

CHI

Ouverte vers l'extérieur → Données sensibles exposées → Besoin de sécuriser
* Sécurité informatique : ensemble des moyens pour minimiser la vulnérabilité contre les menaces

Risque = Vulnérabilité × Menace × Impact

Menaces
accidentelles : Pannes, Erreurs, vol de matériel
Intentionnelles : actions malveillantes

Activer : Modification de l'information
Passiver : Espionnage & Copie - - -

Types de logiciels malveillants

* Virus

- s'attache à un document (hôte), l'infecte, propage à d'autres documents
- Besoin d'une intervention humaine pour se propager.

* Vers (Worm)

- Utilise le réseau pour se reproduire dans plusieurs ordinateurs
- Pas besoin d'une interaction humaine pour se propager.

Objectifs

- Espionner
- Offrir une porte à des pirates
- Détruire des données
- Envoyer de plusieurs requêtes (Surcharge d'un serveur = dénie de service)

Effets

- Ralentissement de la machine
- - - du Réseau
- Plantage du système

* SPAM:

- Courrier électronique envoyé à un très grand nombre de personnes sans leur accord → But publicitaire.

* SCAM:

- SPAM de but d'escroquerie

* Trojan (Cheval de Troie):

- Programme caché un autre programme, s'exécute en même temps.
 - Le programme caché peut ouvrir une porte cachée.
- Consequences: Contrôle de l'extérieur, Espionnage, Perte des données ...

* Ransomware:

- Chiffre les données et demande de l'argent en échange de la clé de déchiffrage.

Comment se protéger?

- Limitez vos droits (Utilisateur SURF)
- Auto Update
- Anti-Virus
- Pare-feu (firewall)

* Niveaux de Sécurité:

- Sécurité physique
- " du personnel
- " des communications
- " des opérations

* Politique de sécurité

ensemble des règles formalisées que les personnes doivent se soumettre

Prohibitives: tout ce qui n'est pas explicitement autorisé est interdit

Permissives: tout ce qui n'est pas explicitement interdit est autorisé

- * Etapes d'établissement d'une politique de sécurité
 - Identification des points faibles / vulnérabilités
 - Evaluation des probabilités des menaces
 - Evaluation du coût d'une intrusion réussie
 - Choix des contre-mesures
 - Evaluation des coûts des contre-mesures
 - Décision

Les services de la sécurité

* Authentification:

- vérifier l'identité pour autoriser l'accès, c'est valider l'entité
 - consiste à vérifier une preuve de son identité
- Simple : 1 élément Forte : au moins 2 Mutuelle : double authentification (Renfort)

* Identification:

- connaît l'identité d'une entité

* Intégrité

- vérifier que les données n'ont pas été altérées accidentellement ou frauduleusement
 - au cours de leurs transmission ou leur stockage.

* Non Réplication:

- empêcher à une personne de暮れ le fait qu'elle a effectué une opération

* Confidentialité:

- assurer qu'une information ne peut pas être lue que par des entités cibles.

Ch TI

- Cryptosystème : Mécanisme assurant les services de la sécurité de l'information
- Cryptographie : Art de concevoir des cryptosystèmes
- Cryptanalyse : Art de casser les cryptosystèmes

Objectifs de la cryptographie :

- Confidentialité : contenu accessible par le destinataire seul (chiffrement)
- Authenticité : Assurer l'identité de l'interlocuteur (Identification / Signature)
- Intégrité : Assurer que le contenu n'a pas été modifié (Hachage)
- Non-repudiation : une entité ne pourra pas nier un acte (Signature)

{ La cryptographie n'est pas la sécurité }

Historique de la cryptographie :

- * Système : méthode du cylindre.
- * Cryptogramme de César : décalage de toutes les lettres par un entier (clé) entre 1 et 26
- * Permutation des lettres : permutations sur 26 lettres , $26!$ clés.
Cryptanalyse correspond à en utilisant la fréquence d'apparition des lettres
- * Chiffrement de Vigenère :
 - $\text{Chiffré}[i] = (\text{Text}[i] + \text{Clé}[i]) \bmod 26$
 - on répète la clé suffisamment de fois
- * Chiffrement de Vernam :
 - chiffrement uniforme
 - jamais utilisé en pratique .

~~SECRET~~

~~SECRET~~

C'est un chiffrement de Vigenère avec les conditions suivantes:

- i) Clé aussi long que le message
- ii) // utilisée une seule fois
- iii) // purement aléatoire.

* Chiffrement de Hill's

- correspondance lettre - nombre $A = 0, B = 1, \dots, Z = 25$

$$\begin{pmatrix} c_1 \\ c_2 \\ \vdots \\ c_p \end{pmatrix} = \begin{pmatrix} a_{11} & \dots & a_{1p} \\ a_{21} & \dots & a_{2p} \\ \vdots & \ddots & \vdots \\ a_{p1} & \dots & a_{pp} \end{pmatrix} \begin{pmatrix} m_1 \\ m_2 \\ \vdots \\ m_p \end{pmatrix} \pmod{26}$$

avec } m_i : lettre du texte clair
 } c_i : lettres du texte chiffré
 } a_{ij} : matrice clé de chiffrement

Rque: Transposition: modifier l'ordre des lettres

1. Substitution: Remplacer/cacher les lettres

Comment renforcer la force des chiffrements?

- Combiner Substitution et Transposition
- Changer les paramètres très souvent.

Principes de Kerckhoff's

Premier cas: un grand nombre de combinaisons ingénieruses repoussé au bout

Second cas : un système avec des conditions g

- i) indéchiffrable ni matériellement ni mathématiquement
 - ii) n'exige pas le secret ni l'omnipotence dans l'ennemi
 - iii) clé communiquée sans notes écrites.

* Cryptographie moderne :

- repose sur l'utilisation de :
 - i) un algorithme public
 - ii) une clé

- les algorithmes sont employés pendant nombreuses années
 - les clés utilisées sont courtes

La sécurité d'un système cryptographique doit reposer sur la clé et pas sur le système lui-même.

* Cryptage à clé symétrique

- la mère de employée pour chiffrer ou déchiffrer ('le virus')

* Cryptage à cle symétrique

- une clé publique comme clé pour le chiffrement (Sécurité des clés)
 - une clé privée pour le déchiffrement (Sécurité du Destinataire)

Chiffrement à clé symétrique

Approche :

Cryptogramme = Fonction (Message, Clé)

Message = Facteur-Inverse (Cryptogramme, clé)

Advantages

- rapid
 - s'applique à un fort débit de données

Inconveniences

- échange des dés (Secret)
 - Gestion des clés (Vinterlocuteur)
 - = $\frac{N \times (N-1)}{2}$ clés

2 types → Blocs DES, IDEA, AES ... (par bloc)

Flots RC4 ... (odit par octet)

* Chiffrement par blocs:

Le message est coupé en blocs de m tailles (64 bits, 128...) qui sont encryptés un par un et finalement concaténés.

! modes d'opérations

ECB: Electronic Code Book (maphient, blocs indép / non sûr)

CBC: Cipher-Block Chaining (plus sûr / Dépendance des blocs)

CFB: Cipher FeedBack (utilise "ou exclusive" entre les blocs successifs et un retour d'init)

OFB: Output FeedBack

* Diagramme de Feistel

voir diapo #8

* D.E.S (Data Encryption Standard)

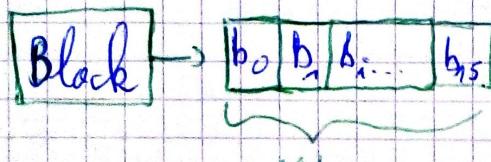
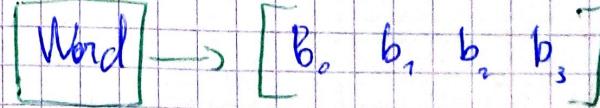
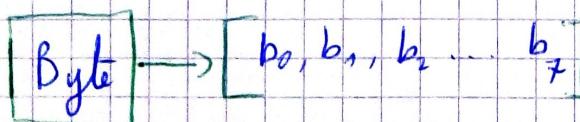
- décompose le texte clair en bloc de 64 bits qui seront chiffré un par un en utilisant une clé de 56 bits

- DES → Faible → Triple DES → obsolète

* AES (Advanced Encryption Standard)

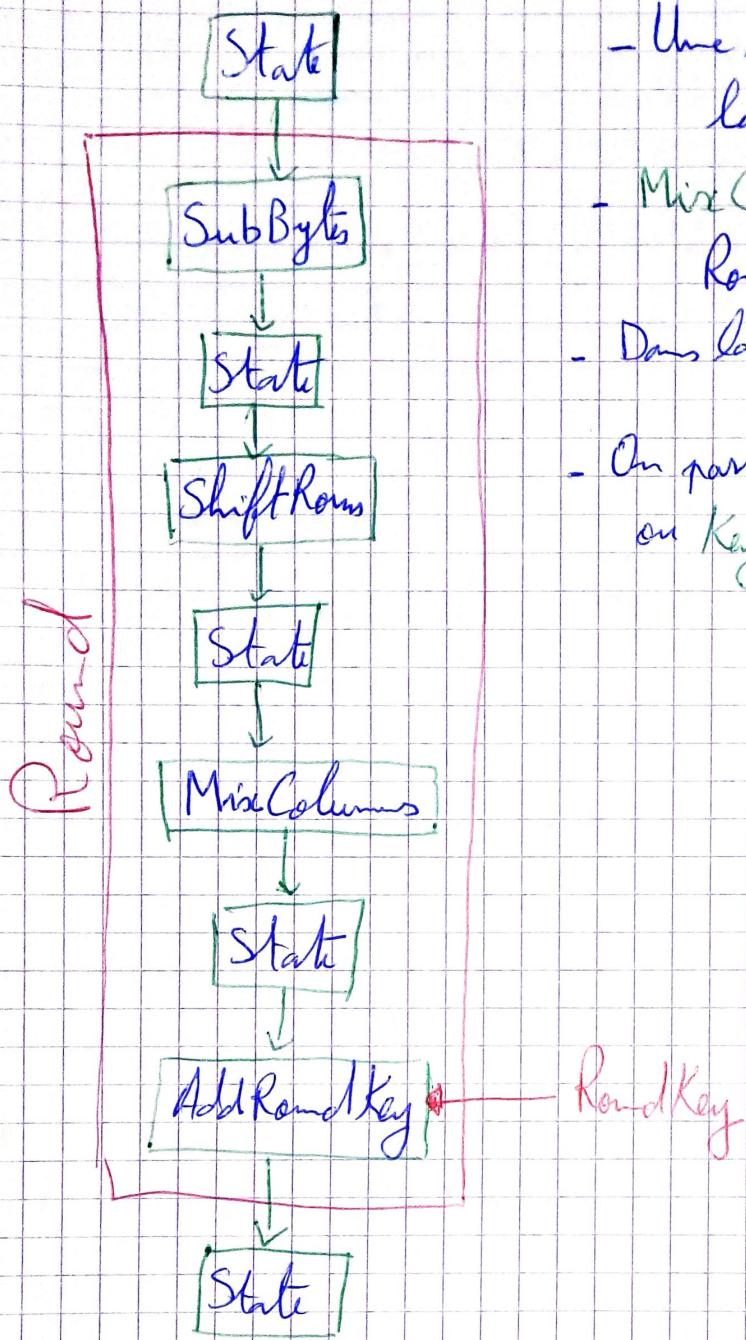
- Blocs de 128 bits et des clés de 128, 192 ou 256 bits en fonction du nombre des tours (10, 12, 14) avec des clés des tours (round Keys) de 128 bits

- les unités utilisées:



S_{00}	S_{01}	S_{02}	S_{03}
S_{10}	S_{11}	S_{12}	S_{13}
S_{20}	S_{21}	S_{22}	S_{23}
S_{30}	S_{31}	S_{32}	S_{33}

Transformations dans les rounds



- Une AddRoundKey est appliquée avant la 1^{ere} Round
- MixColumns est omis dans la dernière Round
- Dans la 1^{ere} Round, RoundKey = Cipher Key
- On passe par une diversification du clé ou Key Expansion pour le Cipher Key

* Transformation SubBytes :

- utilise une table de substitution S-Box et S-Box Inverse.

Format des S-Box et S-Box Inverse

0	1	2	...	F
0	63	7C	77	-
1	CA	82	E9	-
2	B7	FD	93	-
3
4
5
6
7

* Transformation Shift Rows () :

$$S'_{r,c} = S_{r, (c+r) \bmod 4}$$

- on aura dans la 1^{ère} colonne après transformation S'
- on décale les lignes jusqu'à avoir ce résultat

$$\begin{bmatrix} S_{00} & - & - & - \\ S_{11} & - & - & - \\ S_{22} & - & - & + \\ S_{33} & - & - & - \end{bmatrix}$$

* Transformation Mix Columns () :

$$\begin{bmatrix} S'_{0c} \\ S'_{1c} \\ S'_{2c} \\ S'_{3c} \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 03 & 02 \end{bmatrix} \begin{bmatrix} S_{0c} \\ S_{1c} \\ S_{2c} \\ S_{3c} \end{bmatrix} \quad \text{Matrice MixColumns}$$

$$\begin{bmatrix} S_{0c} \\ S_{1c} \\ S_{2c} \\ S_{3c} \end{bmatrix} = \begin{bmatrix} 0E & 0B & 0D & 09 \\ 09 & 0E & 0B & 0D \\ 0D & 09 & 0E & 0B \\ 0B & 0D & 09 & 0E \end{bmatrix} \begin{bmatrix} S'_{0c} \\ S'_{1c} \\ S'_{2c} \\ S'_{3c} \end{bmatrix} \quad \text{Matrice Inv MixColumns}$$

* Transformation AddRoundKey () :

- "ou exclusif" entre State et RoundKey

* Key Expansion :

* Force d'un mot de passe :

- dépend de L longueur du mot de passe
- N nombre de caractères possibles
- Calculé par la formule N^L
- Equivaut à une MDP et clé \mathbf{c}
- un mot de passe sur L caractères avec N caractères possibles est équivalent à une clé de $\log_2(N^L) = L \cdot \log_2(N)$ bits

Chiffrement à Clé Asymétrique :

- Repose sur :
 - i) Fonction Unidirectionnelle
 - ii) Fonction porte-à-vitre.

* Fonction Unidirectionnelle (one way function)

- une fonction $y = f(x)$ tq si on connaît y , il est pratiquement impossible de calculer x (i.e.: calculer $f^{-1}(y)$)

* Fonction porte-à-vitre (Trap Door function)

- une fonction $s = g(y, z)$ tq si on connaît z on peut calculer x à partir de y

* Problème ? trouver le couple de fonctions

* Solutions Clé publique K est le produit de deux grands nombres entiers
Clé privée d est l'un de ces deux nombres entiers

* Nombre premier :

- Ne possède que deux facteurs, 1 et lui-même
- 2 nombres sont premiers entre eux s'ils n'ont pas d'autre facteur commun que 1

$$m^{(p-1)} \bmod p = 1$$

m: n'importe quel nombre
p: nombre premier

$$m^{(p-1)(q-1)} \bmod n = 1$$

$$\begin{array}{l} q \\ n \\ \hline \end{array}$$

$$n = p * q$$

Entre Modifré's

$$m^{(p-1)(q-1)+1} \bmod n = m$$

$$n = p * q$$

- donc il suffit de décomposer cette valeur en deux sous valeurs:

- * une permettant de passer de m à une valeur intermédiaire
- * l'autre permettant de passer de la valeur intermédiaire vers m

Principe de RSA

- Utiliser deux modules, un pour la génération des clés et l'autre pour le chiffrement.

$$\text{- pour les clés } (p-1)(q-1) = \phi(n)$$

$$\text{- pour chiffrer } p * q = n$$

- Utiliser 3 algorithmes:

i) Pour la génération des clés publique et privée

ii) \rightarrow le chiffrement

iii) \rightarrow le déchiffrement.

M le message en clair

C \rightarrow m encrypté

n produit de 2 nombres premiers

(e, n) clé publique (césarée)

(d, n) clé privée (désinariée)

Pour chiffrer un message

$$C = M^e \text{ mod } n$$

Pour Déchiffrer un cryptogramme :

$$M = C^d \text{ mod } n$$

* Construction des clés

- i) prendre 2 nombres premiers p et q donc $n = p * q$
- ii) choisir e qui n'a aucun facteur en commun avec $(p-1)(q-1)$
- iii) Calculer d tq $e * d \text{ mod } (p-1)(q-1) = 1$

enfin clé publique (e, n)

clé privée : (d, n)

* La méthode Indienne (calcul de $V = A^\theta$)

Init $V = 1$

Tq $B > 1$ faire

| Si B est pair : multiplier V par A et retrancher $\frac{1}{2}$ de B

| Sinon : éléver A au carré et diviser B par 2

Si on décompose l'exposant B en binaire :

Init $V = 1$

Pour chaque bit de B (commence par les poids forts)

| éléver V au carré

| Si ce bit = 1

| Multiplier V par A .

* Antithétique Modulaire :

$$(A \text{ mod } B)(C \text{ mod } B) = AC \text{ mod } B$$

$$a^n \text{ mod } m = (a \text{ mod } m)^n \text{ mod } m$$

Rq: Calcul de d par étude d'ordre (diapo 115)

En Résumé:

* Chiffrement asymétrique:

Pratique mais clé grise + Opérations complexes

* Chiffrement symétrique:

Performance, Opérations simples, facile à implémenter mais clé comme secrète (Prob d'échange de clé)

* Chiffrement Hybride (Mixte)

- on chiffre le texte avec une clé secrète (symétrique)

- on chiffre la clé secrète par la clé publique du destinataire

envoie des deux

- on déchiffre la clé secrète par la clé privée du destinataire

- on déchiffre le texte avec la clé secrète déchiffrée

* Protocole d'échange des clés : Diffie-Hellman:

i) choix d'un nombre premier n et d'un générateur g de \mathbb{Z}_n^*

ii) n et g sont publiques

iii) Alice choisit un entier a et envoie $A = g^a \text{ mod } n$ à Bob

iv) Bob // --> B = $g^b \text{ mod } n$ à Alice

v) Alice calcule $K = B^a \text{ mod } n$

vi) Bob calcule $K' = A^b \text{ mod } n$

vii) $K = K'$ est la clé secrète commune. $\{K = K' = g^{ab} = g^{ba}\}$

* El-Gamal:

* Chiffrement ECC

Authentification

* approche traditionnelle

Combinaison d'une identification et d'un mot de passe

* approche évoluée (challenge/réponse)

Alice envoie à Bob un message aléatoire (challenge)

Chiffre à clé secrète

- Bob renvoie à Alice le message chiffré par la clé secrète partagée par les 2
- Alice peut déchiffrer le message chiffré avec la clé secrète. C'est Bob

Chiffre à clé publique

- Bob renvoie à Alice le message chiffré à l'aide de sa clé privée
- Alice peut déchiffrer le message par la clé publique de Bob. C'est Bob

$$\rightarrow \text{Chif}(\text{dechif}(M)) = \text{dechif}(\text{chif}(M))$$

* Signature

- échanger les rôles des clés e et de d

Signature (Secret) : $\text{Sig}_d(m) = \{m\}^d = s$

Vérification : $\text{Ver}_e(m, s) \left\{ \begin{array}{l} \text{Vrai si } \{s\}^e = m \\ \text{Faux si } \{s\}^e \neq m \end{array} \right.$

* Hashage

- calculer un résumé du message aléatoire initial, une empreinte 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 hashage

Message M

$M \in \{0,1\}^*$



Hashé $H(M)$

$H(M) \in \{0,1\}^n$

nde Taille fixe \longrightarrow Le hashé de taille n fixe

Propriété d'une fonction de hachage

- i) publique
- ii) rapide à calculer / facile à calculer
- iii) à sortie de Taille fixe (compression)
- iv) bien répartie en sortie
en terme de sécurité:
- v) Résistance à la préimage : (fonction à sens unique)
 $y = H(m)$ est difficile de retrouver m à partir de y
- vi) Résistance à la recherche de préimage:
 m et $H(m)$ données, difficile de trouver $m' \neq m$ tel que $H(m') = H(m)$
- vii) Résistance aux collisions:
difficile de trouver m et m' vérifiant $m' \neq m$ et $H(m') = H(m)$

Type de fonctions de hachage

- * **MDC** (modification detection code) sans clé pour assurer l'intégrité d'un message
- * **MAC** (message authentication code) avec clé pour vérifier l'intégrité et la provenance du message.

Principe d'une fonction de hachage

- les fonctions de hachage à sens unique sans collision sont construites par itération d'une fonction de compression:
 - * le message M est décomposé en ublocs m_1, m_2, \dots, m_n
 - * une fonction de compression F s'applique à chaque bloc et au résultat de la compression du bloc précédent
 - * L'empreinte $h(M)$ est le résultat de la dernière compression

* Algorithmes

MD2, MD4, MD5 (Message Digest)

↳ empreinte digital de 128 bits

SHA0, SHA1, SHA2, SHA3 (Secure Hash Algorithm)

empreinte de 160 bits

↳ devient le standard SHS

* MD5.

- produit une empreinte de 128 bits
- Manipule le texte par blocs de 512 bits

Fonctionnement: