafic est analysé au niveau des datagrammes IP (adresse, utilisateur, contenu...).
- Un datagramme non autorisé sera simplement détruit, IP sachant gérer la perte d'information.
- Une translation d'adresse pourra éventuellement être effectuée pour plus de sécurité (protocole NAT Network Address Translation).
- **Attention : un firewall est inefficace contre les attaques ou les bévues situées du coté intérieur et qui représentent 70% des problèmes de sécurité !**

Chapitre 2

CRYPTOLOGIE

2.1 CONCEPTS DE BASE

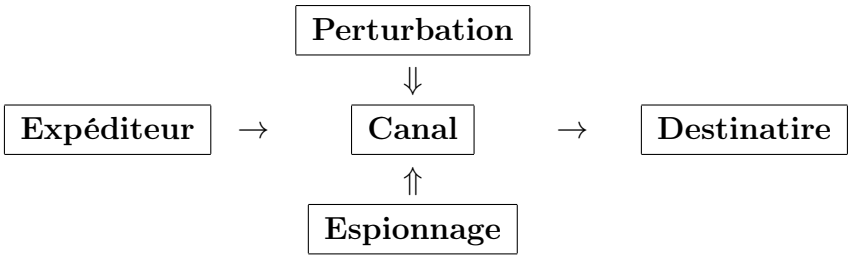
UN PEU DE VOCABULAIRE

- **Information :** élément de connaissance : texte, son, image, vidéo,...
- **Traiter/manipuler une information :** lire, écrire/modifier, effacer une information ;
- **Canal :** moyen de transmission permettant de convoyer une information entre deux partenaires d'une communication ; par exemple ligne téléphonique, fibre optique, moyen de communication sans fils,...
- **Entités :** quelqu'un (= personne) ou quelque chose (= machnie,...) capable de d'envoyer, de recevoir ou de traiter/manipuler une information. Dans la pratique, c'est l'un des partenaires d'une commmunication.
- **Un schémas de communication :** échange d'informations entre entités distantes

SCHEMA DE COMMUNICATION

Un schémas de communication (échange d'informations entre entités distantes) fait intervenir :

- **un expéditeur** (personne, machine,...),
- **un destinataire** (personne, machine,...),
- **un canal de transmission** (ligne téléphonique, fibre optique, systèmes de communication sans fils,...)
- **et un message** (information : texte, son, image, vidéo,...).



► Habituellment (ou systématiquement), le canal connait des perturbations (électromagnétique,...) et des personnes ou entités tiers interviennent pour espionner ou compromettre les communications.

► Avant la transmission, le message subit des transformations particulières appélées **codages** qui prennent en charge tous les problèmes (d'efficacité, de sécurité,...) qui se posent lors de la transmission.

► Il y'a trois types de codes : les **codes de compressions**, les **codes correcteurs d'erreurs** et les **codes secrets**.

DEFINITION DES DIFFERENTS CODES

► **Codes de compréssion** : Lors d'une communication, l'information est transmise sous la forme d'une séquence de signaux et donc pour des soucis d'éfficacité (relativment à la capacité du canal), il faut utiliser des codes qui minimisent la longueur de la séquence (**Compression**).

► **Codes corecteurs d'erreurs :** Lors d'une transmission, après codage, des erreurs apparaissent car il y'a des signaux qui sont perdus ou altérés. A la reception, il est necessaire de pouvoir detecter et si possible de corriger les erreurs survenues (**Détection/Correction**).

► **Codes secrets** Lors d'une communication, un espion peut tenter de violer la confidentialité des données, de détourner et de modifier les données ou d'essayer d'usurper l'identitité de l'un des partenaires de la communication. Ainsi pour des raisons de sécurité, lors de l'envoie, un codage et/ou un protocole bien spécifique doit etre utilisé pour protéger la communication par rapport à des besoins de sécurité bien identifiés (**Cryptographie**).

TERMINOLOGIE : cryptologie, crytographie, cryptanalyse

► **Cryptologie** : c'est une science qui comporte deux branches la **cryptographie** et la **cryptanalyse**.

► **cryptographie** : traditionnellement c'est l'étude des méthodes permettant de transmettre des données de manière confidentielle. Afin de protéger un message, on lui applique une transformation qui le rend incompréhensible ; c'est ce qu'on appelle le **chiffrement**, qui, à partir d'un texte clair donne un **texte chiffré** ou **cryptogramme**. Inversement le **déchiffrement** est l'action qui permet de reconstruire le texte clair à partir du texte chiffré.

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

► **cryptanalyse** : à l'inverse, c'est l'étude des procédés cryptographiques dans le but

de trouver des faiblesses et, en particulier, de pouvoir décrypter des textes chiffrés. Le **décryptement** est l'action consistant à retrouver le texte en clair sans connaitre la clé de déchiffrement.

Remarque 2.1.1. *Les termes "**cryptage**" et "**crypter**" sont des anglicismes dérivés de l'anglais* **to encrypt**, *souvent employés incorrectement à la place de **chiiffrement** et **chiffrer**.*

En toute rigueur, ces termes n'existent pas dans la langue française. Si le cryptage existait, il pourrait être défini comme l'inverse du décryptage, c'est-à-dire comme l'action consistant à obtenir un texte chiffré à partir d'un texte en clair sans connaître la clé.

SYSTEMES DE CRYPTOGRAPHIE

Un système de cryptographie est composé d'un quintuplet $(\mathcal{P}, \mathcal{C}, C_k, D_{k'}, \mathcal{K})$ où :

- $\mathcal{P}$ est un ensemble appelé espace des textes clairs
- $\mathcal{C}$ est un ensemble appelé espace des textes chiffrés
- $\mathcal{K}$ est un ensemble appelé espace des clés
- $Gen_{\mathcal{K}}$ un algorithme de génération de clés (=les éléments de $\mathcal{K}$) ;
- $C_k : \mathcal{P} \to \mathcal{C}$ est une fonction inversible à gauche appelée fonction de chiffrement et qui dépend d'un parametre k appelé clé.
- $D_{k'} : \mathcal{C} \to \mathcal{P}$ est la fonction inverse gauche de C_k (i.e $D_{k'} \circ C_k(m) = m, \forall m \in \mathcal{P}$) et est appelée fonction de déchiffrement (dépendant de la clé k'.)

TYPES DE SYSTEMES

En cryptographie les systémes peuvent être classés en deux catégories :

- les systèmes à clés secrètes (voir chapitre 2) ;
- les systèmes à clés publique/privée (voir chapitre 3) ;

Les systèmes à clés publique/privée sont, à leur tour, composés de deux familles :

- ceux basés sur **des algorithmes déterministes** (*pour une même donnée d'entrée, l'agorithme déterministe exécute toujours la même séquence d'operations et produit le même résultat*) :
- ceux basés sur **des algorithmes probabilistes** :(*pour une même donnée d'entrée, l'algorithme choisit la séquence d'opérations à exécuter avec une certaine probabilité et peut produire des résultats différents même si la même donnée est prise plusieurs fois en entrée*)

DEFINITION DES SYSTEMES

- **Système symétique** : un système est dit symétrique si une seule clé est utilisée pour le chiffrement et pour le déchiffrement. On parle dans ce cas **système à clés secrètes**.

- **Système asymétique** : si deux clés différentes sont utilisées, l'une pour le chiffrement, l'autre pour le déchiffrement, on parle de clés asymétriques. Dans ce modèle généralement, l'une des clé est publiée (**clé publique**) et l'autre est gardée par son propriétaire (**clé privée**). Par opposition aux systèmes à clés secrètes, les systèmes asymétriques sont aussi appelés **systèmes à clés publiques**.

- **Système hybride** : c'est l'utilisation des deux systèmes à la fois dans un schéma de communication.

2.2 MECANISMES ET SERVICES DE SECURITE

La cryptologie est à la fois une science et une technologie. Science, dont les principes les plus récents sont encore l'occasion de nouvelles découvertes. Technologie, utile et nécessaire dans l'industrie de la sécurité et pour tous ceux qui veulent protéger leur information.

Si le but traditonnel de la cryptographie est d'élaborer des méthodes permettant de transmettre des données de manière confidentielle, **la cryptographie moderne s'attaque plus généralement aux problèmes de sécurité des communications**.

Le but est d'offrir un certain nombres de services de sécurité comme la **confidentialité** des données échangées, l'**authentification** des données transmises et des tiers, l'**intégrité** de l'information échangée et la **non-répudiation** des participants. Pour cela on utilise un certains nombres de **mécanismes** basés sur les **algorithmes cryptographies**.

La cryptologie couvre couramment quatre grandes fonctions de sécurité :

- **Confidentialité** : il s'agit de garantir le secret de l'information transmise ou archivée. En général, on utilise le **chiffrement** au moyen d'une clé symétrique.

- **Authentification (L'identification)** : il s'agit de garantir l'origine d'une information.
  En général, on utilise la **signature numérique** avec un couple de clés dont celle permettant de créer les signatures est gardée secrète, et dont l'autre permettant de vérifier la signature est rendue publique.
  Le courrier électronique, un bon de commande transmis en ligne, un acte administratif peuvent être signés pour prouver leur origine et engager le signataire, à l'identique d'un paraphe sur le papier.
  L'identification : il s'agit de garantir l'identité et la qualité d'une personne qui souhaite accéder à des informations ou à des ressources matérielles.
  En général, on utilise le contrôle d'accès par mot de passe. Pour consulter son courrier électronique, pour se connecter à un ordinateur distant, ou pour entrer dans un lieu protégé, on peut ainsi s'assurer de l'identité du demandeur.

- **Intégrité** : il s'agit de garantir l'intégrité, c'est-à-dire l'absence de modification d'un message ou d'un document. On peut utiliser la **signature numérique** sous sa forme symétrique ou asymétrique, ou encore le **chiffrement**.
  Il est particulièrement important que, dans toute négociation et accord contractuel, on puisse vérifier qu'aucune modification du document électronique n'a été faite.

- **Non-répudiation** : il s'agit de garantir qu'aucun des partenaires d'une transaction ne pourra nier d'y avoir participée.

2.3 NOTIONS DE CRYPTANALYSE

2.3.1 Définitions

La cryptanalyse a pour principal objet, d'étudier les faiblesses des outils de sécurité produits par la cryptographie dans le but de les corriger ou nuire au système de communication .

Attaquant : Entité [nommée **Charlie**] suceptible d'agir sur un schémas de communication dans le but de nuire c'est à dire :

- de violer la confidentialité des données,
- de détourner et de modifier des données ou de récupérer des informations,
- d'usurper l'identitité de l'un des partenaires de la communication.

2.3.2 Types d'attaques

Il y'a deux classes d'attaques :

1. **Attaques passives** : l'attaquant écoute seulement. Donc, c'est une attaque qui cible la confidentialité.
2. **Attaques actives** : l'attaquant agit sur le système de commnucation pour :
   - détourner et modifier des données ou récupérer des informations,
   - usurper l'identitité de l'un des partenaires de la communication.

   Donc les attaques actives ciblent toutes les fonctions de sécurité : confidentialité, intégrité, authentification-identification et non répudiation.

Pour celà, l'attaquant cherche généralement à mettre à défaut l'une des primitives de cryptographie. Par exemple :

- **pour les algorithmes** : récupérer les clés de chiffrement, déchiffrer les cryptogrammes ou de casser complètement les algorithmes utilisés ;
- **pour les protocoles** : détourner l'objectif d'un protocole, compromettre le déroulement d'un protocole ;

- **pour le hachage** : fabriquer de fausses empreintes,
- **pour les signatures** : falsifier les signatures.

Ces attaques peuvent être dirigées :

- sur les modèles mathématiques utilisés pour fabriquer les primitives de cryptographie ;
- sur l'implémentation matériel et/ou logiciel des pritmitives de cryptographie ;
- sur les entités propriétaires (légitimes) de données ou d'objets cryptographiques (secrets) ;
- sur les acteurs légitimes d'un scénarios de communication ;
- sur la gestion (fabrication, distribution, stockage, tests de validité,...) de données ou d'objets cryptographiques ;

2.3.3 Sécurité d'un chiffrement

Un système de chiffrement est dit sûr si la probabilité d'obtenir une information sur le texte clair ou la clé de déchiffrement à partir du chiffré est presque nulle.

Celà veut dire que :

- toute information qu'on peut tirer du texte clair sera si faible qu'elle ne permettra pas de violer sa confidentialité partielle ou complète ;
- ou que toute information qu'on peut extraire de la clé sera si faible qu'elle ne permettra pas de la reconstituée substentiellement.

Il existe essentiellement deux modèles de sécurité dépendant du système de chiffrement en question :

- **Sécurité inconditionelle :** exhibie des critères suffisants pour la sécurité d'un chiffrement (d'un système symétrique) indépendamment des moyens (calculatoire et de stockage) de l'adversaire. **La modélisation mathématique est basée sur la théroie de l'information formulée par Shanonn**.
- **Sécurité prouvée :** exhibie des critères pour que la sécurité d'un chiffrement (d'un système à clé publique) dépende de la résolution d'un problème mathématique calculatoirement difficile (reconnu comme tel !). **La modélisation mathématique est basée sur la complexité des algorithmes** qui permet de définir la sécurité calculatoire.

Dans le sécurité prouvé du modèle calculatoire, souvent, la dépence entre la sécurité del'algorithme et la difficulté du problème mathématique associé n'est pas une équivalenceet on se contente de modèles plus faibles tels l'**indistinguabilité qui stipule qu'on ne peut distinguer deux chiffrés distincts d'un même texte clair et celà necessite au moins un algorithme probabiliste**.

2.3.4 Taille des données

Un PC à $1GHz = 10^9 Hz$ effectue 10^9 opérations élémentaires par seconde.

Opérations élémentaires : affectation, instructions de contrôle, calcul binaire,...

PUISSANCE DE CALCUL DES MACHINES

Temps	Nbre operations / 1 PC	Nbre operat / 10^{18} PC
1s	10^9	...
1 an	$3,1.10^{16}$	$3,1.10^{34}$
1000 ans	$3,1.10^{19}$	$3,1.10^{37}$
10^9 ans	...	...
15.10^9 ans	$46,5.10^{25}$	$46,5.10^{43}$

- La vitesse de la lumière est de $300000kms/s = 3 \times 10^8 m/s$ donc elle traverse une pièce de 3 métres de largeur en un dix milliardième de seconde. Pendant ce temps, un PC à $1GHz$ peut effectuer 10 opérations élémentaires !.
- Un PC à $1GHz = 10^9 Hz$ effectue 10^9 opérations ou instructions élémentaires par seconde. Combien de temps faut-il pour qu'il puisse casser une clé de taille m par force brute ?
- Pour cela la machine doit tester toutes les 2^m clés possibles ! On suppose que le PC peut tester une clé par instruction c'est à dire 10^9 clés par seconde. Comme $10^9 = 2^{30}$ et $1an = 31536000s \cong 2^{25}s$, alors le PC peut tester 2^{55} clés par an d'où on a le tableau suivant :

SECURITE SUR LA TAILLE DES DONNEES "SECRETS"

taille m	Temps pour 1 PC	Temps pour $10^{18} = 2^{60}$ PCs
56	$2^{26}s = 2ans$	$2^{-59}s$=...
64	$2^{34}s = 2^9 ans = 512ans$	$... = 2^{-51}ans$
128	$2^{98}s = 2^{73}ans$	$2^{13}ans =$ **8192ans**
256	$2^{226} = 2^{201}ans$	$2^{141}ans \gg$ **age Univers**
1024	...	...
2048	$2^{2018} = ...$	$2^{1993} = ...$

NIVEAU DE SECURITE

► La taille des clés, des données chiffrées et des valeurs aléatoires secrètes doivent au moins être de taille **128**

► S'il n'y a pas d'autres attaques à prendre en compte en dehors de l'attaque par force brute, il n'y a pas de raison de choisir des des données sensibles de taille supérieur à **256**

SECURITE DES MOTS DE PASSE

– **Mots de passe de 8 caractères**

Alphabet	Alphabet et chiffre	Alphanumerique
$26^8 \sim 2^{38}$	$36^8 \sim 2^{42}$	$256^8 \sim 2^{64}$

– **Mots de passe de 22 caractères**

Alphabet	Alphabet et chiffre	Alphanumerique
$26^{22} \sim 2^{104}$	$36^{22} \sim 2^{118}$	$256^{22} \sim 2^{176}$

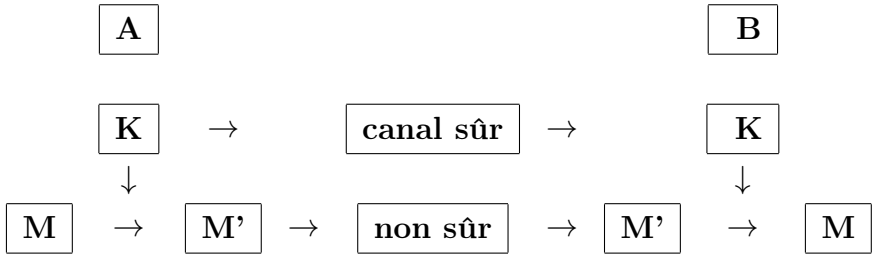
Ainsi :

► Un mot de passe qui protège des données pas trop précieuse doit au moins être de 8 caractères alphanumériques.

► Un mot de passe qui joue le role d'une clé de chiffrement doit au moins etre de 22 caractères alphanumériques.

2.4 CRYPTOGRAPHIE A CLES SECRETES

Un système symétrique est un système construit avec une fonction ou un processus facilement reversible. Généralement on entend par systèmes symétriques les systèmes de chiffrement à clés sécrètes.



Dans un système symétrique, la clé secrète doit être partagée entre les entités en communication d'où la nécessité d'avoir un canal sûr.

Canal sûr :

– Se rencontrer, utiliser la valise diplomatique (avant 1976)

– Utiliser des techniques de cryptographie non symétrique (théoriquement depuis 1976)

Les algorithmes de chiffrements symétriques sont utilisés pour rendre le service de confidentialité et sont composés de deux catégories :

1. les algorithmes de chiffrements par blocs :
   DES, 3DES, Lucifer, FEAL, Blowfish, IDEA, AES

2. les algorithmes de chiffrements par flux :
   RC4, RC5, A5, ORYX, SEAL

2.4.1 Techniques et outils de base

Les systèmes symétriques utilisent plusieurs techniques ou outils dont les plus essentielles sont :

- ▶ **Permutation (Transposition)**
- ▶ **Substitution (Trace d'une permutation globale)**
- ▶ **Chiffrement de Vernam ou One Time Pad (1918)**
- ▶ **Chiffrement par blocs**
- ▶ **Chiffrement sequentiel ou par flux (ou flot)**

Permutation

La permutation utilise généralement :

- la bijection d'un ensemble donnée ;
- le changer l'ordre des symboles dans un messages ;

PRINCIPE

- ▶ Si M est un message et π une permutation, pour caculer la transformé de M par π, on le décompose en morceaux de longueurs égales (**découpage en blocs**) à la longueur de π.
- ▶ Si le dernier morceau est incomplet (longueur plus courte que celle de π) on définit une procédure publique qui permet de faire du padding (= **compléter le bloc incomplet**).
- ▶ On calcule l'image de chaque morceau puis on juxtapose (**concaténation des blocs**) les résultats dans l'ordre du découpage.

NB La concaténation de A et B est noté habituellement $A \parallel B$.

EXEMPLE :

Si M=**Mon premier cours de crypto**" et $\pi = 102538674$,

on découpe $M = \underline{monpremie\ rcoursdec\ ryptozzzz}$

puis on calcule les images des blocs

et enfin on concatène les résultats $\pi(M) = \pi(monpremie)\ \pi(rcoursdec)\ \pi(ryptozzzz)$

$\pi(M) =$**OMNEPEMIR CROSUCDER YRPZTZZZO**

Substitution

PRINCIPE

- ▶ permutation sur l'ensemble des cas possibles appliquées à un nombre fini de cas ;
- ▶ remplacer chaque élément du texte clair (symbole, groupes de symboles) par un autre élément du texte clair (=message) ;

NB Parfois on utilise des substitutions qui ne sont pas reversibles.

EXEMPLE

– S_k = décaler les lettres de k rangs vers la droite dans l'ordre alphabétique".

En numérotant les lettres de l'alphabet latins de 0 à 25 on voit que S_k est la permutation (globale)
$S_k : \frac{\mathbb{Z}}{26\mathbb{Z}} :\to \frac{\mathbb{Z}}{26\mathbb{Z}} : x :\mapsto x + k \mod 26$.
S_k transforme symbole par symbole, on dit que c'est une **substitution monoalphabétique**.

S_3=**chiffrement de CESAR**

– $S_{k_1,k_2,...,k_m} : (\frac{\mathbb{Z}}{26\mathbb{Z}})^m :\to (\frac{\mathbb{Z}}{26\mathbb{Z}})^m : (u_1, ..., u_m) :\mapsto (u + k_1 \mod 26, ..., u_m + k_m \mod 26)$ est une **substitution polyalphabétique**.

Dans ce cas, le message est divisé en blocs de longueur m

Si $S_{k_1,k_2,...,k_m}$ est fixé, le m-uplet $(k_1, ..., k_m)$ est le mot de passe ou clé ;

$S_{k_1,k_2,...,k_m}$= **chiffrement de Vigenaire avec un mot de passe de longueur** m

Opérateur XOR

PRINCIPE

- ▶ $P \oplus Q$ est vrai si P **ou bien** Q est vrai. Si on pose $1 = vrai$ et $0 = faux$, on a : $P \oplus Q = Q \oplus P$, $P \oplus P = 0$ et $(P \oplus Q) \oplus R = P \oplus (Q \oplus R)$
- – on a $0 \oplus 0 = 0$, $0 \oplus 1 = 0$, $1 \oplus 0 = 1$, $1 \oplus 1 = 0$.
- ▶ Le complément de a est $\overline{a} = 1 + a$. Donc $\overline{0} = 1$ et $\overline{1} = 0$: exemple : $\overline{1101011100} = 0010100011$
- – Si on a deux chaines binaires de même longueur a et b on peut les additionner bit à bit, et on retrouve les proprités : $a \oplus b = b \oplus a$, $a \oplus a = 0$, $(a \oplus b) \oplus c = a \oplus (b \oplus c)$ et $(a \oplus b) \oplus b = a$
- ▶ Si M est une chaine binaire $|M|$ représente la longueur de M. Par exemple on a $|011001000110001| = 15$.

Chiffrement de Vernam - Mauborgne - Vigenaire

PRINCIPE

- Chiffrer un message M **suffisamment long** avec une **clé parfaitement aléatoire** K **aussi longue que** M ;
- **Changer de clé à chaque chiffrement** (clé à usage unique= one time pad) ;
- $|M| = |K| \geq 160$; chiffrement : $M \oplus K = M'$; déchiffrement : $M' \oplus K = M$ donc celui qui chiffre et celui qui déchiffre effectue la même opération.

EXEMPLE

texte claire M	10001101011110001101011 1
clé K	01001111010101001111010 1
chiffrement $M' = M \oplus K$	11000010001011000010001 0
clé K	01001111010101001111010 1
Déchifrement $M' \oplus K$	10001101011110001101011 1

Algorithme de chiffrement par blocs (Cipher block)

PRINCIPE

Dans un chiffrement par bloc, le message est divisé en blocs de longueur égale :

- $M = M_1 \oplus M_2 \oplus .... \oplus M_m$, (généralement $l = |M_j| \geq 128$).
- Si $\mathcal{C}_K$ est la fonction de chiffrement, on calcule $M'_j = \mathcal{C}_k(M_j)$ pour tout j.
- Si $\mathcal{D}_K$ est la fonction de déchiffrement, on calcule $\mathcal{D}_k(M'_j) = M_j$ pour tout j
- Il y'a plusieurs façons de combiner les différents blocs chiffrés.

  Par exemple, on peut chiffrer **blocs par blocs** ou **utiliser un bloc chiffré dans le suivant**.

- Généralement, **une fonction de chiffrement par bloc** contient une sous-fonction (principale) qui est itérée plusieurs fois (**chaque itération est appelée tour**) pour créer une **situation de surchiffrement dans le but d'augmenter la sécurité**.
- S'il y'a r tours à réaliser, il faut r sous clés $K_j, 1 \leq j \leq r$ de taille l qui sont dérivées de la clé de chiffrement K, via un **algorithme de génération de sous clés** .
- Si $\overline{\mathcal{C}}_k$ est la sous fonction principale de chiffrement, on calcule par surchiffrement :

  $T' \circ \overline{\mathcal{C}}_{K_r} \circ \overline{\mathcal{C}}_{K_{r-1}} .... \circ \overline{\mathcal{C}}_{K_2} \circ \overline{\mathcal{C}}_{K_1} \circ T(M_j)$ où T et T' sont deux fonctions qu'on applique respectivement au début et à la fin (ce choix T et T', dépend des algorithmes).

DEFINITION : chiffrement produit, chiffrement itératif

- On appelle **chiffrement produit** un chiffrement par blocs qui combine plusieurs trans-

formations élémentairement (**substitutions, transpositions, opérations linéaires ou arithmétiques**)

- Un **chiffrement itératif** résulte de l'application itéré d'un chiffrement (en général un chiffrement produit).

Algorithme de chiffrement par flux (Stream cipher)

PRINCIPE

Le **chiffrement par flux (flot)** se fait sequentiellement en générant une clé aussi longue que le message à chiffrer. Chaque morceau (bit ou byte=octect=8bits) de la clé est composée via la fonction de chiffrement avec la portion de clé correspondante.

- Donc si le message est $M = m_1||m_2||...||m_{l-1}m_l$ et la clé est $K = k_1||k_2||...||k_{l-1}k_l$ alors le chiffrement se fait par morceaux : $c_i = \mathcal{C}_{k_i}(m_i)$ (où $\mathcal{C}_{k_i}$ est la fonction de chiffrement) et le déchiffrement se fait par morceaux : $d_i = \mathcal{D}_{k_i}(c_i) = m_i$ (où $\mathcal{D}_{k_i}$ est la fonction de déchiffrement).
- Par exemple : $c_i = m_i \oplus k_i$ et $d_i = c_i \oplus k_i = m_i$.

– Ainsi, le chiffrement de Vernam est un exemple de chiffrement à la fois de stream cipher et de bloc cipher.

- Le chiffrement par flux est adapté à des modes transmission où le message arrive morceaux par morceaux et si les équipements utilisés ont peu de ressource mémoire ou nécessite une transmission rapide par exemple : chiffrements en ligne , qui sont utilisées en particulier par les armées (sécurité), pour la téléphonie mobile (rapidité) GSM et son réseau (systéme $A51$: algorithme de chiffrement), etc..
- Il y a un autre avantage sur les chiffrements par flux en ce sens que si une erreur se produit sur m_i ou k_i alors cette erreur n'est pas propagée ; elle n'affecte que c_i.
- Toute **la sécurité repose sur l'algorithme de génération de clés** qui est généralement couplé avec le chiffrement.

2.4.2 Avantages et Inconvénients

AVANTAGES

- **Les algorithmes symétriques sont rapides** (parce qu'ils utilisent de petits entiers et des opérations rapides) ;
- En général, il semble que **les algorithmes symétriques sont plus faciles à fabriquer** (plus nombreux !) ;

- Le seul algorithme dont la sécurité est prouvée est un algorithme symétrique à savoir le chiffrement de **Vernam** ;

INCONVENIENTS

- **Confidentialité de la clé sécrète :** problème de partage de la clé à travers un canal sûr et problème de stockage de la clé ;
- **Durée de vie des clés assez courte ;**
- Peut de service de sécurité sont pris en charge par les systèmes symétriques par exemple : on ne peut déterminer qui entre les deux interlocuteurs légitimes, a chiffré un message ;
- **Distribution des clés :** si n personnes communiquent 2 à 2, il faut $C_n^2 = \frac{n(n-1)}{2}$ clés. Pour $n = 1000$ alors $C_n^2 = 499500$.
- En 1973, la **NBS** (National Bureau of Standard) des USA devenu le **NIST** (National Institut of Standard and Technology) a lancé un appel d'offre international pour un algorithme (méthode de chiffrement) donnant lieu à un niveau de sécurité élevé.
- **IBM** proposa **Lucifer** un algorithme développé au début des années 1970, qui fut évalué par la **NSA** (National Security Agency) des USA et publié en 1976 sous le nom de **DES** (Data Encryption Standard). Comme standard $ANSIX3.92$, le DES est proposé en 1974, publié dans le **Federal Register** en 1975, puis adopté comme standard en 1997 (FIPS-46)

2.5 CRYPTOGRAPHIE A CLES PUBLIQUES

2.5.1 Historique

Jusqu'à la fin des années soixante-dix, la cryptologie ne connaissait que les systèmes que l'on appelle maintenant "à clé symétrique". C'est le cas par exemple de l'algorithme DES.

Mais, en novembre 1976, W. Diffie et M.E. Hellman ont émis l'idée de systèmes à clé non-symétrique [?]. Il s'agissait là d'une révolution conceptuelle, dont l'exemple le plus connu est l'algorithme RSA, du nom de ses auteurs Rivest, Shamir et Adleman [?]. Dans ces systèmes, comme le nom l'indique, les clés de chiffrement et de déchiffrement sont différentes. Plus précisément, la connaissance de l'une ne doit pas permettre en pratique de retrouver l'autre.

Ces systèmes sont aussi appelés "à clé publique", une des deux clés pouvant être publiée sans nuire au secret de l'autre. Avec un tel système, n'importe qui peut envoyer à A un message chiffré. Il suffit pour cela d'utiliser la clé publique de A. Seul ce dernier, ayant sa clé privée, aura la capacité de le déchiffrer.

2.5.2 Notions de systèmes non-systèmes

Le système non symétrique peut être résumé comme suit :

- $k \neq k'$ et l'une des clé (soit k) est difficile à calculer à partir de k' ;
- k est alors publiée et est appelée clé publique, k' est gardée secrète par le propriétaire et est appelée clé privée.
- Il n'y a pas de necessité de canal sûr pour échanger les clés mais, il faut que les utilisateurs puissent s'assurer de l'authenticité des clés publiques.
- Ainsi, il est necessaire d'avoir un tiers de confiance (autorite de certification capable de certifier la validité d'une clé publique d'une entité bien identifiée) ou une chaine de confiance comme dans le cas de l'utilisation de GnuPG.
  Comme exemples d'algorithme à clé publique, on a : RSA, Mc-Eliece, El Gamal etc

2.5.3 Schéma d'un système à clé publique

En cryptographie à clé publique, c'est le destinaire du cryptogramme qui crée sa paire de clé publque/privée. Si Bob crée une paire de clé (K^B_{Pub}, K^B_{Pr}), Alice peut lui envoyer un message chiffré suivant le schéma ci après.

2.5.4 Problèmes des systèmes à clé publique

Pour les systèmes à clé publique, le problème du canal sûr de communication ne se pose pas, mais néanmoins deux autres problèmes sont soulevés.

- **Confidentialité de la clé privée** (gardée sur un ordinateur ou une clé USB appelée Token) ;
- **Intégrité de la clé publique** (publiée dans ce qu'on appel un annuaire) ;

Pour garantir l'intégrité de la clé publique de Bob, les différentes parties en communication conviennent de s'en référer à un Tiers de Confiance appelé **Autorité de Confiance** (AC) qui, si elle est sollicitée peut garantir si une clé donnée appartient bien à Bob et si elle n'est pas compromise.

L'utilisation des systèmes à clé publique :

- **à grande échelle**, nécessite des systèmes complexes appelés PKI (Public Key Infrastructure).

- **à petite echelle**, nécessite une chaine de confiance à défaut d'un PKI : c'est le cas de l'Utilisation de Gnu-PG (voir TP) pour le chiffrement des mails sur internet
- Les algorithmes à clé publique utilisent des fonctions à sens unique (voir point suivant) dont la sécurité repose sur l'existence de problèmes difficiles comme la factorisation.
- Tous les algorithmes qui résolvent ces problèmes difficiles sont au moins **sous-exeponentiels** et on montre que les puissances de calculs actuels et les attaques connues imposent de choisir des tailles de clés de plus de 1000 bits (Par exemple le module **RSA** est à 2048 bits actuellement).
- L'utilisation de nombres très grands par les algorithmes à clés publiques ainsi que d'opérations couteuses comme l'exponentiation font ques ces derniers ne sont pas adaptés au chiffrement de données de grande taille.

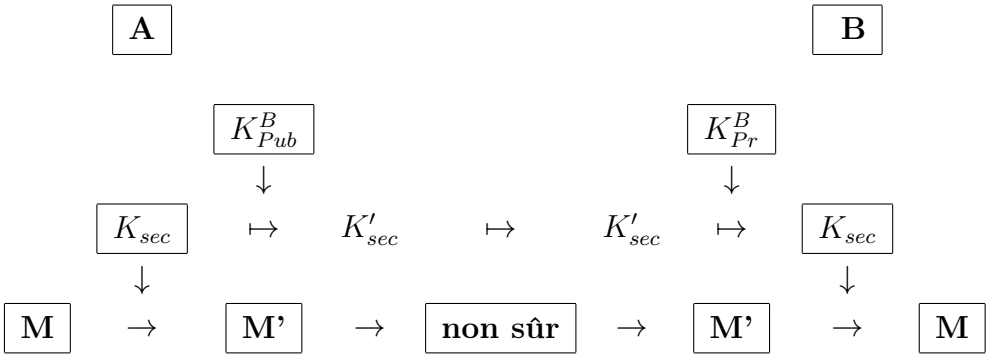
Ainsi les algorithmes à clé publique sont essentiellement utilisés :

- **pour chiffrer des clés symétriques** qui à leur tour seront utilisées pour chiffrer les données pour la **confidentialité** (système hybride : voir point suivant) ;
- **pour signer des messages** pour garantir la **non répudiation** ;
- **pour construire des protocoles (authentification, identification, partage de secrets,...)** ;

2.5.5 Schéma d'un système hybride

Si Alice veut envoyer un message chiffré à Bob, elle génère une clé secrète K_{sec} et l'utilise pour chiffrer le message clair M en M' puis chiffre la clé K_{sec} en K'_{sec} en utilisant la clé publique de Bob K^B_{Pub}. Enfin elle envoie le couple (M', K'_{sec}) à Bob à travers un canal non nécessairement sûr.

A l'arrivée, Bob déchiffre K'_{sec} en K_{sec} en utilisant sa clé privée K^B_{Pr}, puis déchiffre M' en M en utilisant K_{sec}



2.6 PROTOCOLES CRYPTOGRAPHIQUES

2.6.1 Définition

- ▶ De façon générale, un protocole est série finie d'étapes conçu pour accomplir une tâche avec au moins deux partenaires.
- ▶ Un protocole cryptographique est une suite de règles déterminant l'ensemble des opérations cryptographiques nécessaires et leur séquence pour sécuriser une communication (une transaction, un échange de données, ..) entre plusieurs entités.
- ▶ Un protocole cryptographique doit permettre à des personnes qui ne se font pas confiance en général d'échanger en toute sécurité et en présence de personnes malveillantes. Il doit donc empécher l'espionnage et la tricherie sous toutes leurs formes.
- ▶ **Un protocole de cryptographie doit avoir un objectif de securité bien précis : identification, échange de clés, authentification, preuve de connaissance etc.**

2.6.2 Echange de clé Dieffie-Hellman

Pour que Alice et Bob échangent une clé, ils s'entendent d'abord sur un groupe G (noté multiplicative) puis effectuent les étapes suivantes.

- ▶ Alice et bob choisissent un élément g d'ordre premier suffisamment grand dans G ;
- ▶ Alice choisie un parametre secret (valeur aléatoire $a < p$) et calcule g^a dans G et transmet publiquement g^a à Bob ;
- ▶ Bob choisie un parametre secret (valeur aléatoire $b < p$) et calcule g^b dans G et transmet publiquement g^b à Bob ;
- ▶ Chacun deux cacule la valeur commune $k = (g^b)^a = (g^a)^b$ qui constituera leur clé privé ou comme outil pour fabriquer la clé privée.

L'espion Charlie qui suit la communication connait G, H, g, p, g^a et g^b ne doit pas pouvoir calculer $k = (g^b)^a = (g^a)^b$. Ce problème est basé sur le logerithme discret mais reste moins difficle en théorie.

Cette d'échange de clés Diffie-Hellmann a connu depuis lors différentes généralisations.

2.7 FONCTIONS DE HACHAGE

2.7.1 Définition

Une fonction de hachage est une fonction publique $h : \{0,1\}^* \to \{0,1\}^n$ telle que :

- h transforme un message (binaire) de longueur quelconque en un message de longueur fixe ; (**Fonction de Compression**)
- pour tout x, $h(x)$ est facile à calculer ; (**Facilement calculable**)

Les images $h(x)$ sont appelés ***hache*** ou ***empreinte***.

On dira que c'est **fonction de hachage pour la cryptographie** si en plus la fonction a les propriétés suivantes :

- pour presque tout y, il est difficile de trouver x tel que $h(x) = y$; (**Fonction à sens unique sans trappe**)
- Pour presque tout x, il est difficile de trouver x' tel que $h(x) = h(x')$ est faible ; (**Faiblement resistante au collusions**)
- Il est difficile de trouver un couple (x, x') tel que $h(x) = h(x')$ (**Fortement resistante au collusions**).

Pour résumer on retiendra que :

Une fonction de hachage en cryptographie est une fonction publique à sens unique sans trappe (donc facile à calculer et difficile à inverser pour tout le monde) $h : \mathcal{P} \to \mathcal{H}$, **qui transforme un message de longueur quelconque en un message de longueur fixe**.

De plus la probabilité, pour qu'il y ait une collusion doit être faible [c'est-à-dire il doit être difficle de trouver $x \neq x'$ tels que $h(x) = h(x')$] ;

2.7.2 Intégrité

Si x est un message alors pour garantir l'intégrité de x en envoie ou stocke le couple $(x, h(x))$ où $h(x)$ est l'empreinte de x via une fonction de hachage h. Le message est considéré intégre s'il est bien accompagné par son empreinte qu'on peut falsifier.

Les fonctions de hachages comptent deux familles

1. celles utilisant des clés
2. celles n'utilisant pas de clés

Les fonctions de hachage sont utilisés pour :

- l'intégrité
- construire des générateurs aléatoires cryptographiquement sûr ;
- pour la modélisation théorique des fonctions à sens unique tel que le modèle de l'oracle aléatoire.

2.7.3 Algorithemes de hachage

MD : Message Digest,

SHA : Secure Hash Algorithm

SHS : Secure hash Standard

- MD4, Rivest, 1990
- MD5, Rivest, 1992, empreinte sur 128 bits, RFC 1321
- SHA (NIST-1993), FIPS 180, SHA1, empreinte sur 160 bits, FIPS 180-1
- SHS (2001) FIPS 180-2 inclut SHA1 et SHA2 (SHA-256, SHA-384, SHA-512)
- RIPEMD 160

RFC Request For Comment

2.8 SIGNATURE NUMERIQUE

2.8.1 Définition 1

Une signature (digitale-manuelle ou numérique-cryptographique) est un procédé, qui, appliqué à un message, garantit la non répudiation par le signataire et donc réalise les deux objectifs suivants

- identification unique du signataire,
- et preuve d'accord sur le contenu du document.

Elle doit posséder les propriétés suivantes :

- Unique
- Impossible à usurper
- Impossible à répudier par son auteur,
- facile à vérifier par un tiers,
- facile à générer

2.8.2 Modélisation des systèmes de signatures numériques (version1)

Un système de signature est composé d'un quintuplet $(\mathcal{P}, \mathcal{S}, S_{k'}, V_k, \mathcal{K})$ où :

- $\mathcal{P}$ est un ensemble appelé espace des textes clairs ;
- $\mathcal{S}$ est un ensemble appelé espace des signatures ;
- $S_{k'} : \mathcal{P} \to \mathcal{S}$ est une fonction injective dite fonction de signature (non nécessairement bijective) qui dépend d'un paramètre k' appelé clé privée.
- $V_k : \mathcal{P}x\mathcal{S} \to \{vraie, faux\}$ est la fonction de vérification de signature binaire telle que $V_k(m, s) =$ vraie si et seulement si $S_{k'}(m) = s$ (dépendant de la clé publique k) .
- $\mathcal{K}$ l'ensemble des paramètres utilisés est l'espace des clés.

2.8.3 Comparaison des signatures numérique et manuelle

On a vu les 5 points communs entre les signatires manuelle et numérique dans la première définition. Ici on regarde les différences.

Signature manuelle

1. Associé physiquement au document signé ;
2. Identique pour tous les documents venant d'un même signataire ;
3. Habituellment à la dernière page.

Signature numérique

1. Peut être stockée et envoyée indépendamment du document signé ;
2. Fonction du document même si le signataire signe avec la même clé privée
3. Couvre l'entièreté du document (dépend de tout le message)

2.8.4 Exemple de signatures numériques

Une façon simple d'avoir un algorithme de signature est d'utiliser un algorithme de chiffrement à clé publique bijective (comme RSA) :

Pour un chiffrement à clé publique on a :

- $C_k : \mathcal{P} \to \mathcal{C}$ fonction de chiffrement
- $D_{k'} : \mathcal{C} \to \mathcal{P}$ fonction de déchiffrement inverse à gauche de C_k c'est à dire : $D_{k'} \circ C_k(m) = m, \forall m \in \mathcal{P}$).

On peut le tranformer en algo de signature si C_k est aussi l'inverse à gauche de $D_{k'}$ et donc $D_{k'} = C_k^{-1}$.

Dans ce cas, on prend l'algorithme de "déchiffrement" comme étant l'agorithme de signature.

Pour signer m on calcule $D_{k'}(m) = s$ et pour vérifier on calcule $C_k(s)$ et on compare avec m. .

2.8.5 Problème d'intégrité

La signature, telle que présentée jusqu'à présent, ne garantit pas l'intégrité (qu'elle soit manuelle ou numérique).

♦ Dans le cas manuelle c'est l'observation et l'analyse du document qui permet de croire ou de s'assurer que le texte n'a pas été modifié.

♦ Dans le cas numérique, voyons d'abord cet exemple.

- On suppose que la fonction de signature est la bijection réciproque de la fonction de chiffrement : $S_{k'} = D_{k'} = C_k^{-1}$ de Bob,
- On suppose que la fonction de signature est un homomorphisme de semi-group (multiplicatif) : $S_{k'}(m_1 m_2) = S_{k'}(m_1) S_{k'}(m_2)$ c'est-à-dire que la signature du produit est égale au produit des signatures.

► On suppose que Charlie a réussi à faire signer un message m_0 par Bob : $S_{k'}(m_0) = s_0$
Soit k et k' les clés publique et privée respectives de Bob

1. Pour signer m, Bob calcule $S_{k'}(m) = D_{k'}(m) = s$
2. Bob envoie le couple (m, s) à Alice
3. Charlie intercepte (m, s) et calcule $s' = s.s_0$ et $m' = mm_0$.
4. Charlie envoie à Alice (m', s').
5. Alice prend la clé publique de Bob k et calcule $C_k(s')$ et compare avec m'. Mais $C_k(s') = C_k(ss_0) = C_k(s)C_k(s_0) = mm_0 = m'$.
6. **Donc Charlie conclut que le message a été signé par Bob ! Ce qui est faux ! Bob à simplement participer à la signature ! Comme le message a été modifié, il n'est pas intègre (authentique).**

Pour résoudre ce problème, on hache le message avant de le signer pour garantir l'intégrité. La fonction de hachage ne doit pas être homomorphique ! ! !

2.8.6 Définition 2

♦ **Données ajoutées à une unité de données ou transformation cryptographie d'une unité de données, permettant à un destinataire de vérifier la source et l'intégrité de l'unité de données, garantissant la non répudiation et protégeant contre la contrefaçon**

♦ **Une signature numérique doit fournir les services d'authentification (de l'origine des données et de leur intégrité) et de non répudiation (pour le signataire)**

2.8.7 Modélisation des signatures numériques (version 2)

Un système de **signature avec appendice** est composé d'un 6-tuplet $(\mathcal{P}, \mathcal{H}, \mathcal{S}, S_{k'}, V_k, \mathcal{K})$ où :

► $\mathcal{P}$ est un ensemble appelé espace des textes clairs ;
► $\mathcal{S}$ est un ensemble appelé espace des signatures ;
► $h : \mathcal{P} \to \mathcal{H}$ une fonction de hachage,
► $S_{k'} : \mathcal{H} \to \mathcal{S}$ est une fonction injective dite fonction de signature (non nécessairement bijective) qui dépend d'un paramètre k' appelé clé privée.
► $V_k : \mathcal{P}x\mathcal{S} \to \{vraie, faux\}$ est la fonction de vérification de signature binaire telle que $V_k(m, s) =$ vraie si et seulement si $S_{k'}\big(h(m)\big) = s$ (dépendant de la clé publique k).
► $\mathcal{K}$ l'ensemble des paramétres utilisés est l'espace des clés.

Bibliographie

[1] Diffie, W., Hellman, M.E. "*New Directions in Cryptography*". IEEE-IT, 22, n° 6, Nov. 1976.

[2] T. E. Gamal. *A public key cryptosystem and a signature scheme based on discrete logarithms*. IEEE Trans. Inform. Theory, 31 :469-472, 1985.

[3] Rivest, R., Shamir, A., Adleman, L. "*A Method for Obtaining Digital Signatures and Public-Key Cryptosystems*". Communication of the ACM, vol. 21, n° 2, Feb. 1978, pp. 41-54.