Sécurité Informatique

Polycopié de Cours & Exercices Corrigés

Zohra Slimane

Mai 2016

(Version révisée le 21 Mai 2016)

Table des matières

CHAPITRE I

Bases de la cryptographie Classique

CHAPITRE II

Cryptographie Moderne

CHAPITRE III

Problèmes potentiels de la Sécurité informatique – Contre Mesures

CHAPITRE IV

Mesures de sécurité basées sur la cryptographie Contrôle d'accès

CHAPITRE V

Pare-feu (Firewall)

CHAPITRE VI

Réseaux VPN et Protocoles de sécurité

REFERENCES

Livres

1. Cryptographie, libertés individuelles et codes secrets. Pierre Vigoureux. Ellipses 2016
2. La cryptologie, L'art des codes secrets. Philippe Guillot. EDP Sciences 2013
3. Initiation à la cryptographie, Cours et exercices corrigés. Gilles Dubertret. Vuibert 2012 (4e édition)
4. Exercices et problèmes de cryptographie , Licence 3, master, écoles d'ingénieurs. Damien Vergnaud. Dunod 2015
5. Cryptographie, théorie et pratique. Douglas Stinson. Vuibert 2003
6. Cryptographie appliquée. Bruce Schneier. Vuibert 2001
7. Cryptographie en pratique. Niels Ferguson et Bruce Schneier. Vuibert 2004
8. Initiation à la cryptographie. Gilles Dubertret. Vuibert 2002
9. Cryptographie et sécurité des systèmes et réseaux. Touradj Ebrahimi, Frank Leprévost et bertrand Warusfel. Lavoisier 2006
10. Sécurité informatique et réseaux. Solange Ghernaouti. Dunod 2016 (5ème édition)
11. Univers secret de la cryptographie.Gilles Dubertret. Vuibert 2015
12. Sécurité informatique, cours et exercices corrigés. Gildas Avoine, Pascal Junod, Philippe Oechslin, Sylvain Pasini. Vuibert 2015 (3e édition)
13. Sécurité informatique, Ethical Hacking. Collectif Eni. Eni 2015 (2e édition)
14. Sécurité et espionnage informatique. Cédric Pernet. Eyrolles 2014
15. La sécurité des réseaux. André Pérez. Iste 2014
16. Hacker's guide. Eric Charton. Pearson 2013 (5e édition)
17. Les VPN, Fonctionnement, mise en oeuvre et maintenance des réseaux privés virtuels. Jean-Paul Archier. Eni 2013 (2e édition)
18. Protocoles de sécurité d'internet. Stéphane Natkin. Dunod 2002
19. Sécurité maximale des systèmes et réseaux. Anonyme. CompusPress 2003
20. Sécurité des architectures web. Quillaume Plouin, Julien soyer et Marc Eric Trioullier. Dunod 2004
21. Data communication and Network security. Houson H. Carr et Charles Sunyder. Mc Graw-Hill Erwin 2007
22. La sécurité sur Internet – Firewalls. D Brent Chapman et Elizabeth D. Zwiday. O'reilly 1996
23. Tableau de bord de la sécurité réseau. Cédric Lorens et Lorent Levier. Eyrolles 2003

NB : Certains passages et images de ce document sont extraits des références suscitées.

24. Détection des intrusions réseaux. Stephen Northcutt, Judy Norvak et Donald McLachlar. CompusPress 2001
25. Guide pratique de sécurité informatique. Bruno Favre et Pierre-Alain Goupille. Dunod 2005
26. Sécurité informatique, Exercices corrigés. Gildas Avoine, Pascal Junod et Philippe Oechslin. Vuibert 2004

Url : Web

1. Handbook of Applied cryptography. Alfred J. Menezes, Paul C. van Oorschot and Scott A. Vanstone
http://cacr.uwaterloo.ca/hac/
2. Principes de Base de la Sécurité de l'Information
http://sesame-mips.unice.fr/~lips/cours/securite-info/1_principes.pdf
3. Cryptographie et services de sécurité
http://www.mi.parisdescartes.fr/~mea/cours/Mi/crypto_synthese.pdf
4. ISO 27004 Information security management Measurements Métrage et métriques d'un SMSI
http://www.hsc.fr/ressources/presentations/clusif-iso27004/clusif-iso27004.pdf
5. Principes de Base de la Sécurité de l'Information
http://www.labri.fr/perso/chaumett/enseignement/administration/masters/master-sdrp/2008-2009/documents-enseignements/2007-2008-philippe-gros-reseaux3.pdf
6. La sécurité des réseaux et des systèmes
http://perso.telecom-paristech.fr/~riguidel/UESecur/Poly-2%20pp.pdf
7. Concepts fondamentaux de la sécurité
http://security.konicaminolta.fr/perch/resources/livret-blanc-pdf-bd.pdf
8. Notions de base en sécurité informatique
http://dfc.cegep-ste-foy.qc.ca/fileadmin/DFC/PlanDeCours/Notions_de_base_en_securite_informatique.pdf
9. Politique Générale de Sécurité
http://www.hsc.fr/conferences/csm05/Vincent-Trely.pdf
10. Norme iso 27001
http://www.27000.org/iso-27001.htm
11. Norme iso 27002
http://www.27000.org/iso-27002.htm
12. Biométrie
https://www.clusif.asso.fr/fr/production/ouvrages/pdf/ControlesAccesBiometrie.pdf
13. Firewalls
http://www-igm.univ-mlv.fr/~duris/NTREZO/20052006/MasquelierMottierPronzato-Firewall-rapport.pdf
14. IPSec
https://tools.ietf.org/html/rfc4301

NB : Certains passages et images de ce document sont extraits des références suscitées.

http://www.mi.parisdescartes.fr/~mea/cours/DU/IPsec_DUsec.pdf

15. SSL
    https://tools.ietf.org/html/rfc6101

    http://www.hit.bme.hu/~buttyan/courses/BMEVIHI4372/ssl.pdf

16. Quelques supports de cours

    http://www.fil.univ-lille1.fr/~wegrzyno/portail/PAC/Doc/Cours/ChiffrementPolyAlphabetique/chiffrementPolyAlphabetique.pdf

    http://www-sop.inria.fr/members/Frederic.Havet/FeteScience/crypto.pdf

    http://bcs.fltr.ucl.ac.be/FE/07/CRYPT/Crypto07-28.html

    http://www.montefiore.ulg.ac.be/~dumont/pdf/Crypto01.pdf

    http://www.montefiore.ulg.ac.be/~dumont/pdf/crypto09-10.pdf

    http://www.eyrolles.com/Informatique/Livre/management-de-la-securite-de-l-information-9782212126976

    https://solinnicolas.wordpress.com/histoire-de-la-cryptographie/

    http://www.bibmath.net/crypto/

    http://anti-cybercriminalite.fr/s%C3%A9curit%C3%A9-informatique

CHAPITRE 1

Bases de la cryptographie Classique

1 Introduction

Depuis l'Antiquité, on cherche à envoyer des messages sans que l'ennemi ne puisse les intercepter. Il existe deux manières complémentaires de faire:
La CRYPTOGRAPHIE: rendre le message incompréhensible par l'ennemi.
La STEGANOGRAPHIE: cacher le message pour que l'ennemi ne le trouve pas.

Il faut reconnaitre aujourd'hui que la généralisation rapide des communications par Internet engendre un besoin impérieux de sécurisation des informations et des technologies associées. D'où le rôle de plus en plus capital du chiffrement.

Pourtant, l'histoire du chiffrement ne date pas d'aujourd'hui, puisqu'il faut remonter à environ 3000 ans avant notre ère, pour en trouver les premières traces.

La cryptographie est une science donc très ancienne. De ce temps-là et au long de l'histoire, la cryptographie a été utilisée exclusivement à des fins militaires. Aujourd'hui, les réseaux informatiques exigent une phase de cryptographie comme mécanisme fondamental afin d'assurer la confidentialité des informations transmises ou stockées.

2 Terminologie et Définitions

Texte en clair : c'est le message à protéger (à chiffrer).
Texte chiffré : (cryptogramme) , c'est le résultat du chiffrement du texte en clair.
Chiffrement : c'est la méthode ou l'algorithme utilisé pour transformer un texte en clair en texte chiffré.
Déchiffrement : c'est la méthode ou l'algorithme utilisé pour transformer un texte chiffré en texte en clair.

Clé : c'est le secret partagé utilisé pour chiffrer le texte en clair en texte chiffré et pour déchiffrer le texte chiffré en texte en clair. On peut parfaitement concevoir un algorithme qui n'utilise pas de clé, dans ce cas c'est l'algorithme lui-même qui constitue la clé, et son principe ne doit donc en aucun cas être dévoilé.

Cryptosystème : algorithmes + clés

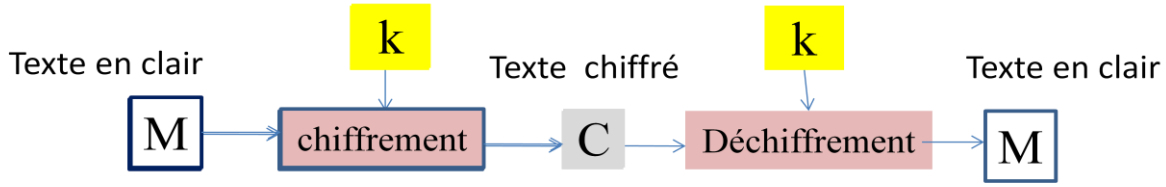


Figure 1.1 - Principe d'un cryptosystème

Cryptographie : cette branche regroupe l'ensemble des méthodes (algorithmes) qui permettent de chiffrer et de déchiffrer un texte en clair afin de le rendre incompréhensible pour quiconque n'est pas en possession de la clé à utiliser pour le déchiffrer.

Cryptanalyse : c'est l'art de révéler les textes en clair qui ont fait l'objet d'un chiffrement sans connaître la clé utilisée pour chiffrer le texte en clair.

Cryptologie : il s'agit de la science qui étudie les communications secrètes. Elle est composée de deux domaines d'étude complémentaires : la cryptographie et la cryptanalyse.

Théorie de la complexité : La cryptologie fait appel à la théorie de la complexité [81,82] pour évaluer la sécurité des schémas cryptographiques, et savoir quelles ressources sont nécessaires pour en mettre en défaut leur sécurité. La théorie de la complexité permet de rendre compte des ressources nécessaires à la résolution d'un problème donné et de classer les problèmes suivant leur difficulté. Les ressources généralement prises en compte sont le temps de calcul, et parfois l'espace mémoire.

Etant donné un algorithme résolvant un problème défini à partir d'une entrée de taille t. La complexité de l'algorithme représente le nombre d'opérations nécessaires pour résoudre le problème en fonction de t. Généralement les complexités sont asymptotiques ($t \to \infty$).
On dit qu'un algorithme est polynomial si sa complexité dans le pire cas est inférieure à un polynôme en la taille des entrées. Les algorithmes polynomiaux s'identifient généralement aux algorithmes efficaces dans la pratique, permettant d'obtenir rapidement un résultat.

Classes importantes de complexité : En général, les algorithmes sont classés dans la théorie de complexité en trois catégories : polynomiale, sous-exponentielle et exponentielle. La fonction $L_t[c, \alpha] = \exp(ct^{\alpha}(logt)^{1-\alpha})$ pour $\alpha \in [0,1]$, par exemple, est :

- polynomiale si $\alpha = 0$,
- exponentielle si $\alpha = 1$,
- sous-exponentielle et super-polynomiale sinon.

Un domaine important de la théorie de la complexité concerne l'évaluation des problèmes de décision, soit l'ensemble des problèmes qui ont pour réponse Oui ou Non. Beaucoup de problèmes dont l'énoncé n'est pas décisionnel peuvent se ramener à des problèmes décisionnels.

La complexité des algorithmes a abouti à une classification de ces problèmes en fonction des performances des meilleurs algorithmes connus qui les résolvent. Il faut noter que cette classification est indépendante des caractéristiques techniques des ordinateurs.

a) **Classe P :** La classe P est l'ensemble des problèmes de décision qui sont solvables par un algorithme à temps polynomial. Autrement dit, un problème de décision est dans P s'il peut être décidé par un algorithme déterministe en un temps polynomial par rapport à la taille de l'instance.

b) **Classe NP :** NP signifie "Nondeterministic Polynomial-time". La classe NP réunit les problèmes de décision pour lesquels la réponse *oui* peut être décidée par un algorithme non-déterministe en un temps polynomial par rapport à la taille de l'instance. Nous avons la conjoncture $P \neq NP$.

c) **Problème NP-Complet :** Un problème décisionnel L est dit NP-complet s'il appartient à la classe NP et si tout problème de la classe NP se réduit en temps polynomial à L. Les problèmes NP-complets sont les problèmes les plus difficiles de la classe NP au sens où ils sont au moins aussi difficiles que tout autre problème dans NP.

Sécurité parfaite ou inconditionnelle : Il y a deux approches fondamentales dans l'étude de la sécurité d'un système cryptographique : la sécurité parfaite ou inconditionnelle, également connue comme la sécurité au sens de la théorie de l'information de Shannon et la sécurité calculatoire.

Dans le cas de la sécurité parfaite, la connaissance du message chiffré n'apporte aucune information sur le message clair : la seule attaque possible est la recherche exhaustive. Ceci mesure la sécurité du système sans borne sur la quantité de calcul que l'attaquant est capable de faire. Un système cryptographique est inconditionnellement sûr s'il ne peut être cassé, même avec une puissance de calcul infinie. Shannon a prouvé qu'une telle sécurité nécessitait que la longueur de la clé soit égale au moins celle du texte chiffré.

Sécurité calculatoire ou algorithmique : La sécurité calculatoire mesure la quantité de calcul nécessaire pour casser un système. On dira qu'un système cryptographique est sûr au sens de la théorie de la complexité si le meilleur algorithme pour le casser est au mieux super-polynomial en temps. La cryptographie moderne abandonne la prétention que l'adversaire dispose de ressources informatiques infinies, et suppose à la place que l'adversaire est est polynomialement borné.

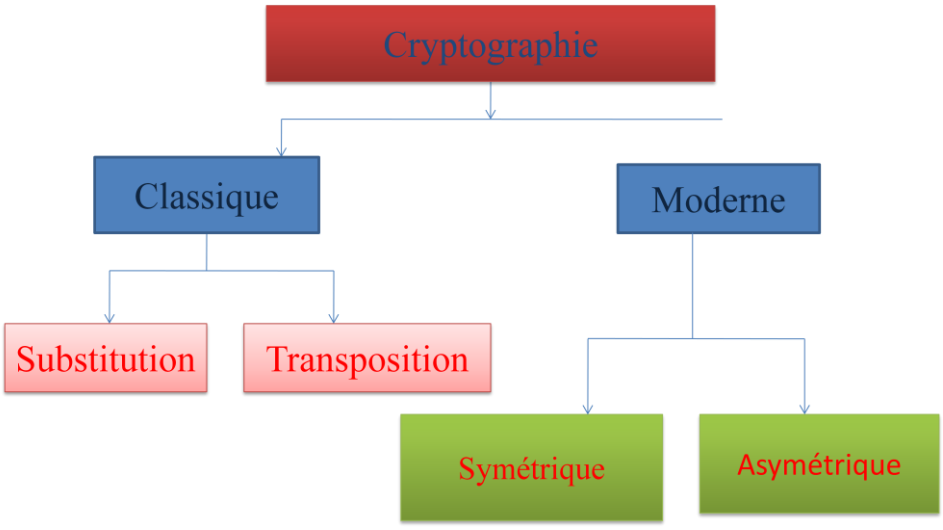


Figure 1.2 – Classification des cryptosystèmes

3 Méthodes de chiffrement classique

L'essentiel des méthodes de chiffrement classique reposent sur deux principes : la substitution et la transposition.
La substitution signifie que l'on remplace certaines lettres par d'autres, ou par des symboles. La transposition signifie qu'on permute les lettres du message afin de le rendre incompréhensible.

3.1 Chiffrement par substitution mon-alphabétique

Le codage par substitution mono-alphabétique est le plus simple à imaginer. Dans le message clair, on remplace chaque lettre par une autre lettre de l'alphabet et toujours la même lettre (mono-alphabétique).

Exemple de tableau de correspondance

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

Exemple :

Chiffrer le message suivant :

MASTER TELECOM

Solution :

UWMNYQ NYIYEDU

Question :
En admettant qu'on ne connait pas la table de correspondance, combien faut-il essayer de combinaisons (de tableaux) pour déchiffrer un message chiffré ?

Solution :
26! =26x25x….x2x1 soit environ 4.10^26 essais, énorme !!!!

Un des problèmes avec le code par substitution est de se souvenir de la clé, c'est-à-dire la nouvelle permutation de l'alphabet employée. Il n'est en effet pas facile de se souvenir de 26 lettres dans un ordre quelconque. C'est pourquoi il existe des variantes :

- Le chiffrement de César, fondé sur un simple décalage de **k** lettres.
- Le chiffrement AtBash, qui consiste simplement à écrire l'alphabet en sens contraire .
- Le chiffrement ROT13 (décalage de **k=13** lettres).
- Le chiffrement par substitution mon-alphabétique à clé.

3.2 Chiffrement par substitution mono-alphabétique de César

Le chiffrement de César est une des plus simples méthodes de chiffrement connues. Chaque lettre du texte en clair est remplacée par une autre lettre a distance fixe dans l'alphabet. Par exemple, si l'on utilise un décalage de 3, A serait remplace par D, B deviendrait E, et ainsi de suite.

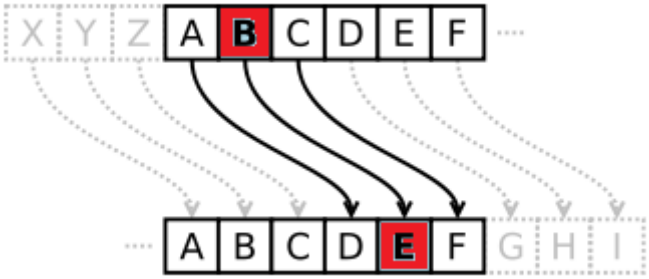


Figure 1.3 - Principe du chiffrement de César

Principe : décaler les lettres de l'alphabet de **k**
Chiffrement : lettre codée = (lettre claire + **k**) mod 26
Déchiffrement : lettre claire = (lettre codée - **k**) mod 26

Exemple :
En utilisant le code de César avec **k**=3, chiffrer le message suivant :
MASTER TELECOM
Solution:
PDVWHU WHOHFRP

Question :
En admettant qu'on ne connait pas la valeur du décalage, combien faut-il essayer de combinaisons (de tableaux) pour déchiffrer un message chiffré ?
Solution :
25 essais seulement !!! Le chiffrement de César est vulnérable à l'attaque par force brute (essai de toutes les combinaisons possibles)

3.3 Chiffrement par substitution mono-alphabétique à clé

L'intérêt de cette méthode est que : pour que le cryptogramme soit déchiffré, il suffit de faire parvenir au destinataire une clé qui peut être un simple mot. Comme il existe des millions de mots clé possibles, il existe des millions de combinaisons possibles.

PRINCIPE

Étape 1 : Choisir un mot clé. Ex : informatique
Étape 2 : Le "nettoyer" en enlevant tout les doubles et les accents :
résultat : INFORMATQUE
Étape 3 : Reporter ce mot dans le tableau de correspondance

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

Étape 4 : Compléter l'alphabet

- En prenant soin de ne pas utiliser 2 fois une lettre
- En partant de la dernière lettre écrite

Exercice : Compléter le tableau

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

Exercice :

Déchiffrez le cryptogramme mono-alphabétique suivant, sachant que la clé est: Il était une fois :
HAWAEZXXIGHAD

Solution :

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

TELECOMMASTER

3.4 Cryptanalyse par étude des fréquences du chiffrement mono-alphabétique

Par sa simplicité et par sa force, la substitution mono-alphabétique a dominé la technique des écritures secrètes pendant tout le premier millénaire. Elle a résisté aux cryptanalystes jusqu'à ce que le savant arabe Abou Yousouf al-Kindi (801-873) mette au point une technique appelée analyse des fréquences.
La principale faiblesse du chiffrement mono-alphabétique est qu'une même lettre est toujours chiffrée de la même façon. Dans les textes longs, les lettres n'apparaissent pas avec la même fréquence. Ces fréquences varient suivant la langue utilisée. En français, les lettres les plus rencontrées sont dans l'ordre :
E S A I N T R U L O D C P M V Q G F H B X J Y Z K W
avec les fréquences (souvent proches et dépendant de l'échantillon utilisé) :

.

E	S	A	I	N	T	R	U	L	O	D
14.69%	8.01%	7.54%	7.18%	6.89%	6.88%	6.49%	6.12%	5.63%	5.29%	3.66%

Méthode de cryptanalyse : dans le texte chiffré, on cherche la lettre qui apparaît le plus, et si le texte est assez long cela devrait être le chiffrement du E, la lettre qui apparaît ensuite dans l'étude des fréquences devrait être le chiffrement du S, puis le chiffrement du A... On obtient des morceaux de texte clair sous la forme d'un texte à trous et il faut ensuite deviner les lettres manquantes.
Par exemple, déchiffrons la phrase :
LHLZ HFQ BC HFFPZ WH YOUPFH MUPZH
On compte les apparitions des lettres :
H : 6 F : 4 P : 3 Z : 3
On suppose donc que le H crypte la lettre E, le F la lettre S, ce qui donne
*E** ES* ** ESS** *E ***SE ****E
LHLZ HFQ BC HFFPZ WH YOUPFH MUPZH
D'après les statistiques :

E	S	A	I	N	T	R	U	L	O	D
14.69%	8.01%	7.54%	7.18%	6.89%	6.88%	6.49%	6.12%	5.63%	5.29%	3.66%

P et Z devraient se décrypter en A et I (ou I et A). Le quatrième mot
"HFFPZ", pour l'instant décrypté en "ESS**", se complète donc en "ESSAI" ou "ESSIA". La première solution semble correcte ! Ainsi P crypte A, et Z crypte I.
La phrase est maintenant :
*E*I ES* ** ESSAI *E ***ASE **AIE
En réfléchissant un petit peu, on décrypte le message :
CECI EST UN ESSAI DE PHRASE VRAIE

3.5 Chiffrement par substitution poly-alphabétique

Contrairement au chiffrement mono-alphabétique qui se contente d'utiliser la même lettre de substitution, un chiffrement de substitution poly-alphabetique permet de remplacer une lettre par une autre lettre qui n'est pas toujours la même.
L'exemple le plus fameux de chiffre poly-alphabétique est sans doute le chiffrement Vigenère (1586)

a) Le chiffrement Vigenère (1586)

L'idée de Vigenère est d'utiliser un chiffre de César, mais où le décalage utilisé change de lettres en lettres. Pour cela, on utilise une table composée de 26 alphabets, écrits dans l'ordre, mais où chaque ligne est décalée à gauche d'un caractère. (Carré de Vigenère)
On écrit encore en haut, horizontalement, un alphabet complet, pour la clé, et à gauche, verticalement, un dernier alphabet, pour le texte à coder :

Lettre de la clé

Lettre du texte clair

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

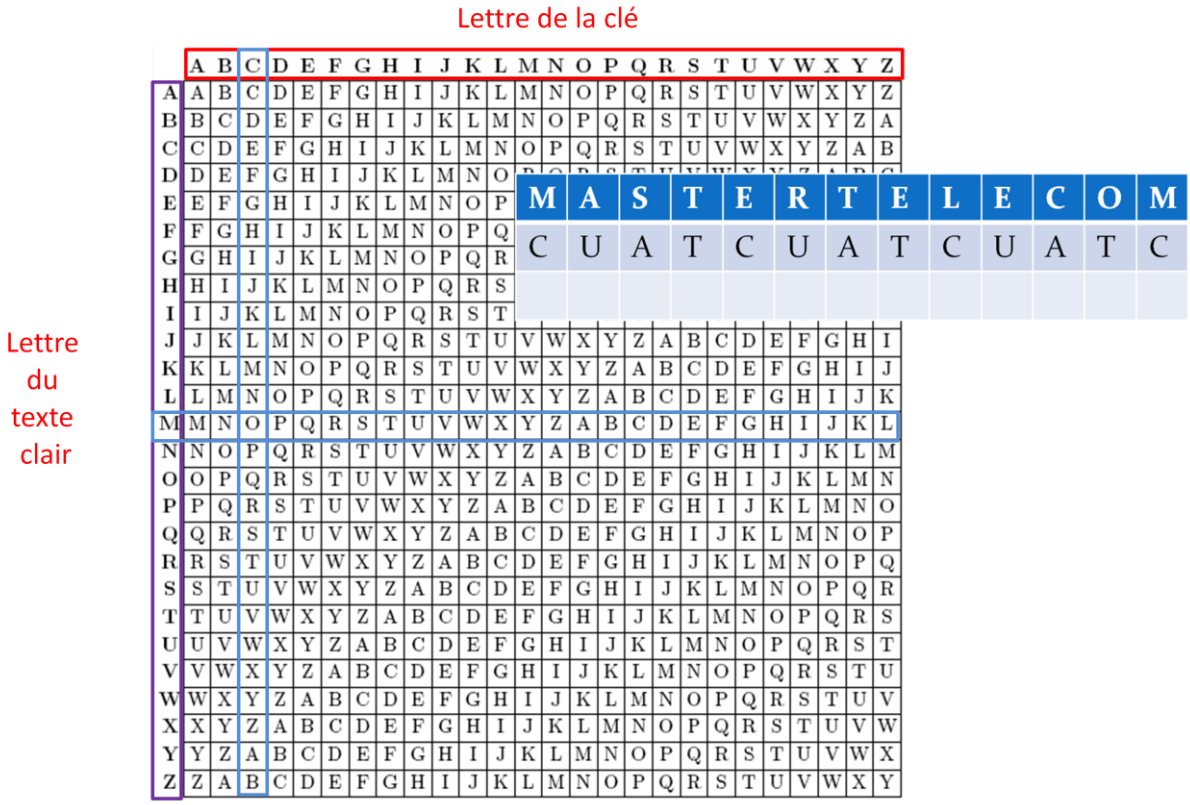
Figure 1.4 - Carré de Vigenère

Pour coder un message:
- on choisit une clé qui sera un mot de longueur arbitraire
- on écrit ensuite cette clé sous le message à coder, en la répétant aussi souvent que nécessaire pour que sous chaque lettre du message à coder, on trouve une lettre de la clé.

Pour coder, on regarde dans le tableau l'intersection de la ligne de la lettre à coder avec la colonne de la lettre de la clé.

Exemple : On veut coder le texte : MASTER **TELECOM,** avec la clé : CUAT

Texte clair	**M**	**A**	**S**	**T**	**E**	**R**	**T**	**E**	**L**	**E**	**C**	**O**	**M**
Clé	C	U	A	T	C	U	A	T	C	U	A	T	C
Texte chiffré													



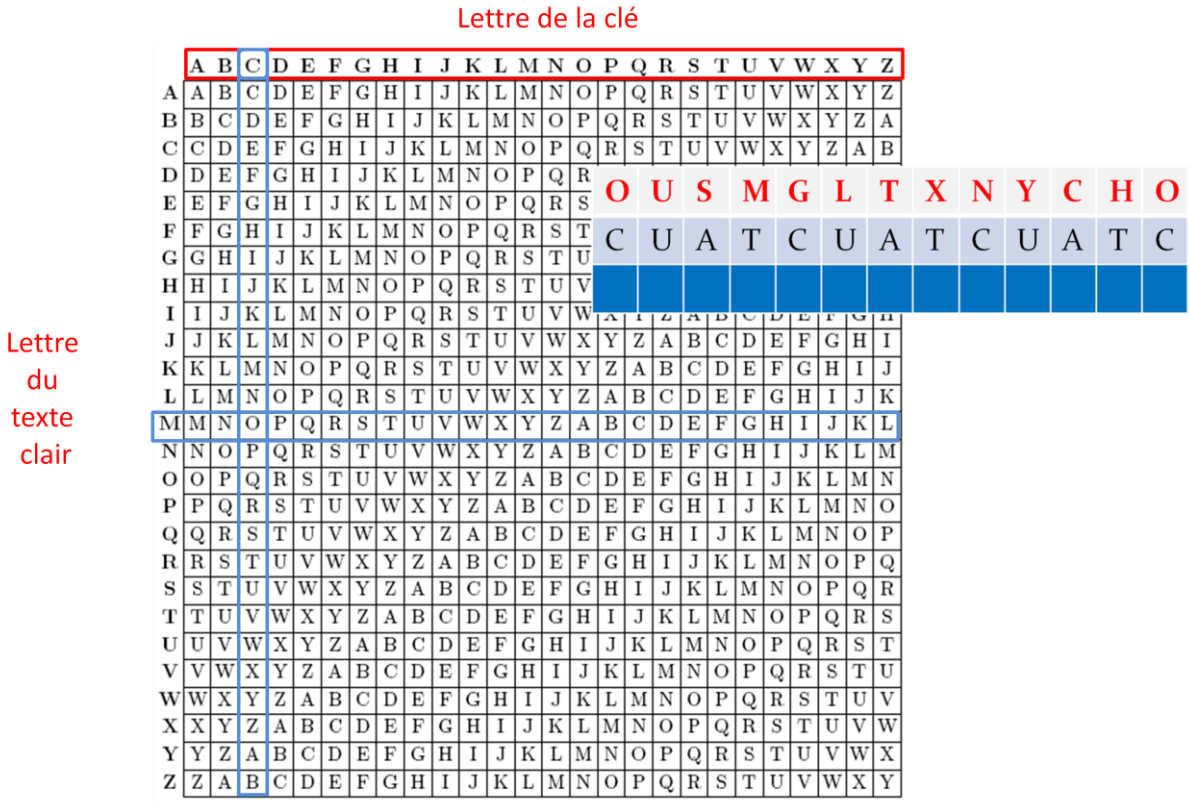
Exemple : On veut coder le texte : MASTER **TELECOM,** avec la clé : CUAT

Texte clair	M	A	S	T	E	R	T	E	L	E	C	O	M
Clé	C	U	A	T	C	U	A	T	C	U	A	T	C
Texte chiffré	O	U	S	M	G	L	T	X	N	Y	C	H	O

Constatations ? Remarquer par exemple les 3 lettres E du texte en clair
Le déchiffrement d'un texte chiffré, est très facile, à condition de connaître la clé qui a été utilisée pour chiffrer.
On introduit à nouveau le cryptogramme (texte chiffré) ainsi que la clé dans un tableau comme suit :

Texte chiffré	O	U	S	M	G	L	T	X	N	Y	C	H	O
Clé	C	U	A	T	C	U	A	T	C	U	A	T	C
Texte déchiffré													

Pour retrouver une lettre du message en clair :
- On regarde la colonne associée à la lettre de la clé correspondante ;
- Dans cette colonne on retrouve la lettre chiffrée;
- La lettre du texte en clair n'est autre que la lettre correspondant à la ligne de cette lettre chiffrée



Exercice 1:
Déchiffrer le message suivant : vvqimsykttzbx
Chiffrement : Vigenère , clé: test

Exercice 2:
On donne le message en clair et son chiffré par le chiffrement Vigenère:

- message en clair : MASTER TELECOM
- message chiffré : AKGDSBHOZOQYA

Trouver la clé.

b) Le chiffrement Venam (1917)

Le chiffrement de Venam appelé également masque jetable (One Time Pad) est défini comme un chiffrement de Vigenère avec la caractéristique que la clé de chiffrement a la même longueur que le message clair.
Pour utiliser ce chiffrement, il faut respecter plusieurs propriétés :
– choisir une clé aussi longue que le texte à chiffrer,
– utiliser une clé formée d'une suite de caractères aléatoires,
– protéger la clé,
– ne jamais réutiliser une clé.

Texte clair	M	A	S	T	E	R	T	E	L	E	C	O	M
Clé	A	Z	E	R	X	T	P	S	S	Y	U	E	Z
Texte chiffré													

Le système du masque jetable, avec les précautions indiquées ci-dessus, est absolument inviolable si l'on ne connaît pas la clef. Il est couramment utilisé de nos jours par les États. En effet, ceux-ci peuvent communiquer les clés à leurs ambassades de manière sûre via la valise diplomatique.
Le problème de ce système est de communiquer les clés de chiffrement ou de trouver un algorithme de génération de clé commun aux deux partenaires.
De plus, la création de grandes quantités des clés aléatoires devient vite problématique. N'importe quel système couramment utilisé pourrait exiger des millions de caractères aléatoires de façon régulière.
La distribution des clés est également complexe. La longueur de la clé étant égale à celle du message, une bonne organisation est nécessaire.

c) Cryptanalyse du chiffrement de Vigenère

La cryptanalyse du chiffrement de Vigenère peut se faire, si le message est assez long, à texte chiffré connu, en remarquant que des répétitions de lettres assez longues doivent correspondre dans le texte clair à des répétitions de lettres aussi. Ceci permet de majorer la taille de la clé. On se ramène alors à la cryptanalyse d'un chiffrement de César.
Exemple:
THOIL BYPZZ **F**LNFU EKFRP **F**LQQH E**H**EJZ ODSYF **T**V
Ici, on trouve les répétitions T, H et F éloignés respectivement de 35 lettres, 25 lettres et 10 lettres.
Taille (clé)=PGCD (35, 25, 10)=5

Une fois que l'on a déterminé la longueur de la clef, le décodage est le même que celui de 5 codes de César (Cryptanalyse par étude des fréquences).

3.6 Chiffrement par transposition (permutation)

Le chiffrement par transposition ou permutation consiste à changer uniquement l'ordre des lettres sans faire de substitution.
Il existe plusieurs méthodes de transposition, citons :

a) La méthode Zig Zag:

Écrire sur 2 ou plusieurs lignes
Lire ligne par ligne

Exemple: pour chiffrer le message « MASTERTELECOM » avec une profondeur égal à 2 (2 lignes), on l'écrit sous la forme:
M S E T L C M
A T R E E O
Message chiffré : MSETLCMTREEO

Exercice
Déchiffrer le message :
« SUCLIUVYGVRAEOGHMNUMNAOPYELEUQIOAEERLLNCEIQIEESNAS »
Profondeur = 2
SEUL CELUI QUI VOYAGE VERRA LE LONG CHEMIN QUI MENE A SON PAYS

b) Méthode de la grille avec clé

La transposition à grille consiste à écrire le message dans une grille rectangulaire ligne par ligne, puis lire colonne par colonne dans l'ordre défini par la clé.

Exemple : clé = TEST (4 lettres -----> 4 colonnes)

Texte en clair : MASTERTELECOM

M	A	S	T
E	R	T	E
L	E	C	O
M			

On lit colonne par colonne, dans l'ordre défini par la clé :

T	E	S	T
3	1	2	4

Il faut prendre l'ordre alphabétique

On obtient :
Texte chiffré : ARESTCMELMTEO

Exercice :
1. Chiffrer le message : « seul celui qui voyage verra le long chemin qui mène à son pays »

En utilisant la transposition avec la clé : désert

SLVVEHUAYEUOELEISSLQARNIENCUGAGNNPUIYROMMOEIELCQEA

2. Déchiffrer le message chiffré suivant : « AGMBHNDJEKCIFL »

Avec la même clé.

ABCDEFGHIJKLMN

3.7 Chiffrement par substitution et transposition

Exemple : le chiffrement allemand **ADFGVX (1918)**
Le chiffrement ADFGVX repose sur une substitution de type carré, suivie d'une transposition.
Pour réaliser la substitution, les 26 lettres de l'alphabet et les 10 chiffres sont rangés dans un tableau 6×6, aux extrémités desquelles on a ajouté les lettres ADFGVX

	A	D	F	G	V	X
A	Q	Y	A	L	S	E
D	Z	X	R	C	H	0
F	F	O	4	M	8	7
G	3	I	T	G	U	K
V	P	D	6	2	N	V
X	1	5	J	9	W	B

Figure 1.5 – Tableau de substitution ADFGVX

Principe du chiffrement **ADFGVX** :
Chaque lettre est codée par le couple de lettres qui correspond à sa ligne et à sa colonne. Ainsi, M est codé FG, et 0 par DX
On choisit ensuite, pour faire la transposition, une clé. On écrit le texte intermédiaire sous ce mot, puis on réordonne les colonnes par ordre alphabétique croissant. Il ne reste plus qu'à relire le tableau de gauche à droite, et de haut en bas.

Exemple
Chiffrer le message : RENFORT COMPIEGNE 16H10
Message en clair : RENFORT COMPIEGNE 16H10

Message chiffré par substitution :
DFAXV VFAFD DFGFD DFDFG VAGDA XGGVV AXXAV
FDVXA DX

Message chiffré par transposition (clé : DEMAIN)
XDFVA VDFAD FFDGF FDGDV AAGXV GGAVX FXADV VXXAD

D	E	M	A	I	N
D	F	A	X	V	V
F	A	F	D	D	F
G	F	D	D	F	G
V	A	G	D	A	X
G	G	V	V	A	X
X	A	V	F	D	V
X	A	D	X		

A	D	E	I	M	N
X	D	F	V	A	V
D	F	A	D	F	F
D	G	F	F	D	G
D	V	A	A	G	X
V	G	G	A	V	X
F	X	A	D	V	V
X	X	A		D	

Exercice :
Déchiffrer le message chiffré suivant : (clé : PRINCE)
XXFD GFDF FDAX GFFD VVXX FDGF AX
TO BE OR NOT TO BE

4 Stéganographie

La stéganographie est l'art de la dissimulation. L'objet de la stéganographie n'est pas de rendre l'information à transmettre incompréhensible (chiffrée) mais de la faire passer inaperçue, c'est-à-dire cacher un message dans un autre. L'idée est de prendre un message et de le modifier de manière aussi discrète que possible afin d'y dissimuler l'information à transmettre.

Aujourd'hui, il est possible de cacher des informations dans plusieurs types de fichiers couramment échangés sur des réseaux tel les images, la vidéo ou des textes.

Exemple : Usage des bits de poids faible d'une image

Le message original est ici une image. La technique de base (dite LSB pour Least Significant Bit) consiste à modifier le bit de poids faible des pixels codant l'image :

une image numérique est une suite de pixels, dont on code la couleur à l'aide de 3 d'octets. Par exemple pour une couleur RGB sur 24 bits, chaque octet indique l'intensité de la couleur correspondante (rouge, vert ou bleu) par un niveau parmi 256. Passer d'un niveau n au niveau immédiatement supérieur (n+1) ou inferieur (n-1) ne modifie que peu la teinte du pixel, or c'est ce que l'on fait en modifiant le bit de poids faible de l'octet.

Considérons par exemple 2 pixels successifs d'une image :

Chaque pixel est représente par 3 nombres codes sur 8 bits : R représente l'intensité du rouge (un entier entre 0 et 255), G celle du vert, B celle du bleu. Si l'on modifie les 2 bits de droite de R, on modifie très peu sa valeur, et cela est imperceptible à l'œil humain.

On remplace alors les 2 bits de droite de R par les 2 premiers bits du message. Puis on continue pour les composantes G, B, puis pour le 2ème pixel, etc...

Il est impossible, à l'œil, de distinguer l'image qui cache le message, et l'image initiale.

Image initiale	R1=01001110 R2=01110011 ⋮	G1=01101111 G2=01110110 ⋮	B1=11111111 B2=10101010 ⋮
Message	101100011011		
Image qui cache le message	R1=01001110 R2=01110001 ⋮	G1=01101111 G2=01110110 ⋮	B1=11111100 B2=10101011 ⋮

CHAPITRE 2

Cryptographie Moderne

Dans les télécommunications modernes, l'information est codée en binaire. Donc, contrairement à la cryptographie classique, la cryptographie moderne manipule des séquences binaires (le message à chiffrer est une suite de bits).
Nous distinguons 2 grandes classes dans la cryptographie moderne :

La Cryptographie symétrique

Dans la cryptographie symétrique, la clé de chiffrement est la même que la clé de déchiffrement. De ce fait, la clé doit être un secret partagé uniquement entre l'émetteur et le destinataire. Il existe plusieurs algorithmes qui fonctionnent sur ce principe : DES, RC4, RC5, Blowfish, IDEA, AES, ….

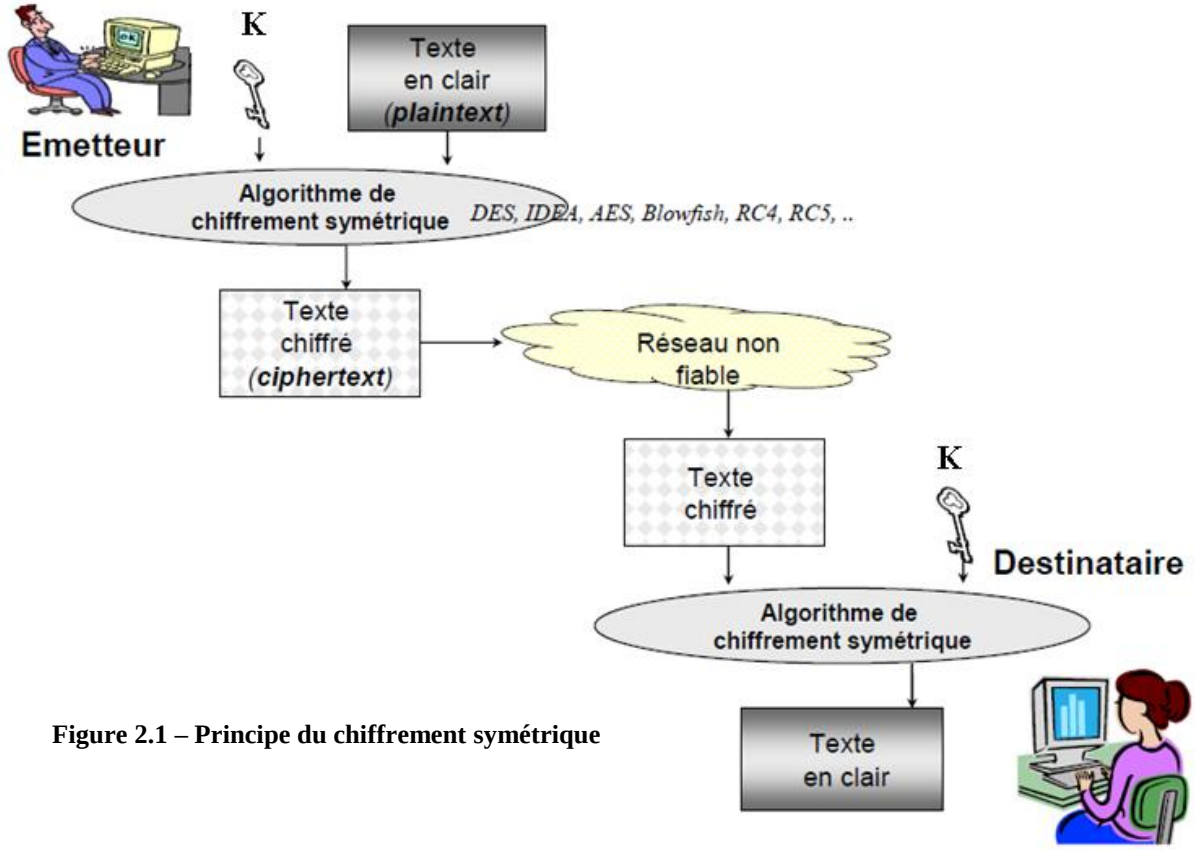


Figure 2.1 – Principe du chiffrement symétrique

La Cryptographie asymétrique

Dans la cryptographie asymétrique, la clé de chiffrement n'est pas la même que la clé de déchiffrement. Les algorithmes les plus connus sont : RSA, Diffie-Hellman, El Gamal

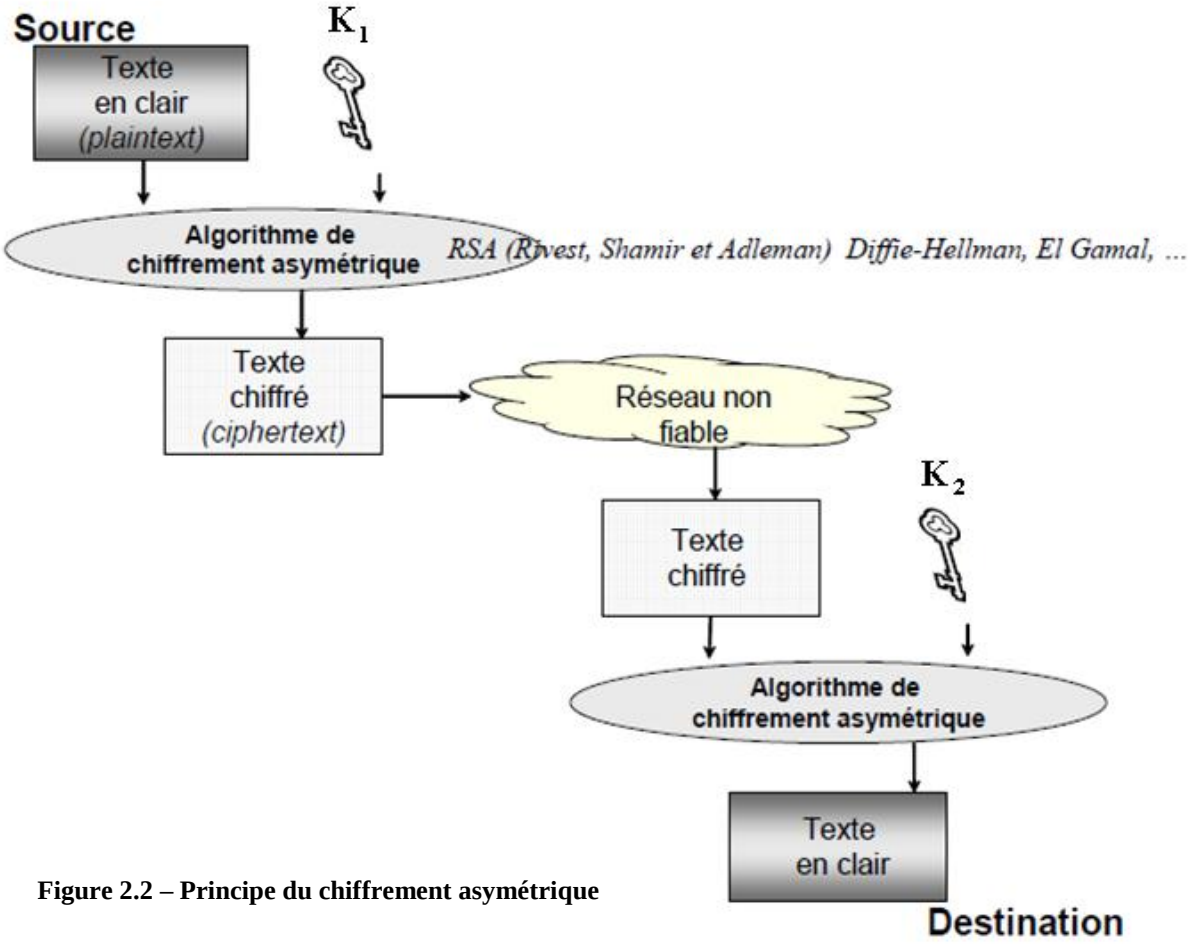


Figure 2.2 – Principe du chiffrement asymétrique

1. Cryptographie symétrique

La cryptographie symétrique utilise deux types de chiffrement manipulant des mots binaires :

- Le chiffrement par bloc
- Le chiffrement par flot

1.1 Le chiffrement par bloc

Le chiffrement par bloc consiste en premier lieu à découper le message à chiffrer en bloc de taille fixe (exemple : 64 bits). Puis appliquer l'algorithme de chiffrement sur chaque bloc.
On découpe le message M de n bits en s blocs de l = n/s bits (on ajuste initialement la taille du message en ajoutant des caractères sans signification afin que sa taille soit un multiple de l). Un algorithme de chiffrement par blocs opère sur des blocs de l bits, pour produire en général un bloc de l bits afin d'assurer la bijectivité du code.

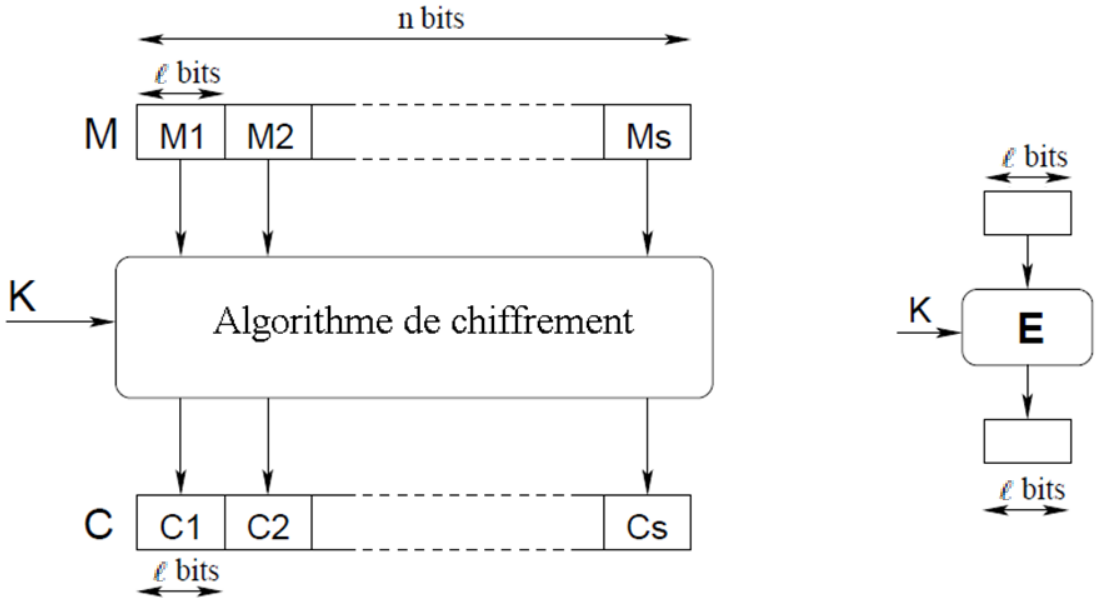


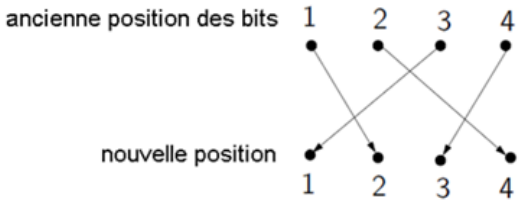
Figure 2.3 – Principe du chiffrement par bloc

L'algorithme de chiffrement est en général une succession des opérations suivantes:

- Permutation
- Substitution
- Opération XOR (Ou Exclusif)

Dans ce qui suit, nous allons examiner un exemple pour chaque opération

Exemple de chiffrement par permutation binaire
On considère M=1010001001011
On représentera la permutation P soit avec un tableau
[P(1),P(2),P(3),….. ,P(l)] l : taille d'un bloc
Ici nous prenons P : {1, 2, 3,4} → {1, 2, 3,4} donné par
[P(1), P(2), P(3),P(4)] = [3, 1, 4,2]



On décompose M en bloc de 4 bits et on permute chaque bloc avec P

Exemple de chiffrement par substitution binaire
On considère M=10100010000 , et la substitution S suivante

X	00	01	10	11
↓	↓	↓	↓	↓
$S(X)$	11	10	01	00

On décompose M en bloc de 2bits et on applique S à chacun des blocs

M=	10	10	01	00	00
C=	01	01	10	11	11

Ou exclusif ⊕
On considère l'alphabet binaire $\sum = \{0,1\}$.
On notera par la suite ⊕ l'opérateur logique ou exclusif

X ⊕ Y

X\Y	0	1
0	0	1
1	1	0

On étendra la notation ⊕ à des blocs de bit où l'on effectue le ou exclusif bit à bit
[0, 1, 0, 1,1]⊕ [1, 1, 0, 0,1]= [1, 0, 0, 1,0].

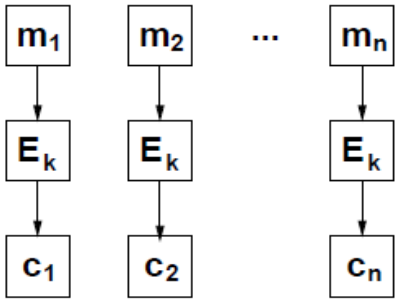
1.1.1 Modes d'opérations

On distingue plusieurs modes d'opérations utilisés dans le chiffrement par bloc, les plus courants sont les suivants :

a) Le mode Electronic Code Book (ECB)

Dans ce mode, le message M est découpé en blocs ***mi*** de taille fixe.
Chaque bloc est alors chiffré séparément par une fonction ***Ek***, paramétrée par une clé ***k*** . Ainsi un bloc de message donné ***mi*** sera toujours codé de la même manière. Ce mode de chiffrement est le plus simple mais il est très vulnérable aux attaques.

$$c_i = E_k(m_i)$$



Le déchiffrement nécessite l'inverse de la fonction de codage: $D_k = E_k^{-1}$

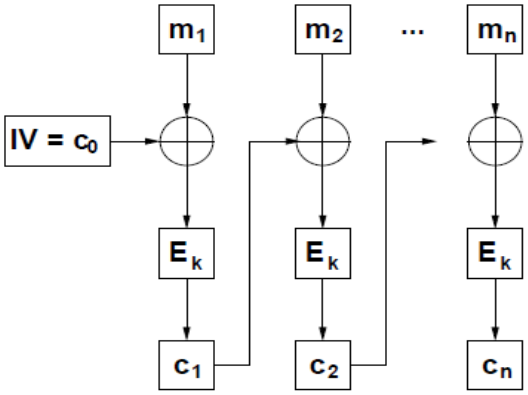
Déchiffrement :

$m_i = D_k(c_i)$

b) Le mode Cipher Block Chaining (CBC)

Le mode CBC a été introduit pour qu'un bloc ne soit pas codé de la même manière s'il est rencontré dans deux messages différents. Il faut ajouter une valeur initiale C0 aléatoire (ou IV pour « Initial Value »). Chaque bloc est d'abord modifié par XOR avec le bloc chiffré précédent avant d'être lui même chiffré. CBC est le mode de chiffrement le plus utilisé.

$$c_i = E_k(m_i \oplus c_{i-1})$$



Le déchiffrement nécessite l'inverse de la fonction de codage: $D_k = E_k^{-1}$

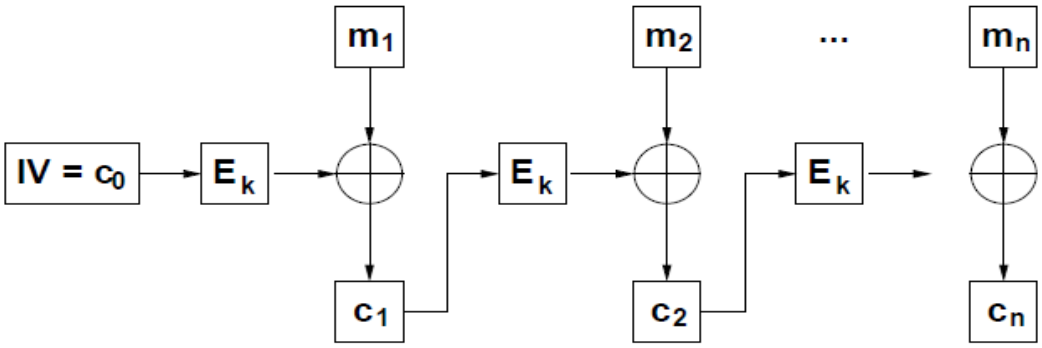
Déchiffrement :

$m_i = c_{i-1} \oplus D_k(c_i)$

c) Le mode Cipher FeedBack (CFB)

L'intérêt de ce mode est que le déchiffrement ne nécessite pas l'implémentation de la fonction : $D_k = E_k^{-1}$. Ce mode est donc moins sûr que le CBC.

$$c_i = m_i \oplus E_k(c_{i-1})$$



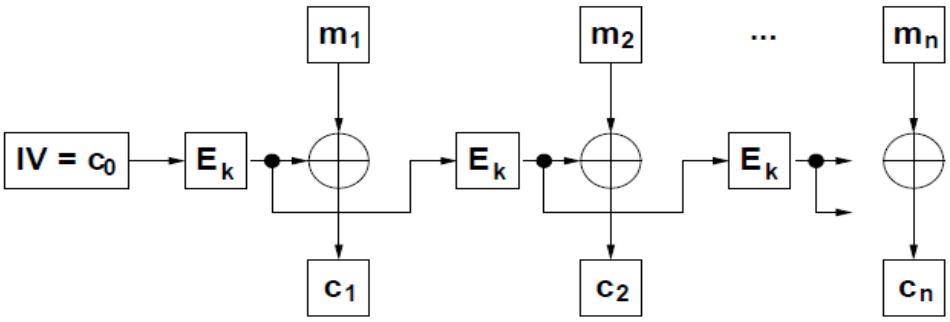
Déchiffrement :

$m_i = c_i \oplus E_k(c_{i-1})$

d) Le mode Output FeedBack (OFB)

Une variante du mode précédent permet d'avoir un chiffrement et un déchiffrement totalement symétrique, c'est le mode OFB.

$$z_0 = c_0 \ ; \ z_i = E_k(z_{i-1}) \ ; \ c_i = m_i \oplus z_i$$



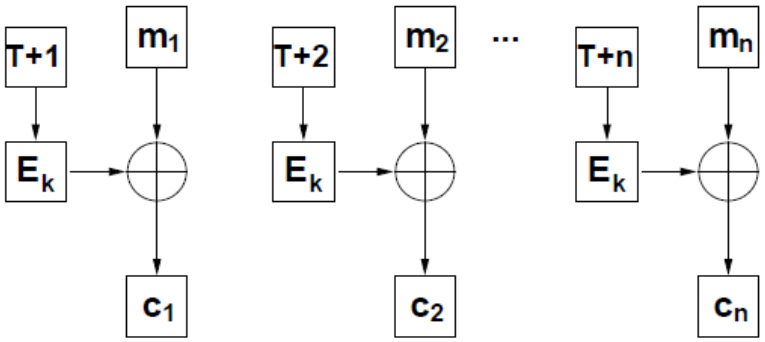
Déchiffrement

$z_i = E_k(z_{i-1})$; $m_i = c_i \oplus z_i$

e) Le mode Counter-mode encryption (CTR)

Ce mode est également totalement symétrique, mais en outre facilement parallélisable (chiffrement de plusieurs blocs en parallèle). Il fait intervenir le chiffrement d'un compteur de valeur initiale T. L'intérêt d'un tel mode est que les différents calculs sont indépendants, comme pour le mode ECB, mais qu'un même bloc n'est jamais codé de la même façon.

$$c_i = m_i \oplus E_k(T + i)$$



Déchiffrement :

$m_i = c_i \oplus E_k(T+i)$

1.1.2 Chiffrement par blocs itérés

Les fonctions de chiffrement sont en général des itérations d'une fonction de ronde $\boldsymbol{g}$.

Les paramètres d'un chiffrement par blocs itérés sont la taille des blocs ***n***, le nombre de rondes (round) ***r*** et une clé ***K***.
Le chiffré est obtenu par ***r*** applications itérées de la fonction de ronde ***g*** au même clair dépendant d'une clé de ronde. Les clés de ronde sont dérivées de la clé ***K***.
Ainsi dans un chiffrement par blocs itérés à r rondes :

$C_i=g(C_{i-1}, K_i)$ pour i=1,…,r

avec
C_0 : le clair,
g : la fonction de ronde,
K_i : les clés de ronde et
C_r : le chiffré

Le déchiffrement se déroule suivant le processus inverse ⇒ g doit être inversible.

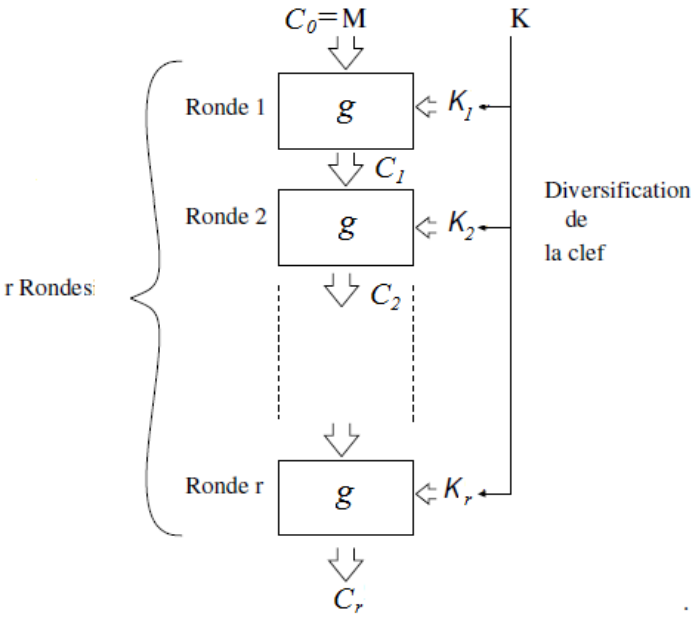


Figure 2.4 – Principe du chiffrement par blocs itérés

Il existe deux variantes :

- **SPN**
- **Feistel**

a) Réseau SPN (Permutation Substitution Network)

Les réseaux SPN sont constitués d'une succession de :

- substitution,
- permutation,
- XOR bit à bit avec la clé.

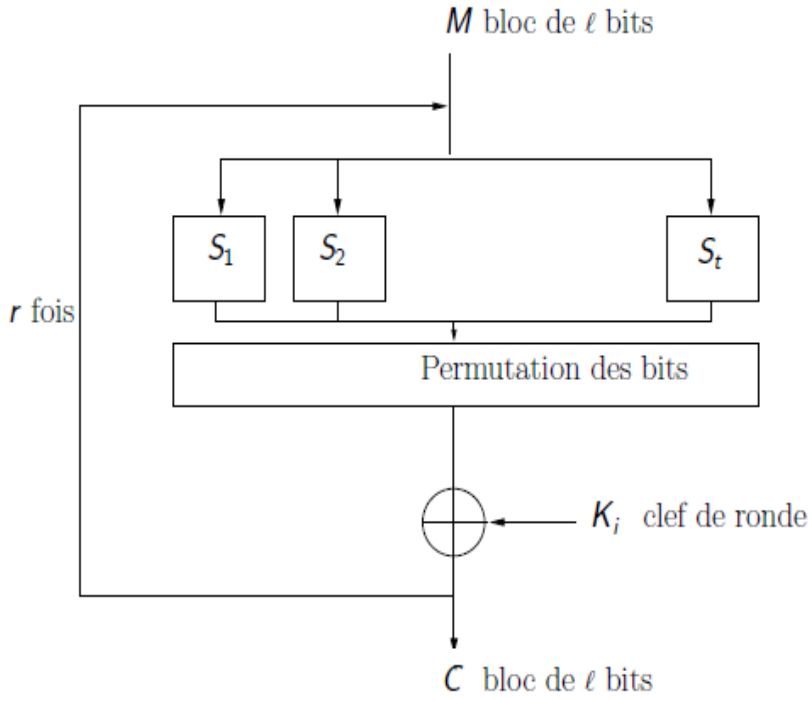
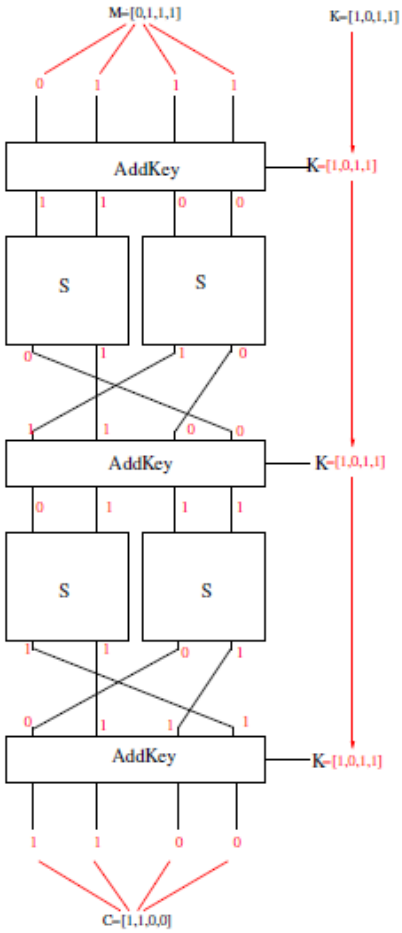


Figure 2.5 – Réseau SPN

Les boites S_i de substitution sont des fonctions

$S_i : \{0,1\}^{l/t} \rightarrow \{0,1\}^{l/t}$



X	00	01	10	11
$S(X)$	10	11	00	01

Figure 2.6 – Exemple de chiffrement par réseau SPN

b) Réseau Feistel

Une des techniques utilisées dans les algorithmes de chiffrement par bloc est le réseau de Feistel.
Il s'agit d'une succession d'étapes semblables ("rondes"). On effectue une opération sur la moitié des données, dont le résultat est combiné à l'autre moitié, et on inverse les deux parties.
Un réseau de Feistel est constitué de plusieurs tours successifs. Au final on obtient une bijection partant des N bits d'entrée vers N bits de sortie.

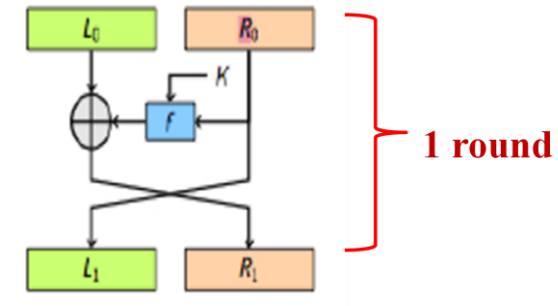


Figure 2.7 – Opération élémentaire d'un réseau de Feistel

Nous avons :

$$\begin{cases} L_1 = R_0 \\ R_1 = L_0 \oplus f(R_0, K) \end{cases}$$

Ainsi :

$$\begin{cases} R_0 = L_1 \\ L_0 = R_1 \oplus f(L_1, K) \end{cases}$$

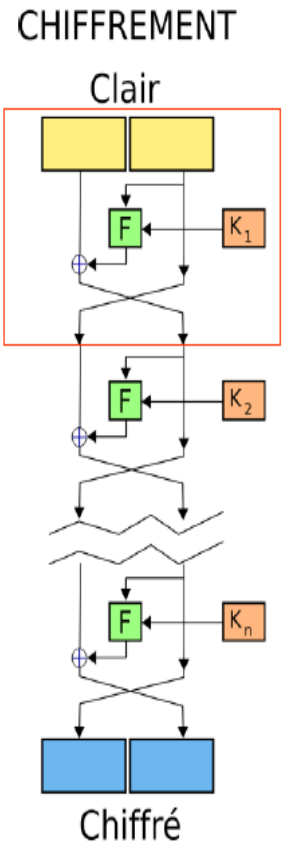


Figure 2.8 – Principe de chiffrement par réseau de Feistel

$L_i=R_{i-1}$ $R_i=L_{i-1}\oplus f(R_{i-1},K_i)$

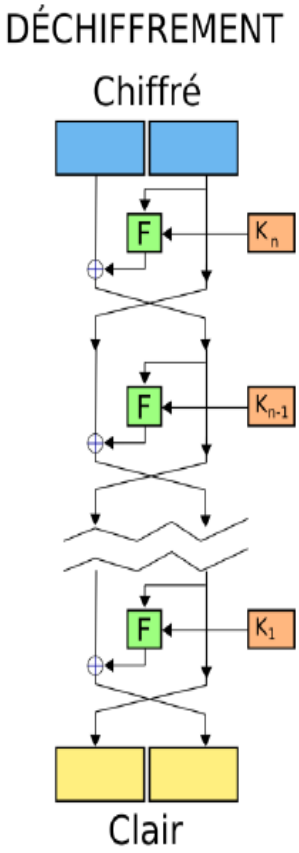


Figure 2.9 – Principe de déchiffrement par réseau de Feistel

$L_{i-1}=R_i \oplus f(L_i, K_i)$ $R_{i-1}=L_i$

Exemple d'algorithme fonctionnant avec le réseau Feistel:

- **DES (Data Encryption Standard): bloc=64bits, clé=56bits**
- **Triple DES**

1.2 Le chiffrement par flot (stream cipher)

Le chiffrement par flot (ou chiffrement en continu, stream cipher): permet de travailler sur un message de taille arbitraire, et le traiter octet par octet (voir bit par bit). Ces algorithmes sont généralement plus rapides mais moins résistants que les chiffrements par blocs.

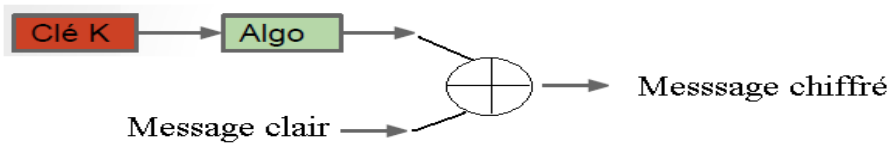


Figure 2.10 – Principe du chiffrement par flot

Exemple:
- A5/1 (chiffrement GSM): clé=64bits (seulement 54 pour le GSM)
- RC4 (WEP): clé variable
- E0 (Bluetooth): clé=128bits en général

Un chiffrement par flot se présente sous la forme d'un générateur de nombres pseudo-aléatoires avec lequel on opère un ou exclusif (XOR) entre un bit à la sortie du générateur et un bit provenant du message en clair.

Propriétés :

- La séquence qui sert au chiffrement ne dépend pas du message clair, mais uniquement de la clé.
- Il est possible de chiffrer des messages de tailles variables.
- Le chiffrement et le déchiffrement s'effectuent de la même manière (XOR)
- L'impact de la modification d'une partie du message chiffré pendant la transmission du message est limité à cette partie du message déchiffré.

Une des principales caractéristiques des algorithmes de chiffrement par flot est qu'ils permettent d'atteindre un très haut niveau de performances. Ces performances s'expriment soit en termes de vitesse de chiffrement soit en termes d'efficacité matérielle. On distingue deux principaux types d'algorithmes par flot :

- Les algorithmes adaptés à une implantation logicielle, qui peuvent atteindre des vitesses de chiffrement très élevées (plusieurs Gbits/s).
- Les algorithmes adaptés à une implantation matérielle, dont les implantations sont efficaces en termes de taille ou de consommation électrique.

1.2.1 Comparaison avec le chiffrement par bloc

Dans le cas d'un algorithme de chiffrement par bloc, on ne peut commencer à chiffrer et à déchiffrer un message que si l'on connaît la totalité d'un bloc. Ceci induit un délai de transmission et nécessite le stockage successif des blocs dans une mémoire tampon. Au contraire, dans les procédés de chiffrement par flot, chaque bit transmis peut être chiffré ou déchiffré indépendamment des autres, en particulier sans qu'il soit nécessaire d'attendre les bits suivants. Un autre avantage du chiffrement par flot est que le chiffrement par blocs demande au message clair d'être au moins de la taille du bloc. Les messages plus courts doivent être soumis au remplissage («padding»). Ceci est inutile pour le chiffrement de flux.

D'autre part, dans les chiffrements par flot contrairement aux chiffrements par bloc, le processus de déchiffrement ne propage pas les erreurs de transmission. Supposons qu'une erreur survenue au cours de la communication ait affecté un bit du message chiffré. Dans le cas d'un chiffrement à flot, cette erreur affecte uniquement le bit correspondant du texte clair, et ne le rend donc généralement pas complètement incompréhensible. Par contre, dans le cas d'un chiffrement par bloc, c'est tout le bloc contenant le bit erroné qui devient incorrect après déchiffrement. C'est pour cette raison que le chiffrement par flot est utilisé pour protéger la confidentialité dans les transmissions via des canaux bruités.

1.2.2 Présentation de l'algorithme RC4

RC4 est un algorithme de chiffrement symétrique par flot (à la volée) utilisant le mode OFB (Output Feedback). Cette méthode est extrêmement rapide, 10 fois plus que DES. De plus, les S Boxes évoluent avec l 'exécution : le même bit ne donne pas toujours le même résultat. RC4 est donc très efficace.

RC4 est un algorithme de génération de bits pseudo aléatoires. Il permet, à partir d'une clé secrète d'obtenir une séquence binaire aléatoire et unique de même longueur du texte clair. Ensuite, il procède à faire un XOR bit par bit pour obtenir la forme chiffrée.
Le destinataire va faire un XOR entre la séquence reçue et la séquence aléatoire originale pour retrouver le message en clair.

Problème : Transmettre la séquence aléatoire
Solution : Utiliser une séquence de chiffrement pseudo aléatoires (keystream).
RC4 peut produire des bits aléatoires en continu à partir d'une table d'états donnée. Si le destinataire peut reconstruire la table d'états donnée, il pourra régénérer la suite aléatoire et déchiffrer le message.
La table d'états (permutation) est construite par le Key Scheduling Algorithm (KSA) à partir de la clé de chiffrement. La clé sert à configurer la table d'états et la table d'états sert à générer un flux binaire pseudo-aléatoire : Pseudo Random Generator Algorithm (PRGA).

Key Scheduling Algorithm (KSA)
Algorithme d'initialisation de RC4 : K est une clé de longueur L. S est une permutation des entiers de 0 à 255.

Initialisation de la permutation initiale
for i in 0 to 255 do
S[i] := i
end for
j := 0
Mélange de S dépendant de K
for i in 0 to 255 do

j := (j + S[i] + K[i % L]) % 256
swap(S[i], S[j])
end for

Une fois son état interne initialisé, RC4 peut générer un par un des octets pseudo aléatoires.

Pseudo Random Generator Algorithm (PRGA)
i := i + 1
j := j + S[i]
swap(S[i], S[j])
octet := S[(S[i] + S[j]) mod 256]
retourner octet

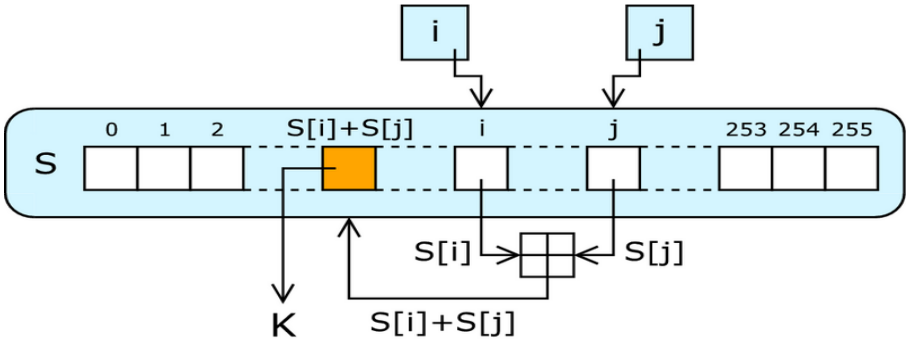


Figure 2.11 – Schéma de la génération d'un octet par RC4

1.3 Algorithme AES

L'AES (Rinjdaël) est le standard actuel utilisé pour les chiffrements symétriques. Contrairement à DES, il n'utilise pas de réseau de Feistel.
Taille de bloc : 16 octets (128 bits)
Taille de clé: trois variantes (128, 192, 256 bits)
Il a été publié en 2001 suite à un appel du NIST ayant donné lieu à cinq ans de compétition (FIPS-PUB 197) entre plusieurs algorithmes (MARS, RC6, Rinjdaël, Serpent, Twofish)

Performances:

AES128 est 2,7 fois plus rapide que 3DES et presque aussi rapide que le DES

1.3.1 Rappel algèbre et corps AES

Nombres binaires et hexadécimaux

Nombres hexadécimaux .Les chiffres vont de 0 à F et permettent de travailler en base 16 :

0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Un chiffre hexadécimal représente 4 bits (car 2^4 =16), donc, par exemple, il faut 32 chiffres hexadécimaux pour représenter 128 bits.

0	1	2	3	4	5	6	7
0000	0001	0010	0011	0100	0101	0110	0111

8	9	A	B	C	D	E	F
1000	1001	1010	1011	1100	1101	1110	1111

Exercice : Convertir 6453 en hexadécimal, puis en binaire.

Le corps F_2 :

L'ensemble F_2= {0,1} des entiers modulo 2 est un corps (la multiplication correspond au « et logique » et l'addition à l'addition modulo 2).

Polynômes irréductibles :

Un polynôme P(x) $\in F_2[x]$ et de degré ≥ 1 est irréductible s'il n'existe pas deux polynômes $P_1(X)$, $P_2(x) \in F_2[x]$, de degré ≥ 1, et tels que
$P(x) = P_1(x)\, P_2(x)$.
Pour tout d ≥ 1, Il existe des polynômes irréductibles de degré d.

Les extensions du corps F_2

Pour $P(x) \in F_2[x]$ de degré $d \geq 1$ et irréductible, l'ensemble $F_2[x] / P(x)$ est un corps, noté F_2^d. Sa structure ne dépend pas du choix du polynôme P, mais juste du degré d.

Le corps A.E.S

Représentation :
Mot de 8 bit correspond à un mot de deux chiffres hexadécimaux et à des polynômes de F_2 de degré ≤ 7.
Les octets sont représentés en hexadécimal.

Exemple : On identifie
8A =1000 1010
$=1.x^7+0.x^6+0.x^5+0.x^4+1.x^3+0.x^2+1.x+0$
$= x^7+ x^3+x$

Pour le corps A.E.S, on travaille dans le quotient $F_2[x] /R(x)$; R(x) est le polynôme de Rinjdael (irréductible sur F_2)
$R(X) = x^8+x^4+x^3+x+1$
$F_2[x] /R(x) = F_2^8$

Exemple :
$2A + 37 = (x^5+x^3+x) + (x^5+x^4+x^2+x+1)$
$= x^4+x^3+x^2+1 = 1D$
$2A \times 37 = x^6+x^5+x^4+x^2+x+1 = 77$

1.3.2 Fonctionnement de AES

Le système de chiffrement à **clé secrète AES** est un système basé sur le système Rijdndael construit par Joan Daemen et Vincent Rijmen.
Pour AES les blocs de données en entrées et en sortie sont des blocs de 128 bits, c'est-à-dire de 16 octets.
Les clés secrètes ont au choix suivant la version du système : 128 bits (16 octets), 192 bits (24 octets) ou 256 bits (32 octets).
On découpe les données et les clés en octets et on les place dans des tableaux.
Les données comportent t_d=16 octets P_0, P_1,….. P_{15} qui sont classés dans un tableau ayant 4 lignes et 4 colonnes. Le tableau est rempli colonne par colonne.
De même la clé est découpée en octets (t_k=16, t_k=24 ou t_k=32 octets) k_0 ,k_1 ,…… ,k_{tk-1} .Ces octets sont aussi classés dans un tableau de 4 lignes et N_k colonnes (N_k=4, N_k=6 ou N_k=8).

p_0	p_4	p_8	p_{12}
p_1	p_5	p_9	p_{13}
p_2	p_6	p_{10}	p_{14}
p_3	p_7	p_{11}	p_{15}

k_0	k_4	k_8	k_{12}	k_{16}	k_{20}	k_{24}	k_{28}
k_1	k_5	k_9	k_{13}	k_{17}	k_{21}	k_{25}	k_{29}
k_2	k_6	k_{10}	k_{14}	k_{18}	k_{22}	k_{26}	k_{30}
k_3	k_7	k_{11}	k_{15}	k_{19}	k_{23}	k_{27}	k_{31}

Le système AES effectue **plusieurs tours** d'une même composition de transformations.

- **Le nombre de tours**

Suivant la version (la taille de la clé), ce nombre de tours noté n_r, est différent. Le nombre n_r est donné dans le tableau suivant.

N_k	4	6	8
n_r	10	12	14

- La fonction d'étage 'g' est l'ensemble des opérations que l'on fait subir au texte qui arrive à l'étage i. Elle est la même pour tous les étages sauf le dernier

- La fonction 'g' consiste en l'application successive de 4 opérations :

  - SUBBYTES
  - SHIFTROWS
  - MIXCOLUMNS
  - ADDROUNDKEY

Pour le dernier étage, on applique seulement les opérations :

- SUBBYTES
- SHIFTROWS
- ADDROUNDKEY

- **La clé de tour**

A partir de la clé secrète K, on génère des sous clés d'étages par l'algorithme de diversification de clé, EXPANDKEY[i]
On obtient les clefs, K^1, K^2, …, K^{nr}

Figure 2.12- Schéma du chiffrement AES (n_r =10)

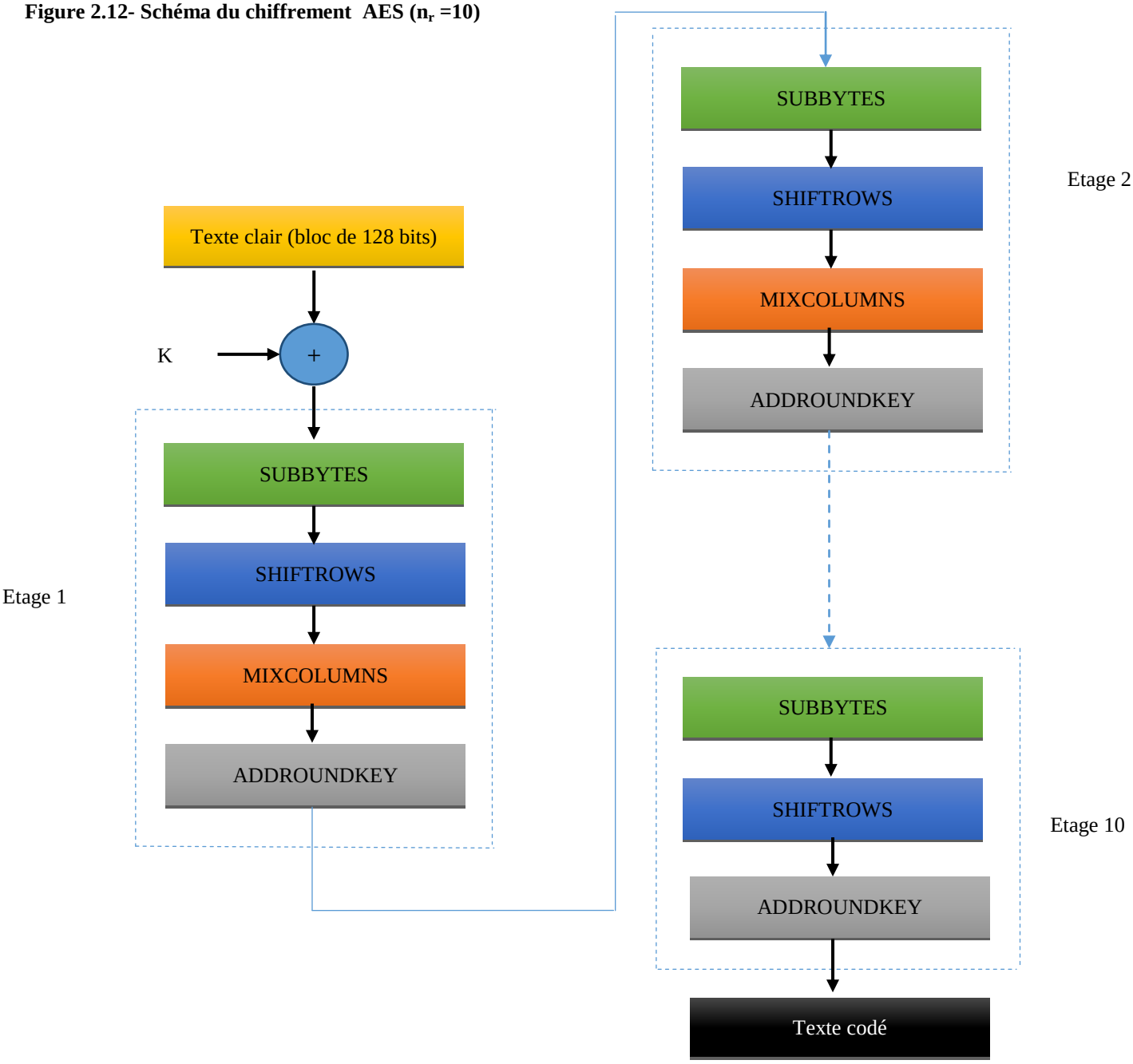
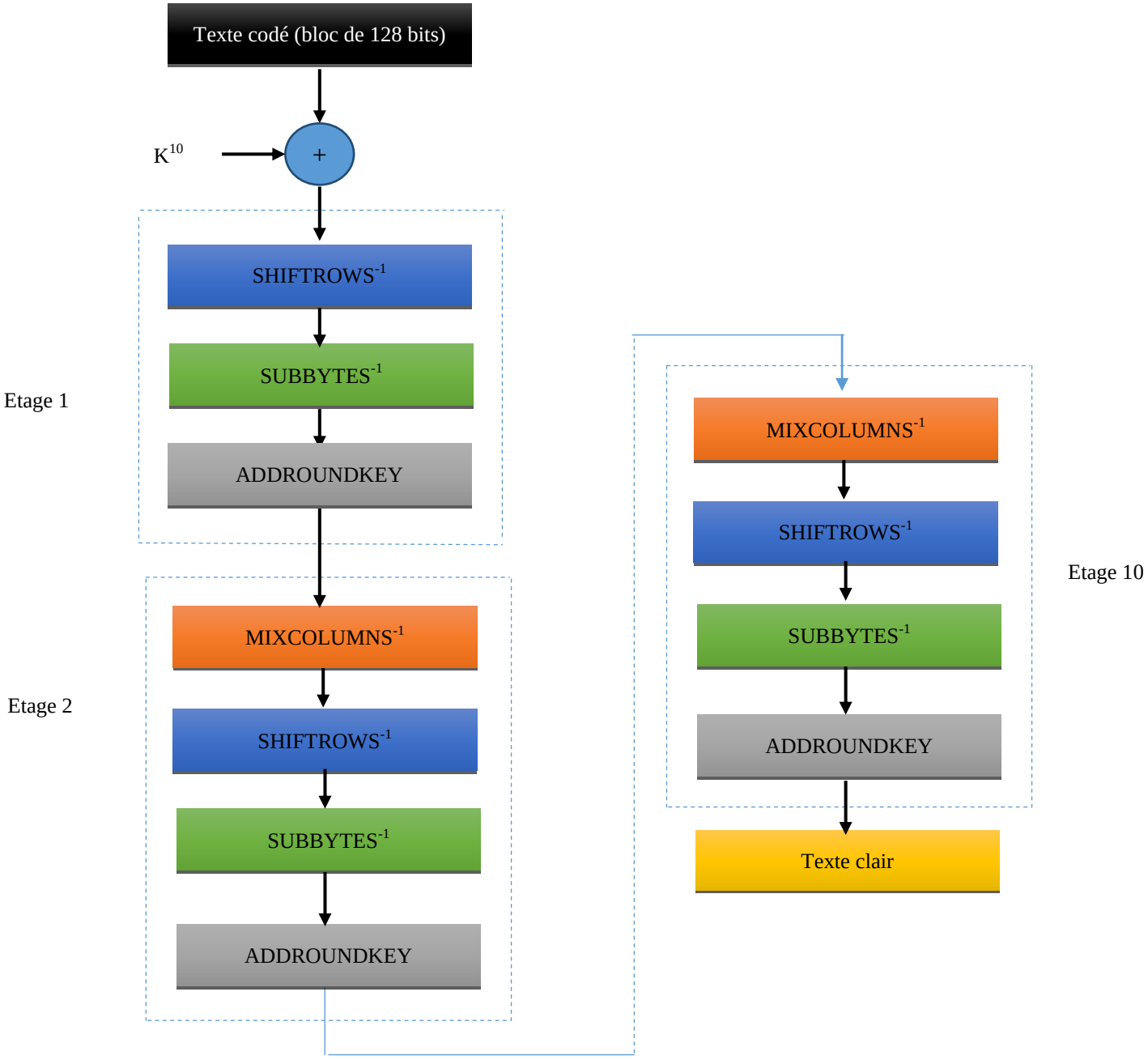


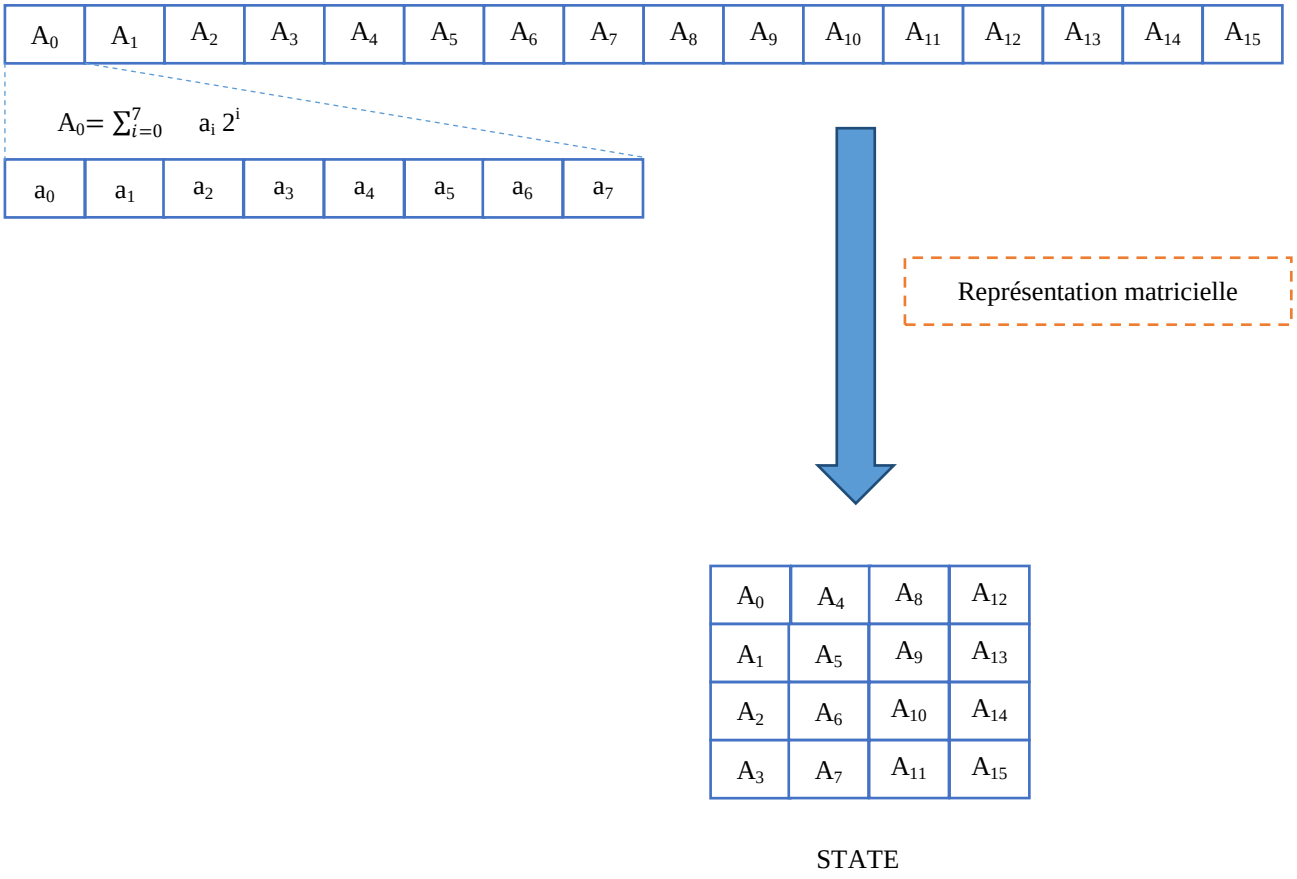
Figure 2.13- Schéma de déchiffrement AES (n_r =10)



1.3.3 Description de l'algorithme de chiffrement

- On prend le bloc de 128 bits que l'on considère comme une suite de 16 octets que l'on range dans une matrice 4×4
- On note STATE cette écriture matricielle et qui représente aussi le flux d'entrée-sortie de chaque étage

-



L'algorithme se déroule de la manière suivante :

1- Calculer STATE = STATE⊕K
2- Pour i de 1 à n_r -1 effectuer sur STATE les opérations suivantes :

a) Opération de substitution SUBBYTES

Cette opération agit sur chaque octet et utilise la table de substitution S-Box ci-dessous :

Table de substitution S-Box, utilisée par l'opération SUBBYTES:

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
5	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
6	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
7	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
8	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
9	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
A	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
B	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
C	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
D	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
E	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
F	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

En représentant les octets en hexadécimal, par exemple si la valeur de l'octet Aij vaut 7B, cette dernière sera substituée selon la table par : 21

Lors du déchiffrement, on utilise la table S-Box^{-1} suivante (inverse de S-Box) :

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	52	09	6A	D5	30	36	A5	38	BF	40	A3	9E	81	F3	D7	FB
1	7C	E3	39	82	9B	2F	FF	87	34	8E	43	44	C4	DE	E9	CB
2	54	7B	94	32	A6	C2	23	3D	EE	4C	95	0B	42	FA	C3	4E
3	08	2E	A1	66	28	D9	24	B2	76	5B	A2	49	6D	8B	D1	25
4	72	F8	F6	64	86	68	98	16	D4	A4	5C	CC	5D	65	B6	92
5	6C	70	48	50	FD	ED	B9	DA	5E	15	46	57	A7	8D	9D	84
6	90	D8	AB	00	8C	BC	D3	0A	F7	E4	58	05	B8	B3	45	06
7	D0	2C	1E	8F	CA	3F	0F	02	C1	AF	BD	03	01	13	8A	6B
8	3A	91	11	41	4F	67	DC	EA	97	F2	CF	CE	F0	B4	E6	73
9	96	AC	74	22	E7	AD	35	85	E2	F9	37	E8	1C	75	DF	6E
A	47	F1	1A	71	1D	29	C5	89	6F	B7	62	0E	AA	18	BE	1B
B	FC	56	3E	4B	C6	D2	79	20	9A	DB	C0	FE	78	CD	5A	F4
C	1F	DD	A8	33	88	07	C7	31	B1	12	10	59	27	80	EC	5F
D	60	51	7E	A9	19	B5	4A	0D	2D	E5	7A	9F	93	C9	9C	EF
E	A0	E0	3B	4D	AE	2A	F5	B0	C8	EB	BB	3C	83	53	99	61
F	17	2B	04	7E	BA	77	D6	26	E1	69	14	63	55	21	0C	7D

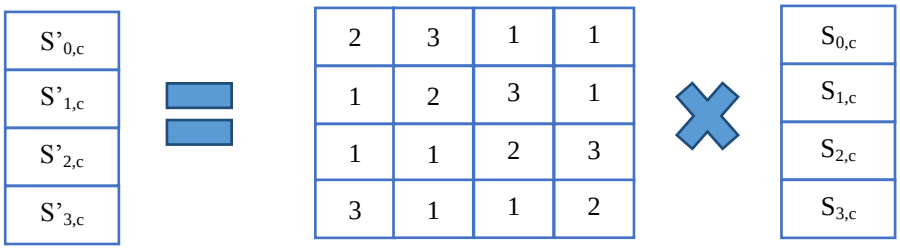
b) Opération SHIFTROWS : sur le résultat de SUBBYTES on effectue une permutation notée SHIFTROWS (permutation circulaire sur les éléments des lignes de la matrice des octets B_{ij})

B_0	B_4	B_8	B_{12}
B_1	B_5	B_9	B_{13}
B_2	B_6	B_{10}	B_{14}
B_3	B_7	B_{11}	B_{15}

B_0	B_4	B_8	B_{12}
B_5	B_9	B_{13}	B_1
B_{10}	B_{14}	B_2	B_6
B_{15}	B_3	B_7	B_{11}

Le même processus est appliqué lors du déchiffrement mais avec un décalage dans le sens inverse.

c) Opération MIXCOLUMNS : Cette opération consiste à réaliser pour chaque colonne de la matrice STATE résultant de l'opération SHIFTROWS le produit matriciel suivant :



$S_{0,c}$ représente l'octet de la ligne 0 et la colonne c de la matrice STATE

Remarque 3: l'opération MIXCOLUMNS assure la diffusion. En effet, une différence sur un octet d'entrée se propage sur les quatre octets de sortie.
Remarque 4: lors du déchiffrement, on utilise le produit matriciel suivant :

d) Opération ADDROUNDKEYS[i]

Cette opération consiste en un XOR de la matrice STATE résultant de l'opération MIXCOLUMNS et de la clef du tour.

Expansion de clefs

- L'AES utilise la clé de chiffrement K et la soumet à une routine d'expansion
- Cette expansion génère $4\times(n_r +1)$ mots
- Le résultat de cette expansion consiste dans un tableau linéaire noté [Wi] avec i variant de 0 à $4\times(n_r +1)-1$

Contenu du [Wi]

- Les N_k (k=4, 6 ou 8) premiers mots (mot=32bits) [w0, …, WN_{k-1}] contient la clé K
- Les mots suivant sont calculés en faisant un XOR du mot précédent [Wi-1] et du mot situé N_k positions avant [WN_{k-1}]

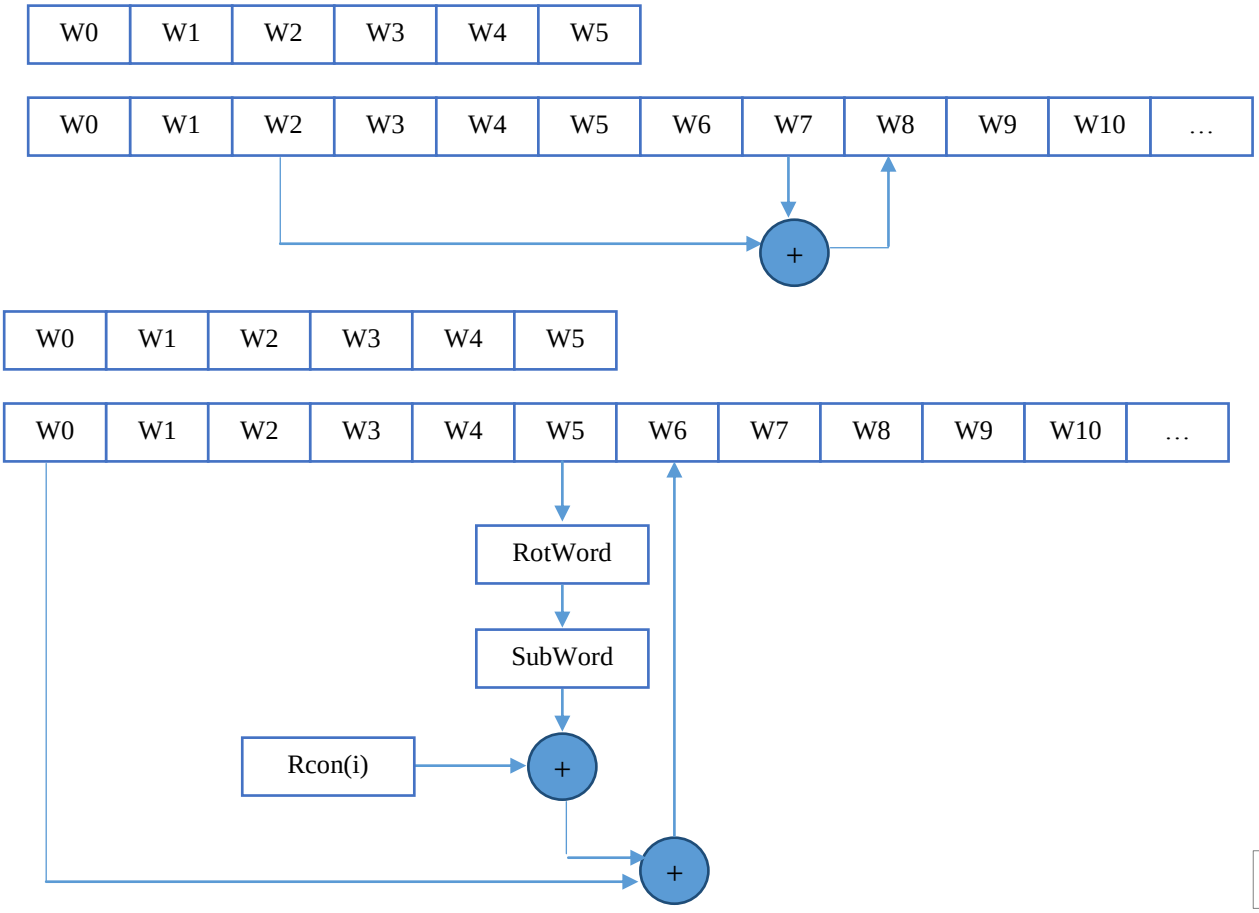
- Pour les mots situés sur une position multiple de N_K, une transformation est appliquée à [Wi-1] avant le XOR
- Cette transformation consiste en une permutation cyclique nommée '' ROTWORD'' d'un cran vers la gauche suivie d'une application de la S-Box ''SUBWORD'' séparément sur chaque octet du mot puis d'un XOR avec un vecteur dépendant du tour noté Rcon (i/N_K)
- Rcon (i/N_K) contient la valeur [X^{i-1}, 00, 00, 00]

Rcon(i) sont définies par :

Rcon (i)= x^{i-1} mod $x^8+x^4+x^3+x+1$
Rcon (1)=x^0=01
Rcon (2)=x=02
Rcon (3)=x^2=04
Rcon(4)=x^3=08
Rcon(5)=x^4=10
Rcon(6)=x^5=20
Rcon(7)=x^6=40
Rcon(8)=x^7=80
Rcon(9)=x^8mod $x^8+x^4+x^3+x+1=x^4+x^3+x+1$=1B
Roc(10)=x^9mod$x^8+x^4+x^3+x+1=x^5+x^4+x^2+x$=36
…

Il est important de noter que l'expansion dans le cas d'une clé de 256 bits est différente des deux autres cas (clés de 128 bits et 192 bits). En effet, dans ce cas si le rang i-4 est un multiple de 8 la fonction SUBWORD est encore appliquée à Wi-1 avant le XOR.

Exemple : N_k=6 → clé de 192 bits



2 Cryptographie asymétrique

La cryptographie symétrique consiste à chiffrer puis déchiffrer un message en utilisant la même clé et le même algorithme.

La distribution des clés a été le point faible des systèmes de cryptographie symétrique, d'où la proposition des algorithmes à clés publiques (algorithmes asymétriques)
La cryptographie asymétrique (à clés publiques) exige que chacun des correspondants possède une clé publiée dans un annuaire utilisée par tous le monde pour chiffrer des messages destinés à un individu particulier, et l'autre privée que cet individu est seul à détenir et qui lui permet de déchiffrer les messages qu'il reçoit.

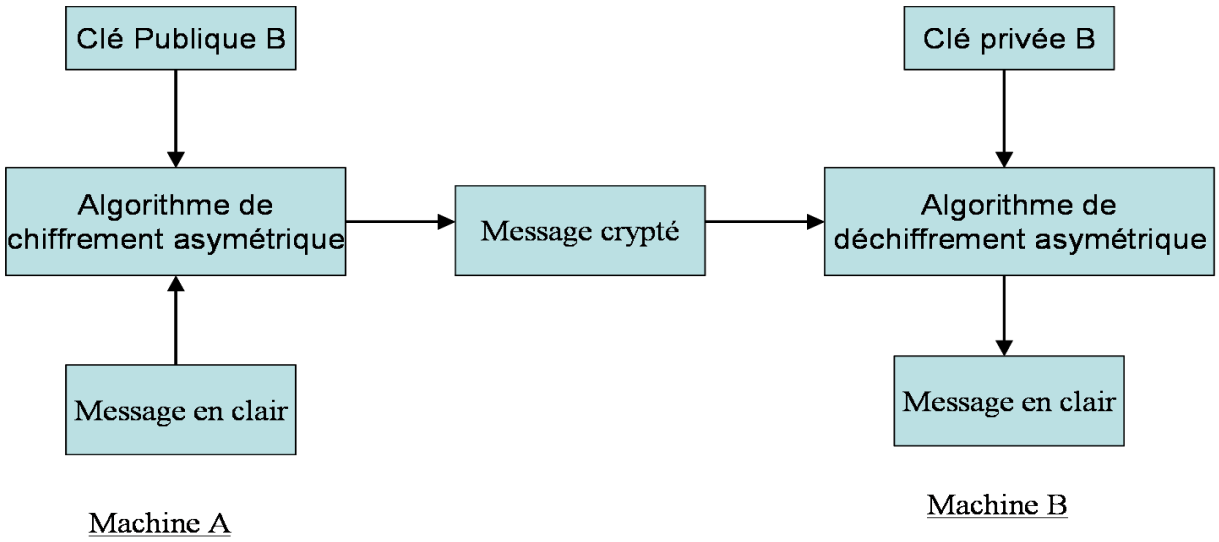


Figure 2.14 – Principe du chiffrement à clé publique

Etant donné les avantages potentiels de la cryptographie à clé publique les chercheurs se sont attelés à la tache et quelques algorithmes ont été publiés.

Principe de fonctionnement du chiffrement asymétrique

- Afin de permettre à ses correspondants de lui envoyer des messages cryptés, Ali doit préparer sa clé publique qu'il diffusera à tous le monde dans un annuaire, il prépare également sa clé privée qu'il sera seul à connaitre.
- Pour chiffrer un message destiné à Ali, Omar doit récupérer la clé d'Ali publiée dans l'annuaire.
- Grace à sa clé privée, Ali peut déchiffrer et lire le message que Bob a envoyé.

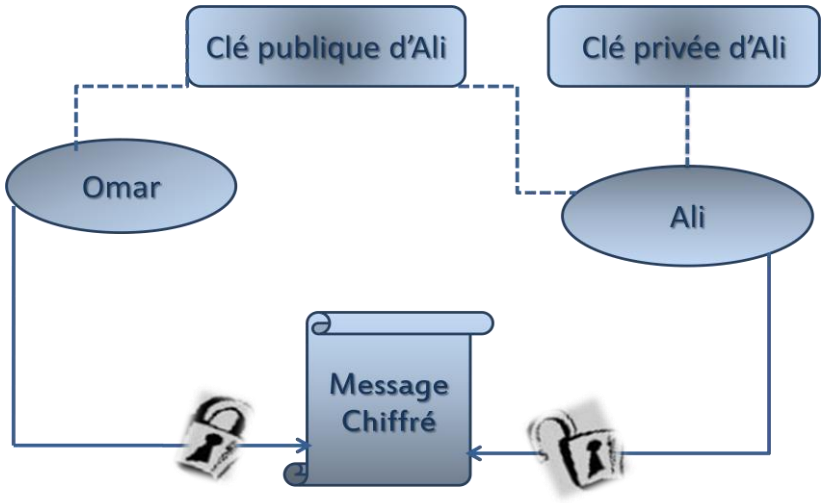


Figure 2.15 – Principe du chiffrement/déchiffrement asymétrique
Omar chiffre avec la clé publique d'Ali . Ali le déchiffre avec sa clé privée

2.1 Background mathématique

- **Nombres premiers**

Un nombre premier p est un nombre différent de 1 qui n'admet pas d'autres diviseurs que 1 et lui-même. Un nombre qui n'est pas premier est appelé nombre composé. ,On a les résultats suivants :

- Tout nombre admet au moins un facteur premier.
- L'ensemble des nombres premiers est infini.
- Tout entier peut se décomposer en produit de facteurs premiers.

Le problème de la factorisation en nombres premiers est un problème difficile qui nous le verrons a permis de mettre en place des systèmes cryptographiques presque inviolable. Le deuxième théorème s'interprète sur le plan cryptographique comme l'existence de grands nombres premiers.

a et *b* sont premiers entre eux ssi :
pgcd (a,b)=1.
$\exists u, v \in \mathbf{Z}$; $au + bv = 1$.

- **Congruence dans Z**

Les entiers relatifs a et b sont congrus modulo n s'ils ont même reste par la division par n. Il revient au même de dire que b-a est multiple de n. On note :

$$a \equiv b[n] \text{ ou } a \equiv b(\text{mod } n).$$

Si $x \equiv x'[n]$ et $y = y'[n]$, alors $x + y \equiv x' + y'[n]$ et $xx' \equiv yy'[n]$.

- **Ensemble Quotient $\mathbf{Z}|n\mathbf{Z} = \mathbf{Z}_n$**

La relation de congruence est une relation d'équivalence sur $\mathbf{Z}$.
L'ensemble $\bar{x}$ des éléments congrus à x de $\mathbf{Z}$ est dit classe d'équivalence de x modulo n.
L'ensemble des classes d'équivalence modulo n est noté $\mathbf{Z}|n\mathbf{Z} = \mathbf{Z}_n$.

on définit deux lois internes sur $\mathbf{Z}_n$ par :

$$\overline{xy} = \overline{yx} \quad \text{et} \quad \bar{x} + \bar{y} = \overline{x + y.}$$

L'ensemble quotient $\mathbf{Z}_n$ muni des lois précédentes est un anneau.

- **Algorithme d'Euclide**

Cet algorithme a été établi et baptisé ainsi par Etienne Bézout, il permet de calculer le pgcd de deux entiers a et b en effectuant un nombre fini de divisions euclidiennes.
Soient a et b deux entiers positifs, on pose $r_0 = a$ et $r_1 = b$, et tant que $r_i > 0$ on effectue les divisions euclidiennes successives suivantes.

$$\begin{cases} r_0 = r_1 q_1 + r_2, & \text{où} \quad 0 \le r_2 < r_1 \\ r_1 = r_2 q_2 + r_3, & \text{où} \quad 0 \le \mathrm{r}_3 < r_2 \\ \ldots \\ r_{k-2} = r_{k-1} q_{k-1} + r_k, & \text{où} \quad 0 \le \mathrm{r}_k < r_{k-1} \\ r_{k-1} = r_k q_k + r_{k+1}, & \text{où} \quad 0 \le \mathrm{r}_{k+1} w < r_k \end{cases}$$

Pour chaque $k \ge 0$, on a $p\gcd(a,b) = p\gcd(r_k, r_{k+1})$. La suite des restes $(r_1, r_2, r_3, \ldots)$ étant une suite strictement décroissante d'entiers positifs, on obtient nécessairement un reste nul au bout d'un nombre fini de divisions.
Soit r_n le dernier reste non nul. On a $r_{n+1} = 0$, ce qui signifie que

$$p\gcd(a,b) = p\gcd(r_n, r_{n+1}) = p\gcd(r_n, 0) = r_n.$$

D'où l'algorithme d'Euclide : On effectue les divisions euclidiennes successives décrites ci-dessus jusqu'à obtenir un reste nul, le pgcd de *a* et *b* est le dernier reste non nul.

- **Algorithme d'Euclide étendu**

L'algorithme d'Euclide étendu permet de déterminer $d = p\gcd(a,b)$ ainsi que deux entier u et v vérifiant.

$$d = au + bv$$

Reprenons la suite des divisions euclidiennes de l'algorithme d'Euclide et à chaque étape $k \geq 0$, calculons deux entiers u_k et v_k tels que

$$r_k = au_k + bv_k$$

On constate que $u_0 = 1,\ v_0 = 0,\ u_1 = 0$ et $v_1 = 1$.

On en déduit que pour $k \geq 0$, on a $\begin{cases} u_{k+1} = u_{k-1} - u_k q_k \\ v_{k+1} = v_{k-1} - v_k q_k \end{cases}$

Si $d = r_n$ est le dernier non nul, on a $u = u_n$ et $v = v_n$.

D'où l'algorithme d'Euclide étendu :
Soient deux entiers positifs a et b. Pour déterminer $d = p\gcd(a,b)$, ainsi que deux entiers u et v tels que $d = au + bv$,

On écrit $\begin{cases} r_0 = a, \\ u_0 = 1, \\ v_0 = 0, \end{cases} \begin{cases} r_1 = b, \\ u_1 = 0, \\ v_1 = 1, \end{cases}$ et $\forall k \geq 1, \begin{cases} r_{k+1} = r_{k-1} - r_k q_k, \\ u_{k+1} = u_{k-1} - u_k q_k, \\ v_{k+1} = v_{k-1} - v_k q_k, \end{cases}$

Jusqu'à obtenir un reste nul.

Si r_n est le dernier reste non nul, on a $\begin{cases} d = r_n, \\ u = u_n, \\ v = v_n, \end{cases}$

- **Théorèmes de Fermat et d'Euler.**

Petit théorème de Fermat (Pierre de Fermat, 1601-1665) :
Etant donné un nombre premier p et un entier $a \in \mathbf{Z}$, on a :

$$a^p \equiv a \pmod{p}$$

Indicatrice d'Euler : Soit n un entier positif, l'indicatrice d'Euler de n, notée $\varphi(n)$, est définie comme étant égal au nombre des entiers k vérifiant

$$(1 \leq k \leq n) \ \ et \ \ (p\gcd(k,n) = 1).$$

Notons que pour tout entier positif n, on a pgcd (1,n) = 1, ce qui fait que $\varphi(n) \geq 1$.

Théorème d'Euler : Soit n un entier positif, et soit $a \in \mathbf{Z}$ un entier premier avec n, alors :

$$a^{\varphi(n)} \equiv 1 \pmod{n}.$$

– **Théorème des restes chinois**

Un système de congruences est un système de la forme :

$$(S): \begin{cases} x \equiv a_1 \pmod{m_1}, \\ x \equiv a_2 \pmod{m_2}, \\ \ldots \quad\quad \ldots \\ x \equiv a_k \pmod{m_k}. \end{cases}$$

Où les a_i, et les m_i sont des entiers donnés. Résoudre le système (S) consiste à déterminer tous les entiers $x \in \mathbf{Z}$ vérifiant le système.

Lorsque les entiers m_i sont premiers entre eux deux à deux, nous allons voir que le système (S) admet toujours des solutions. Ce résultat connu sous le nom de Théorème des restes chinois parce que les Chinois en utilisaient des cas particuliers pour déterminer les dates de certains événements astronomiques.

Théorème chinois : Soit m et n deux entiers premiers entre eux.

Pour tout couple d'entiers $(a,b) \in \mathbf{Z}^2$, il existe $c \in \mathbf{Z}$ que l'on ait l'équivalence :

$$(SC): \begin{cases} x \equiv a \pmod{m}, \\ x \equiv b \pmod{n}. \end{cases} \Leftrightarrow (x \equiv c \pmod{mn}).$$

Le système (SC) admet donc une infinité de solutions dans $\mathbf{Z}$, ce sont les entiers de la forme $x = c + kmn,\ où\ k \in \mathbf{Z}$. Deux solutions distinctes sont congrus modulos *mn*.

– **Décomposition d'un entier en facteurs premiers**

Soit p un nombre premier. Si p divise un produit $q_1 q_2 ... q_n$ de n entiers, il existe au mois un indice $i \in \{1, 2, ..., n\}$ tel que *p* divise q_i.

Soit p un nombre premier. Si p divise un produit $p_1 p_2 ... p_n$ de n nombres premiers, il existe un indice $i \in \{1, 2, ..., n\}$ tel que $p = p_i$.

Théorème fondamental de l'arithmétique (décomposition de l'entier a en facteurs premiers) :
Tout entier $a > 1$ s'écrit de façon unique :

$$a = p_1^{\alpha 1} p_2^{\alpha 2} ... p_n^{\alpha n},$$

Où $\begin{cases} \text{les entiers } p_i \text{ sont premiers et vérifient } p_1 < p_2 < ... < p_n, \\ \text{les entiers } \alpha_i \text{ sont positifs.} \end{cases}$

– **Résidu quadratique**

Soit $n \geq 2$ un entier, a un entier inférieur à n. On dit que a est un résidu quadratique modulo n si $a = b^2$ pour une classe b. C'est-à-dire :

$$b^2 \equiv a \pmod{n}$$

Dressons par exemple les listes des résidus quadratiques modulo quelques nombres premiers :

2 : 1
3 : 1
5 : 1, 4
7 : 1, 2, 4
11 : 1, 3, 4, 5, 9
13 : 1, 3, 4, 9, 10, 12
17 : 1, 2, 4, 8, 9, 13, 15, 16
19 : 1, 4, 5, 6, 7, 9, 11, 16, 17
23 : 1, 2, 3, 4, 6, 8, 9, 12, 13, 16, 18
29 : 1, 4, 5, 6, 7, 9, 13, 16, 20, 22, 23, 24, 25, 28

Notons qu'il ya $(n-1)/p$ résidus quadratiques pour chacun de ces nombres premiers.

- **Symbole de Legendre**

Définition 2.09 : Soit p un nombre premier et a un entier. On pose :

$$\left(\frac{a}{p}\right)=\begin{cases}0 & \text{si } p \text{ divise } a\\ 1 & \text{si } a \text{ est résidu quadratique modulo } p\\ -1 & \text{si } a \text{ est non résidu quadratique modulo } p\end{cases}$$

On appelle symbole de Legendre l'expression $\left(\frac{a}{p}\right)$

- **Logarithme discret**

Le calcul du nombre x qui fait en sort que $y=b^x$, pour y et b donnés, correspond à trouver le logarithme, $\log_b(y)$. Dans le cas des entiers modulos un nombre premier, on peut considérer le même problème ; parce que les puissances successives b^x parcourent toutes les entiers $(\neq 0)$modulo p, lorsque x parcourt tous les entiers modulo $p-1$. Ainsi, avec $p=11$, on a :

y	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
$\log_6(y)$	0	9	2	8	6	1	3	7	4	5

Logarithme discret de quelques entiers (p=11 et b=6)

On obtient ainsi la notion de logarithme discret. Ici, le mot discret sert à distinguer de la notion usuelle qu'on qualifie souvent de continue. Pour p un nombre premier, et b entre 2 et $p-2$, on peut reformuler le théorème d'Euler-Fermat comme

$$y \equiv b^x (\text{mod p})$$

Exactement lorsque $x \equiv \log_b(x)(\text{mod } p-1)$.

2.2 Algorithme de Diffie-Hellman

Diffie et Hellman ont proposé en 1976 un protocole d'échange de clés totalement sécurisé. Il repose sur le problème du logarithme discret.

Le problème est le suivant. Alice et Bob veulent s'échanger un message chiffré en utilisant une clé K. Ils veulent s'échanger cette clé K, mais ils ne disposent pas de canal sécurisé pour cela. Le protocole d'échange de clés de Diffie et Hellman répond à ce problème lorsque K est un nombre entier.

Étant donnés des entiers p,a,x, avec p premier et $1 \leq a \leq p-1$:

- il est facile de calculer l'entier $y = ax \pmod{p}$
- si on connait $y = ax \pmod{p}$, a et p, il est très difficile de retrouver x, pourvu que p soit assez grand. Retrouver x connaissant $ax \pmod{p}$, a et p revient à résoudre le problème du logarithme discret.

Le protocole de Diffie-Hellman est décrit dans le tableau suivant :

	Alice	Bob
Étape 1 :	Alice et Bob choisissent ensemble un grand nombre premier p et un entier $1 \leq a \leq p-1$. Cet échange n'a pas besoin d'être sécurisé.	
Étape 2 :	Alice choisit secrètement x_1.	Bob choisit secrètement x_2.
Étape 3 :	Alice calcule $y_1 = a^{x_1} \pmod{p}$.	Bob calcule $y_2 = a^{x_2} \pmod{p}$.
Étape 4 :	Alice et Bob s'échangent les valeurs de y_1 et y_2. Cet échange n'a pas besoin d'être sécurisé.	
Étape 5 :	Alice calcule $y_2^{x_1} = (a^{x_2})^{x_1} = a^{x_1 x_2} \pmod{p}$ et appelle ce nombre K, la clé secrète à partager avec Bob.	Bob calcule $y_1^{x_2} = (a^{x_1})^{x_2} = a^{x_1 x_2} \pmod{p}$ et appelle ce nombre K, la clé secrète à partager avec Alice.

Ce protocole est vulnérable à l'attaque man in the middle, qui implique un attaquant capable de lire et de modifier tous les messages échangés entre Alice et Bob.

Cette attaque repose sur l'interception de a^{x1} et a^{x2}, ce qui est facile puisqu'ils sont échangés en clair ; l'élément a étant supposé connu par tous les attaquants. Pour retrouver les nombres x_1 et x_2 et ainsi casser complètement l'échange, il faudrait calculer le logarithme discret de a^{x1} et a^{x2}, ce qui est impossible en pratique.

Mais un attaquant peut se placer entre Alice et Bob, intercepter la clé a^{x1}, envoyée par Alice et envoyer à Bob une autre clé a^{x11} , se faisant passer pour Alice. De même, il peut remplacer la clé a^{x2} envoyée par Bob à Alice par une clé a^{x22}, se faisant passer pour Bob. L'attaquant communique ainsi avec Alice en utilisant la clé partagée $g^{ab'}$ et communique avec Bob en utilisant la clé partagée $g^{a'b}$, Alice et Bob croient communiquer directement. C'est ce que l'on appelle attaque de l'homme du milieu ou man in the middle.

Alice et Bob croient ainsi avoir échangé une clé secrète alors qu'en réalité ils ont chacun échangé une clé secrète avec l'attaquant, l'homme du milieu.

C'est pourquoi ce protocole fut en réalité très vite supplanté par les algorithmes de chiffrement à clé publique de type RSA.

2.3 Algorithme RSA

L'algorithme RSA est sans doute le plus utilisé des systèmes à clé publique actuellement ; il a été présenté en 1977 par Ron Rivest, Adi Shamir, et Len Adlmen. Il nécessite des clés d'au moins 1024 bits pour obtenir une sécurisation satisfaisante.

Principe de fonctionnement du RSA

Etape 1: Création des clés

Ali choisit au hasard deux nombres premiers p et q et calcule $n = pq$. Il choisit au hasard e tel que :

$$\begin{cases} 1 < e < \phi(n) = (p-1)(q-1) \\ p \gcd(e, \phi(n)) = 1 \end{cases}$$

$\phi(n)$: le nombre d'entiers inférieurs à n, premiers avec n
Ali calcule l'entier d pour inverser la fonction de chiffrement tel que :

$$\begin{cases} 1 < d < \phi(n) \\ ed = 1 \bmod \phi(n) \end{cases}$$

La clé publique d'Alice est (n,e) et sa clé secrète est (n,d).

Etape 2: Chiffrement du message

Omar récupère la clé publique (n,e) d'Ali et souhaite lui envoyer la version cryptée d'un texte en clair, représenté par la donnée d'un entier m tel que :

$$0 \leq m < n.$$

Omar calcule :

$$c = m^e \bmod n$$

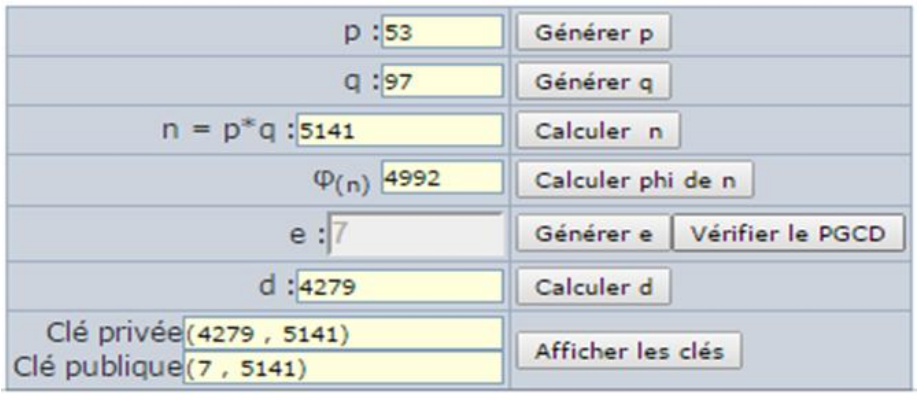
Etape 3: Déchiffrement du message

Lorsque Ali reçoit c , il calcule c^d, et récupère ainsi le message m puisque :

$$m = c^d \bmod n$$

Ce qu'est en rouge sont des nombres privée d'Ali.

Exemple



On va chiffrer le message **'Bonjour'.**

- On va prendre le code **ASCII** de chaque caractère :
  m = 66(B) 111(o) 110(n) 106(J) 111(o) 117(u) 114(r)

- En suite on va appliquer l'opération **$C= m^e \bmod n$** sur chaque valeur ASCII des caractères à chiffré :

$66^7 \bmod 5141 = 386$
$111^7 \bmod 5141 = 1858$
$110^7 \bmod 5141 = 2127$
$106^7 \bmod 5141 = 2890$
$111^7 \bmod 5141 = 1858$
$117^7 \bmod 5141 = 1774$
$114^7 \bmod 5141 = 737$

Le message chiffré est 386 1858 2127 2809 1858

On peut reconstituer le message en clair avec d :
- On va appliquer l'opération **$m= C^d \bmod n$** sur chaque valeur chiffrée des caractères du message:

$386^{4279} \bmod 5141 = 66 <=> B$
$1858^{4279} \bmod 5141 = 111 <=> o$
$2127^{4279} \bmod 5141 = 110 <=> n$
$2809^{4279} \bmod 5141 = 106 <=> j$
$1858^{4279} \bmod 5141 = 111 <=> o$
$1774^{4279} \bmod 5141 = 117 <=> u$
$737^{4279} \bmod 5141 = 114 <=> r$

On retrouve ainsi notre message en clair **'Bonjour'.**

.
Le principe de l'algorithme RSA repose sur le fait qu'il est très difficile et très long de factoriser un très grand nombre en deux facteurs premiers, pour cela il faut procéder de la manière inverse : générer les deux nombres premiers (p et q), puis les multiplier pour générer le nombre n.

Le réel problème du RSA (et des autres systèmes à clé publique) n'est pas la sécurité, mais la lenteur. Tous les algorithmes à clé publique sont 100 à 1000 fois plus lents que les algorithmes à clé secrète, quelle que soit leur implémentation (logicielle ou matérielle) !

En dépit de sa lenteur pour encoder de grands volumes de données, le **RSA** est le système de cryptographie asymétrique le plus utilisé actuellement dans les cartes bleues, les logiciels, les cartes bancaires, les messageries, ...

2.4 Algorithme El Gamal

L'algorithme El Gamal repose sur l'intractabilité du logarithme discret.

Principe :

On commence par choisir un grand nombre premier p, et un nombre g (mod p), qui sont tous deux connu de tous. L'utilisateur A choisit un grand nombre a (mod $p-1$) qui sera sa clé secrète de déchiffrement. La clé publique de A est le nombre $g^a \pmod{p}$.

Pour envoyer un message m à A, l'utilisateur B choisit aléatoirement un grand nombre entier $k \pmod{p}$, et il l'envoie à A la paire :

$$(K, M), \text{ où } K = (q^K \bmod p), \text{ et } M = (m \cdot g^{a-K} \bmod p).$$

Le destinataire A, qui connaît la clé secrète a, récupère le message m à partir de cette paire de la façon suivante :

Il calcule d'abord $(K^{-a} \bmod p) = (g^{-a \cdot K} \bmod p)$, à partir du premier élément du couple reçu ; puis il multiplie M par ce résultat pour obtenir :

$$\begin{aligned} M \cdot g^{-aK} &\equiv (m \cdot g^{a \cdot K}) \cdot g^{-a \cdot K} \pmod{p} \\ &\equiv m \cdot g^{a \cdot K - a \cdot K} \pmod{p} \\ &\equiv m \end{aligned}$$

Intuitivement, le message chiffré *C* envoyé à A est une version masquée de *m* obtenue par la multiplication par $g^{a \cdot K}$. Le nombre *K*, qui accompagne le message chiffré *C*, est un indice qui permet à A de retirer le masque. Cet indice $K = (g^K \bmod p)$ ne peut être utilisé que par quelqu'un qui connaît la clé *a*. Il semble que pour qu'un attaquant puisse casser le cryptosystème d'El Gamal, il doit retrouver la clé *a* à partir de la clé publique g^a. C'est donc dire qu'il aura trouvé une solution efficace (temps polynomial) au problème du calcul du logarithme discret.

3 Fonction à sens unique : Fonction de Hachage

Les fonctions de hachages sont des fonctions à sens uniques « sans collision », générant une sortie de taille fixe (appelée condensat ou empreinte), caractéristique des données fournies en entrée.

Ces fonctions sont dites à sens unique car il est impossible de retrouver les données initiales à partir de l'empreinte.

Une fonction est dite « sans collision » ou « injective » lorsqu'il est réputé très difficile de trouver deux sources différentes conduisant à un même résultat.

Les fonctions de hachage permettent entre-autres, grâce au calcul du condensat d'un document et la comparaison de celui-ci avec sa valeur initiale, de :

- contrôler l'intégrité d'un document.
- comparer un mot de passe entré par un utilisateur à un mot de passe stocké dans une base de données
- publier l'empreinte d'un logiciel : pour comparer l'empreinte fournie par l'éditeur et l'empreinte qu'il obtient sur le fichier téléchargé
- vérifier l'intégrité d'un message de son point d'envoi jusqu'au destinataire

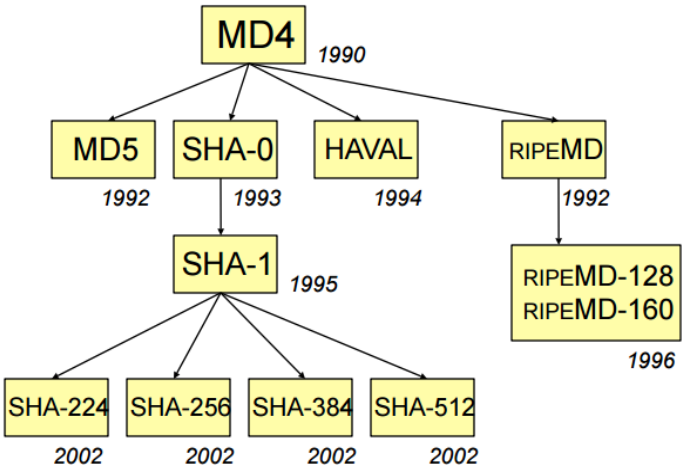


Figure 2.16 – Les différents standards de fonctions de hachage

3.1 MD5 (Message Digest 5)

Empreinte : 128 bits soit 32 caractères.
Le message est divisé en blocs de 512 bits
Applique un remplissage

- Ajouter un bit à 1.
- Ajouter autant de 0 que nécessaire pour arriver à 64 bits de moins que le prochain multiple de 512
- Ajouter un entier sur 64 bits donnant la longueur du message original.

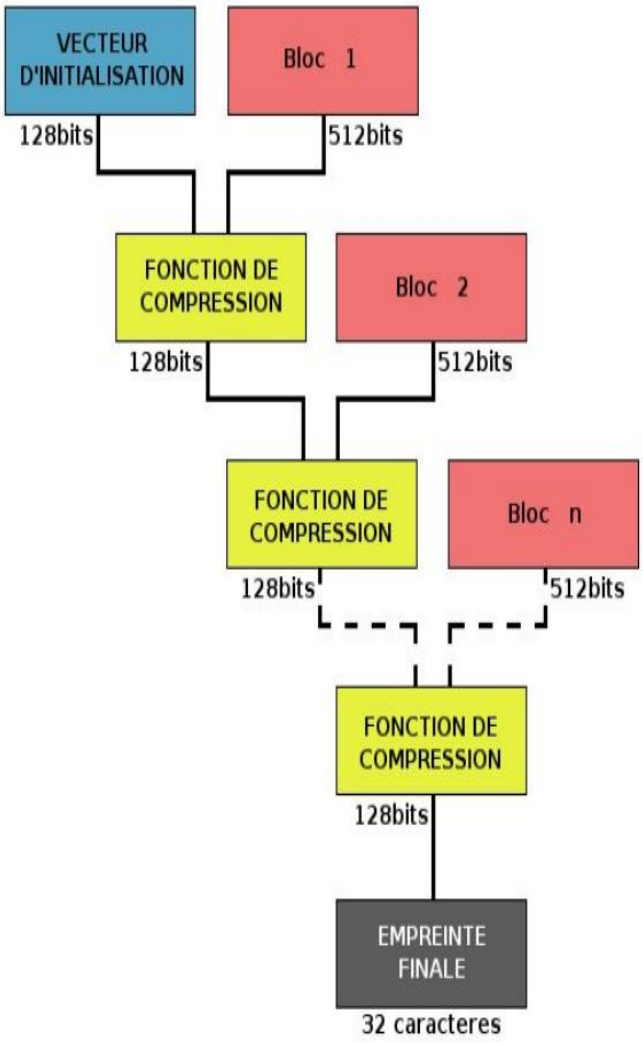


Figure 2.17 – Principe de fonctionnement de MD5

La fonction de hachage MD5 est devenue complètement obsolète :

- ✓ comportait trop de risques de collision
- ✓ n'est pas fiable

3.2 Secure Hash Algorithm SHA-1 et SHA-2

SHA-1 présente quelques améliorations:

- ➢ non inversible qui permet de caractériser un texte de manière très robuste.
- ➢ L'empreinte finale fait 160 bits soit 40 caractères.

Les fonctions de compression sont améliorées

SHA-2 : empreinte de 256, 384 voir 512 bits (résistance aux collisions)

3.3 Applications des fonctions de hachage

- Calcul d'empreinte
- Vérification d'intégrité
- Vérification d'authenticité
- Dérivation de clés :
  - Utilisation avec des clés secrètes
  - Hiérarchie de clés
- Générateur pseudo-aléatoire

4 Signature électronique

La cryptographie à clé publique existe depuis un certain temps déjà. L'essor Internet dans le monde, modifie de manière fondamentale les modes de communication. Ainsi, de plus en plus de documents qui étaient auparavant transmis sur papier sont maintenant échangés électroniquement. Cela présente de nombreux avantages : les communications entre une entreprise et ses clients sont beaucoup plus rapides, et la réduction de la quantité de papier nécessaire à ces communications ne peut être que bénéfique pour l'environnement.

La cryptographie à clé publique est extrêmement attrayante et riche en perspectives, intégrant à la fois le chiffrement et la signature numérique. Elle constitue une véritable percée par rapport aux systèmes cryptographiques à clé symétrique.

Une signature électronique est un ensemble de données informatiques générées à partir d'un document électronique qui permet d'authentifier ce document. Elle peut être intégrée au document ou séparée de celui-ci.

Une signature électronique doit garantir deux propriétés : elle doit identifier le signataire du document, et garantir que le document n'a pas été altéré depuis l'apposition de la signature.

La signature électronique repose sur deux familles d'algorithmes, qui seront utilisés de manière complémentaire :
Les algorithmes de chiffrement asymétriques et les fonctions de hachages:

Principe de signature et vérification

Voici comment se déroule la signature d'un document, et la vérification de cette signature :

Signature
Le signataire calcule le condensat du document à signer, puis il chiffre ce condensat à l'aide de sa clé privée.

Il crée ensuite la signature, qui peut être intégrée au document original ou enregistrée dans un fichier séparé. Cette signature est composée de l'empreinte signée (le condensat chiffré) et de son certificat.

Vérification
Le destinataire calcule le condensat du document reçu (en omettant la signature, si celle-ci est intégrée au document), et déchiffre l'empreinte signée, à l'aide de la clé publique contenue dans le certificat du signataire.

La sécurité de la plupart des schémas de signature est basée sur l'intractabilité d'un des problèmes suivants :

- le problème du logarithme discret dans un groupe multiplicatif (DLP),
- le problème du logarithme discret sur les courbes elliptiques (Elliptic Curve Discrete Logarithm Problem, ECDLP)
- le problème de factorisation en nombres premiers (FP).

8.1 Signature RSA

A partir du fait que la transformation de chiffrement RSA est une bijection, les signatures numériques peuvent être créées en inversant les rôles de cryptage et de décryptage.

- Algorithme de génération de clés pour le schéma de signature RSA

Chaque entité crée une clé publique RSA et une clé privée correspondante.
Chaque entité A doit faire ce qui suit:

1. Générer deux grands nombres premiers distincts p et q au hasard, ayant les deux la même taille.
2. Calculer n = pq et $\varphi(n) = (p - 1)(q - 1)$.
3. Sélectionnez un e entier aléatoire, $1 < e < \varphi(n)$, tels que pgcd $(e; \varphi(n)) = 1$.
4. Utiliser l'algorithme d'Euclide étendu pour calculer l'unique entier d, $1 < d < \varphi(n)$, tel que $ed \equiv 1 \pmod{\varphi(n)}$.
5. La clé publique est (n; e); la clé privée est d.

- Algorithme de génération et de vérification de signature RSA

Une entité A signe un message m. Toute entité B peut vérifier la signature A et récupérer le message m à partir de la signature.

1. Génération de signature. L'entité A doit faire ce qui suit:
    a. Calculer $\tilde{m} = R(m)$, un nombre entier dans l'intervalle [0; n - 1].
    b. Calculer $s = \tilde{m}^d \bmod n$.
    c. La signature de m pour A est s.

2. Vérification. Pour vérifier la signature de A et de récupérer le message m, B doit:
    a. obtenir la clé publique authentique de A : (n, e).
    b. Calculer $\tilde{m} = s^d \bmod n$.
    c. Vérifiez que $\tilde{m} = R(m)$, sinon, rejeter la signature.

Preuve que la vérification de signature fonctionne. Si s est une signature pour un message m, alors $s \equiv \tilde{m}^d \bmod n$ où $\tilde{m} = R(m)$. Du fait que $ed \equiv 1 \pmod{\varphi(n)}$, $se \equiv \tilde{m}ed \equiv \tilde{m} \pmod n$.

8.2 Signature ElGamal

Le schéma de signature ElGamal est un mécanisme de signature aléatoire. Il génère des signatures numériques avec appendice sur les messages binaires de longueur arbitraire, et requiert une fonction de hachage $h: \{0; 1\}^* \rightarrow Zp$ où p est un grand nombre premier.

- Algorithme de génération de clés pour le schéma de signature ElGamal

Chaque entité crée une clé publique et clé privée correspondante.
Chaque entité A doit faire ce qui suit:

1. Générer un grand nombre aléatoire p premier et un générateur g du groupe multiplicatif Zp*.
2. Sélectionner un entier aléatoire a, $1 \leq a \leq p - 2$.
3. Calculer $y = g^a \bmod p$
4. La clé publique de A est (p; g; y); la clé privée est a

- Algorithme de génération et de vérification de signature ElGamal

Une entité A signe un message m binaire de longueur arbitraire. Toute entité B peut vérifier cette signature en utilisant la clé publique de A.

1. Génération de signature. L'entité A doit faire ce qui suit:

   a. Sélectionner un k entier aléatoire secret, $1 \leq k \leq p - 2$, avec pgcd (k, p - 1) = 1.
   b. Calculer $r = g^k \bmod p$.
   c. Calculer $k^{-1} \bmod (p - 1)$.
   d. Calculer $s = k^{-1}\{h(m) - ar\} \bmod (p - 1)$.
   e. La signature de A pour m est le couple (r, s).

2. Vérification. Pour vérifier la signature (r, s) de A sur m, B doit effectuer les opérations suivantes:
   a. Obtenir la clé publique authentique de A (p;g; y).
   b. Vérifier que $1 \leq r \leq p - 1$; sinon, rejeter la signature.
   c. Calculer $v_1 = y^r r^s \bmod p$.
   d. Calculer h(m) et $v_2 = g^{h(m)} \bmod p$.
   e. Accepter la signature si et seulement si $v_1 = v_2$.

Preuve que la vérification de signature fonctionne :

Si la signature a été générée par A, alors $s \equiv k^{-1}\{h(m)-ar\} \pmod{p-1}$.
En multipliant les deux côtés par k on aura $ks \equiv h(m)-ar \pmod{p-1}$, et la réorganisation donne $h(m) \equiv ar + ks \pmod{p - 1}$.
Cela implique $g^{h(m)} \equiv g^{ar + ks} \equiv (g^a)^r r^s \pmod{p}$.
Ainsi, $v_1 = v_2$.

8.3 Signature DSA (Digital Signature Algorithm)

L'algorithme DSA est une variante du schéma ElGamal, c'est une signature numérique avec appendice. Le mécanisme de signature nécessite une fonction de hachage h: $\{0; 1\} \rightarrow$ Zq pour un certain entier q. Le DSA exige explicitement l'utilisation de l'algorithme de hachage sécurisé (SHA-1),

- Algorithme de génération de clés pour DSA

Chaque entité crée une clé publique et clé privée correspondante.
Chaque entité A doit faire ce qui suit:

1. Sélectionnez un nombre q premier tel que $2^{159} < q < 2^{160}$.
2. Choisissez t tel que $0 \leq t \leq 8$, et sélectionnez un nombre premier p où $2^{511+64t} < p < 2^{512+64t}$, avec la propriété que q divise (p - 1).
3. Choisissez un générateur α du groupe cyclique unique d'ordre q dans Zp*
   3.1 Choisir un élément $g \in Zp^*$ et calculer $\alpha = g^{(p-1)/q} \bmod p$.
   3.2 Si $\alpha = 1$ alors passez à l'étape 3.1.
4. Sélectionnez un nombre entier aléatoire a tel que $1 \leq a \leq q - 1$.
5. Calculer $y = \alpha^a \bmod p$.
6. La clé publique est $(p, q; \alpha; y)$; clé privée est a.

- Algorithme de génération et de vérification de signature DSA

Une entité A signe un message binaire m de longueur arbitraire. Toute entité B peut vérifier cette signature en utilisant la clé publique de A.

1. Génération de signature. L'entité A doit faire ce qui suit:

a. Sélectionner un entier aléatoire k secret; $0 < k < q$.
b. Calculer $r = (\alpha^k \bmod p) \bmod q$
c. Calculer $k^{-1} \bmod q$
d. Calculer $s = k^{-1}\{h(m) + ar\} \bmod q$.
e. La signature de m pour A est le couple (r, s)

2. Vérification. Pour vérifier la signature de A (r, s) sur m, B doit effectuer les opérations suivantes:

a. Obtenir la clé publique authentique $(p, q; \alpha; y)$.
b. Vérifier que $0 < r < q$ et $0 < s < q$; sinon, rejeter la signature.
c. Calculer $w = s^{-1} \bmod q$ et h (m).
d. Calculer $u_1 = w.\ h\ (m) \bmod q$ et $u_2 = rw \bmod q$.
e. Calculer $v = (\alpha^{u1} y^{u2} \bmod p) \bmod q$.
f. Accepter la signature si et seulement si v = r.

Preuve que la vérification de signature fonctionne :
Si (r, s) est une signature légitime de l'entité A sur le message m, alors $h(m) \equiv -ar + ks \pmod q$. En multipliant les deux côtés de cette congruence par w et en réarrangeant on a : $w.\ h(m) + arw \equiv k \pmod q$. Mais ceci est tout simplement $u_1 + au_2 \equiv k \pmod q$. L'élévation à la puissance des deux côtés de cette équation donne $(\alpha^{u1yu2} \bmod p) \bmod q = (\alpha^k \bmod p) \bmod q$. Par conséquent, r = v,

CHAPITRE 3

Problèmes potentiels de la Sécurité informatique – Contre Mesures

1 Introduction

La sécurité informatique est l'ensemble des techniques qui assurent que les ressources du système d'information (matérielles ou logicielles) d'une organisation sont utilisées uniquement dans le cadre où il est prévu qu'elles le soient.
Elle recouvre l'ensemble des techniques informatiques permettant de réduire au maximum les chances de fuites d'information, de modification de données ou de détérioration des services.
Elle consiste à un très grand nombre de méthodes, de technologies, et d'architectures permettant d'atteindre un certain niveau de protection.

Dans la suite de ce chapitre, nous allons essayer d'apporter des réponses aux questions suivantes :

- Que faut-il protéger ?
- Pourquoi protéger?
- Protéger Contre Qui ?

2 Ressources à protéger

Si on observe la sécurité d'une manière plus générale, elle est présente à plusieurs niveaux, en fonction des différentes portées de l'information comme c'est illustré sur la figure ci-dessous:
En conséquence, les éléments à protéger sont :

- Les données informatiques

- Les systèmes
- Les infrastructures réseaux

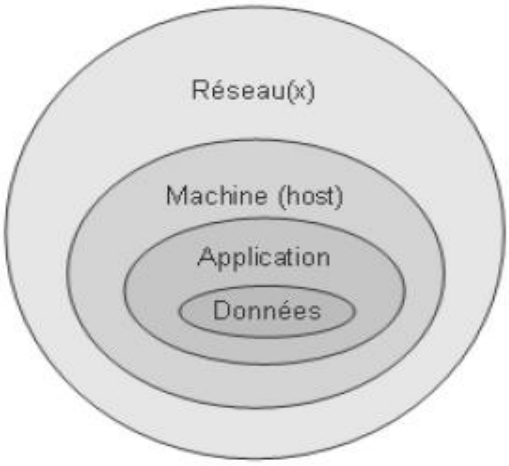


Figure 3.1 – Les différents Ressources à protéger

2.1 Les données informatiques

Quelle que soit la forme prise par l'information ou quels que soient les moyens par lesquels elle est transmise ou stockée, il faut qu'elle soit toujours protégée de manière appropriée.

Pour pouvoir protéger les données informatiques, il faut donc

- Protéger : Les communications
- Protéger : Les fichiers de données et Les bases de données

Les objectifs de base de la sécurité :

- **Confidentialité**

vise à assurer que seuls les sujets (les personnes, les machines ou les logiciels) autorisés aient accès aux ressources et aux informations auxquelles ils ont droit.
La confidentialité a pour objectif d'empêcher que des informations secrètes soient divulguées à des sujets non autorisés.

- **Intégrité**

l'intégrité vise à assurer que les ressources et les informations ne soient pas corrompues, altérées ou détruites par des sujets non autorisés.
L'objectif des attaques sur l'intégrité est de changer, d'ajouter ou de supprimer des informations ou des ressources.

- **Authenticité**

L'authenticité est l'assurance qu'un message, un ordre ou une information provient bien de la source (une personne, une machine, ou un programme) dont il prétend venir.
L'authenticité induit une preuve d'identité, elle est vérifiée à travers le processus d'authentification ou de signature.

- **Disponibilité**

La disponibilité vise à assurer que le système soit bien prêt à l'emploi, que les ressources et les informations soient en quelque sorte consommables, que les ressources ne soient pas saturées, que les informations, les services soient accessibles et que l'accès au système par des sujets non autorisés soit prohibé. L'objectif des attaques sur la disponibilité est de rendre le système inexploitable ou inutilisable.

- **Non-répudiation**

Le service de non-répudiation consiste à prévenir le refus, le démenti qu'un message ait été émis ou reçu ou qu'une action, transaction ait eu lieu. Cela permet de prouver par exemple qu'une entité est liée à une action ou à un événement.
La non-répudiation est basée sur une signature unique ou sur une identification qui prouve qui a créé le message. Pour assurer ce service, on peut faire appel à un algorithme de chiffrement à clé publique.

- **Traçabilité**

Une fonction qui consiste à repérer l'histoire des entités et leur mouvement. La traçabilité peut localiser par intermittence la position d'une personne ou d'un objet, peut dater des transactions, peut noter des renseignements sur des situations, le tout avec des attributs de sécurité. Cette fonction s'avère irremplaçable pour contrôler un objet, pour pister un suspect ou pour reconstituer un scénario lors d'une enquête informatique

2.2 Les systèmes

- Protéger : Les logiciels, les Systèmes d'exploitation, les outils de développement, les utilitaires...

2.3 Les infrastructures réseaux

- Protéger : Les serveurs informatiques, PC, portables, Matériels réseaux, Alimentation Électrique, Climatisation, etc.

3 Besoins de protection. Le Pourquoi ?

Malgré les facilités inhérentes au monde numérique et aux télécommunications, les problèmes:

- de conception,
- de mise en œuvre,
- de gestion et contrôle de l'informatique,
- de pannes,
- des erreurs,
- des incompétences,
- ou encore des catastrophes naturelles,

ont engendré un certain niveau d'insécurité aux infrastructures numériques, de sorte qu'elles sont devenues des cibles attrayantes pour la réalisation de crimes économiques via les nouvelles technologies
De ce fait, les technologies de l'Internet ont permis aujourd'hui toute sorte d'infractions :

- Vol d'informations et du savoir faire (dans un contexte de haute technologie notamment)

- sabotage d'informations,
- atteintes au copyright, au droit d'auteur, à la violation du secret professionnel, de l'intimité numérique, de la propriété intellectuelle,
- dissémination de contenus illégaux,
- espionnage industriel,
- atteinte à l'image de marque
- diffusion de fausses informations,
- dénis de service,
- fraudes diverses, etc.

4 Menaces, attaques et adversaires

Contre qui faut-il se protéger ?

- Les menaces
- Les différents types de pirates (attaquants)
- Les différentes attaques.

4.1 Définitions

4.1.1 Vulnérabilité

Une vulnérabilité est une faille du système : Faiblesse dans les procédures de sécurité, les contrôles administratifs, les contrôles internes d'un système, qui pourrait être exploitée pour obtenir un accès non autorisé au système d'information, à un de ses services, à des informations ou à la connaissance de leur existence et de permettre de porter atteinte à la confidentialité, à l'intégrité et à la disponibilité du système d'information et de ses informations.

Les vulnérabilités peuvent être à plusieurs niveaux :

- **Dans la conception**

- Matériel
- Protocole
- Architecture (Système, Réseau, …)
- Logiciel (OS, application, …)

- **Dans l'implémentation**

- Matériel
- Protocole
- Architecture (Système, réseau …)
- Logiciel (OS, application, …)

- **Dans la configuration, ou l'exploitation**

- Équipement (Routeurs, Firewalls, Serveur, …)
- Logiciel (OS, application, …)

4.1.2 Menace - Risque

- Une menace est une violation potentielle de la sécurité
- Vulnérabilité + Menace = Risque/Contre-mesure

Les menaces engendrent des risques et coûts humains et financiers : perte de confidentialité de données sensibles, indisponibilité des infrastructures et des données, dommages pour le patrimoine intellectuel et la notoriété. Les risques peuvent se réaliser si les systèmes menacés présentent des vulnérabilités.

a) Type de menaces :

Avec la popularité grandissante des réseaux, des échanges de données, et donc des transmissions entre individus, de nombreuses menaces émergèrent. Parmi celles-ci, on trouve diverses catégories :

– Les menaces accidentelles
– Les menaces intentionnelles

– **Menaces accidentelles**

Les menaces accidentelles ne supposent aucune préméditation. Dans cette catégorie, sont repris les :

- bugs logiciels,
- les pannes matérielles,
- et autres défaillances incontrôlables
- Action exécutée par erreur
- publicité, invitation

etc…

– **Menaces intentionnelles**

Par opposition à la précédente, elle est le fait d'un acte délibéré,
Action exécutée pour violer la sécurité

- **Menace active**

Menace de modification non autorisée de l'état du système d'information (Dommage ou altération)

- **Menace Passive**

Menace de divulgation non autorisée des informations, sans que l'état du système d'information soit modifié (Écoutes, lecture de fichiers, …)

- **Menace Physique**

Menace l'existence ou l'état physique du système d'information (Sabotage, incendie volontaire, vol, …)

b) Catégories de menaces intentionnelles

- L'espionnage (Obtention d'informations)

- Le vol (Obtention d'informations ou de ressources)
- La perturbation (Affaiblir le système d'information)
- Le sabotage (Mise hors service du système d'information)
- Le chantage (Gain financier, menace de sabotage, …)
- La fraude physique (récupération de bandes, listings, …)
- Accès illégitimes (usurpation d'identité)

c) Motivations des de menaces intentionnelles

✓ **Menaces à caractère stratégiques**

➡ Pour un état

- Le secret défense et la sûreté de l'état
- Le patrimoine scientifique, technique, économique, diplomatique
- La disponibilité des systèmes d'information et le fonctionnement des institutions

➡ Pour une entreprise

- Informations concernant ses objectifs et son fonctionnement
- Les clients, procédés de fabrication, recherche et développement

➡ Catégories de menace

Espionnage, vol, perturbation, sabotage, fraude physique, accès illégitimes …

➡ Type d'attaquants

Espions (Entreprises, Gouvernements)

Menace à caractère idéologique

Le combat pour les idées est incessant et peut être le moteur d'actes les plus extrêmes

➡ Idéologie politique, raciale, religieuse, économique, …

➡ Catégorie de menace

Espionnage, vol, perturbation, sabotage, chantage, fraude physique, accès illégitimes, …

➡ Type d'attaquants

Militants, Terrori

✓ **Menace à caractère financier**

➡ Gain pour l'attaquant

Financier, technologique, …

➡ Pertes pour la victime

Entraînant un gain pour l'agresseur : parts de marché, déstabilisation du concurrent, …

➡ Catégorie de menace

Espionnage, vol, perturbation, sabotage, chantage, fraude physique, accès illégitimes, …

➡ Type d'attaquant

Espions (concurrent ou pour le compte de)

✓ **Menace à caractère vengeur**

Souvent l'expression d'un employé brimé ou licencié qui peut possédé une très bonne connaissance du système d'information

➡ Catégorie de menace

Vol, perturbation, sabotage, accès illégitimes, …

➡ Type d'attaquant

- Personnes extérieures en désaccord avec l'entité
être un client, un fournisseur, un intervenant, …
- Les employés malveillants
Ont souvent une bonne connaissance de l'entreprise
Administrateurs, informaticiens, …

4.1.3 Attaques

Une attaque est une concrétisation d'une menace.
Une menace peut être concrétisée de différentes manières (c'est-à-dire différentes attaques)

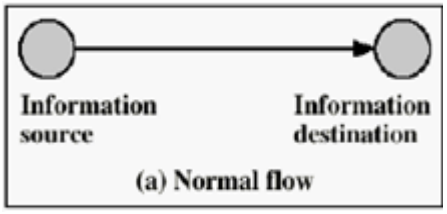
On peut classer les attaques en 2 types :

- Les attaques passives : Interception
- Les attaques actives : Interruption, Modification, Fabrication

a) Attaques Passives

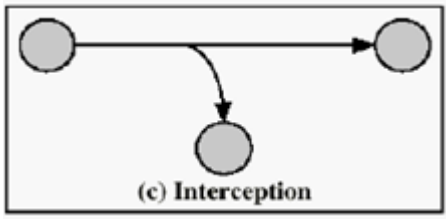
Elles représentent une menace contre la **confidentialité de l'information :**

Une information sensible parvient à une personne autre que son destinataire légitime : **Difficile à détecter**



- **Interception :**

vise la confidentialité des informations



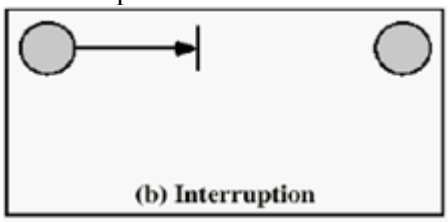
b) Attaques Actives

Représentent une menace contre l'intégrité de l'information :
Destruction, modification, fabrication, interruption.

L'information reçue est interprétée comme provenant d'une personne autre que son véritable auteur.

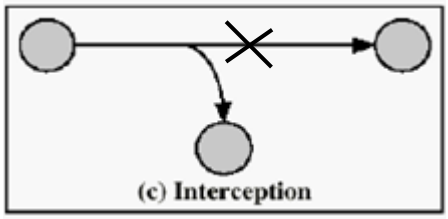
- **Interruption:**

vise la disponibilité des informations



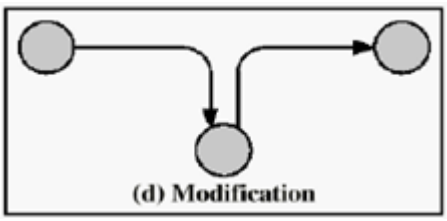
- **Interception + destruction:**

vise la confidentialité des informations



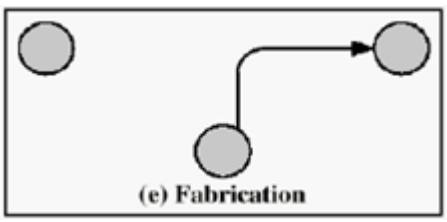
- **Modification:**

vise l'intégrité des informations



- **Fabrication:**

vise l'authenticité des informations



c) Exemples d'Attaques

➢ **Intrusions**

✓ Recherche de mots de passe

- ➡ Dictionnaire (essai de mots de passe contenu dans un dictionnaire)
- ➡ Écoute du réseau
- ➡ par Brute force (essai de toutes combinaisons de caractères)

✓ Le Spoofing

(Utiliser l'identité ou l'adresse de la victime)

- ✓ Les sniffers (logiciels d'Ecoute de réseaux)
- ✓ Les scanners (logiciels d'analyse de réseaux)
- ✓ Les exploits (Attaque d'un site Web)

✓ Denis de service (DOS : Denial Of Service)

- ✓ Bombes logiques (virus)
- ✓ Le Flooding (Inondation contraire de diffusion)
- ✓ Le Spamming (Spams)

✓ Denis de service distribué (DDOS)

- ✓ Amplification des DOS

➢ **Le phreaking.**

Contraction des mots anglais phone (téléphone) et freak (monstre) désignant le piratage de lignes téléphoniques

- ➡ C'est une technique frauduleuse permettant l'utilisation gratuite de ressources téléphoniques depuis l'extérieur.

➢ **Le phising**

Contraction des mots anglais «fishing», en français pêche,

- ➡ Technique frauduleuse utilisée par les pirates informatiques pour récupérer des informations (généralement bancaires) auprès d'internautes.
- ➡ Technique d'«ingénierie sociale» c'est-à-dire consistant à exploiter non pas une faille informatique mais la «faille humaine» en dupant les internautes par le biais d'un courrier électronique semblant provenir d'une entreprise de confiance.

➢ **Le Hoax**

- ➡ Courrier électronique propageant une fausse information et poussant le destinataire à diffuser la fausse nouvelle à tous ses proches ou collègues.
- ➡ Les conséquences
  - ➡ L'engorgement des réseaux et des serveurs de messagerie,
  - ➡ Une désinformation, c'est-à-dire faire admettre à de nombreuses personnes de faux concepts ou véhiculer de fausses rumeurs,
  - ➡ La perte de temps, tant pour ceux qui lisent l'information, que pour ceux qui la relaye ;

- La dégradation de l'image d'une personne ou bien d'une entreprise,

A force de recevoir de fausses alertes les usagers du réseau risquent de ne plus croire aux vraies

- **Les Virus**

Programme informatique situé dans le corps d'un autre, qui, lorsqu'on l'exécute, se charge en mémoire et exécute les instructions que son auteur a programmé

Différents types :

- ✓ Les vers
- ✓ Les trojans
- ✓ Les bombes logiques
- ✓ Les spywares

- **Les vers**

  - ✓ Programme qui peut s'auto-reproduire et se déplacer à travers un réseau en utilisant les mécanismes réseau, sans avoir réellement besoin d'un support physique ou logique pour se propager
  - ✓ Les vers actuels se propagent principalement grâce à la messagerie grâce à des fichiers attachés contenant des instructions permettant de récupérer l'ensemble des adresses de courrier contenues dans le carnet d'adresse et en envoyant des copies d'eux-même à tous ces destinataires.

- **Les chevaux de Troie (Trojans)**

Programme (en anglais trojan horse) caché dans un autre qui exécute des commandes, et qui généralement donne un accès à la machine sur laquelle il est exécuté en ouvrant une porte dérobée (en anglais backdoor) :
Actions:

- Vol de mots de passe
- Copie de données
- Exécution d'action nuisible

- **Les bombes logiques**

Sont appelés bombes logiques les dispositifs programmés dont le déclenchement s'effectue à un moment déterminé en exploitant la date du système, le lancement d'une commande, ou n'importe quel appel au système. Ainsi ce type de virus est capable de s'activer à un moment précis sur un grand nombre de machines (on parle alors de bombe à retardement ou de bombe temporelle

Les bombes logiques sont généralement utilisées dans le but de créer un déni de service en saturant les connexions réseau d'un site, d'un service en ligne ou d'une entreprise.

- **Les spywares ou espiogiciels**

  - ✓ Programme chargé de recueillir des informations sur l'utilisateur de l'ordinateur sur lequel il est installé (on l'appelle donc parfois mouchard) afin de les envoyer à la société qui le diffuse pour lui permettre de dresser le profil des internautes (profiling).

- ✓ Keyloggers : Dispositif chargé d'enregistrer les frappes de touches du clavier et de les enregistrer, à l'insu de l'utilisateur. Il s'agit donc d'un dispositif d'espionnage.

5 Contre-Mesures de sécurité techniques

Pour atteindre les objectifs de sécurité décrits au début de ce chapitre et contrecarrer toutes les attaques précitées, il est nécessaire de mettre en œuvre une politique de sécurité, applicable à l'ensemble des entités à l'intérieur du système d'information.
La Politique de Sécurité des Systèmes d'Information (PSSI) indique l'ensemble des mesures techniques, organisationnelles ou juridiques à prendre et à mettre en place afin :
– d'empêcher la détérioration, l'utilisation anormale ou la pénétration des systèmes et réseaux;
– de détecter toute atteinte, malveillante ou non, à l'intégrité, la disponibilité et la confidentialité des informations ;
– d'intervenir afin d'en limiter les conséquences

Responsable de la Sécurité du Système d'Information (RSSI)
Le RSSI est chargé de la définition et de la mise en œuvre de la politique de sécurité. Il possède en outre un rôle stratégique d'information, de conseil et d'alerte de la direction générale sur les risques en matière de sécurité informatique.
La fonction du RSSI est essentiellement de manager et consiste à encadrer une équipe d'ingénieurs et de techniciens d'exploitation, dont il organise et contrôle le travail. Il est chargé de l'évaluation et de la gestion des risques liés au système d'information.
Dans le volet des mesures techniques d'une PSSI, le RSSI s'appuiera, entre-autres, sur les points suivants :

- Cryptographie
- Contrôle d'accès
- Pare-feu (Firewall)
- Zone Démilitarisée DMZ
- IDS (Détection d'Intrusions)
- Log (Journalisation)
- Audit
- IPsec VPN (Réseaux Privées Virtuels)
- SSL (Sécurité des Services)
- Anti-virus
- Plan de sauvegarde

CHAPITRE 4

Mesures de sécurité basées sur la cryptographie – Contrôle d'accès

1 Cryptographie

La mise en œuvre de techniques de cryptographie permet de réaliser la confidentialité des données, de vérifier leur intégrité et d'authentifier des entités.
3 fonctions :

- le chiffrement
- le hachage
- la signature

1.1 Le chiffrement

Comme il a été déjà vu dans le chapitre 2, il existe deux grands types de système de chiffrement de données :

- le chiffrement symétrique (à clé secrète)
- et le chiffrement asymétrique (à clé publique).

Divers algorithmes de chiffrement existent. Quel que soit leur mode opératoire (symétrique ou asymétrique), ils reposent sur l'usage de clés. Généralement leur degré de robustesse est lié à la capacité à gérer les clés de chiffrement de manière sécurisée, à la longueur de la clé, de la sécurité de la plateforme matérielle et logicielle dans laquelle les algorithmes de chiffrement sont implantés et s'exécutent.

Dans le cas d'un système de chiffrement symétrique, l'émetteur et le récepteur doivent posséder et utiliser la même clé secrète pour rendre confidentielles des données échangées, ceci pose le problème de la gestion et de la diffusion des clés secrètes.
Considérer N utilisateurs (une clé pour chaque couple) !!!!

En contre partie, un système de chiffrement asymétrique est basé sur l'usage d'un couple unique de deux clés ('une clé publique et d'une clé privée). Seule la clé publique peut être connue de tous, tandis que la clé privée doit être confidentielle et traitée comme un secret.
L'émetteur chiffre un message avec la clé publique du destinataire du message et le destinataire le déchiffre avec sa clé privée. Les principaux algorithmes de chiffrement à clé publique, utilisent le plus souvent des clés de longueur variant de 512 à 1024 bits, voire 2048 bits (RSA, DH, El Gamal).
Considérer N utilisateurs (une paire de clés pour chaque utilisateur) !!!!
Inconvénient Vitesse : lent

1.2 Le hachage

Déjà vu au chapitre 2.

- Utilisée seule, elle permet de vérifier l'intégrité d'un message.

- Associé à une clé symétrique, elle permet le calcul d'un sceau ou MAC (Message Authentification Code), pour assurer :
  - Intégrité des données
  - Authentification de la source

- Associé à un chiffrement asymétrique, elle permet le calcul de signatures, pour assurer :
  - Intégrité des données
  - Authentification de la source
  - Non-répudiation de la source

1.3 La signature

Déjà vu au chapitre 2.
La norme ISO 7498-2 définit la signature numérique comme des données ajoutées à un message, ou transformation cryptographique d'un message, permettant à un destinataire de :

-authentifier l'auteur d'un document électronique
-garantir son intégrité
-protéger contre la contrefaçon (seule l'expéditeur doit être capable de générer la signature) → non-répudiation.

1.4 Chiffrement Hybride

Les algorithmes symétriques sont beaucoup plus rapides que les algorithmes asymétriques. Malheureusement, d'une part l'échange de clés symétriques est plus délicat (il faut un canal confidentiel et authentique) et d'autre part la cryptographie symétrique ne répond pas à tous les critères que l'on peut attendre d'un schéma de signature (on ne pas distinguer si c'est l'expéditeur ou le destinateur qui a signé: problème de répudiation).

En utilisant un système hybride, on profite des avantages de la cryptographie asymétrique tout en conservant l'efficacité de la cryptographie symétrique (c'est la solution choisie par PGP et SSL qui chiffrent la clé symétrique avec une clé asymétrique).

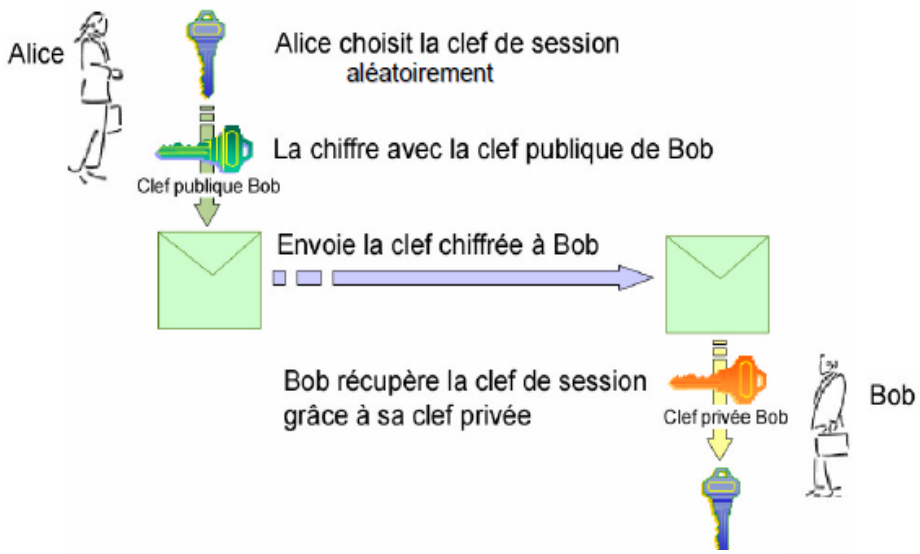


Figure 4.1 – Echange de clé de session

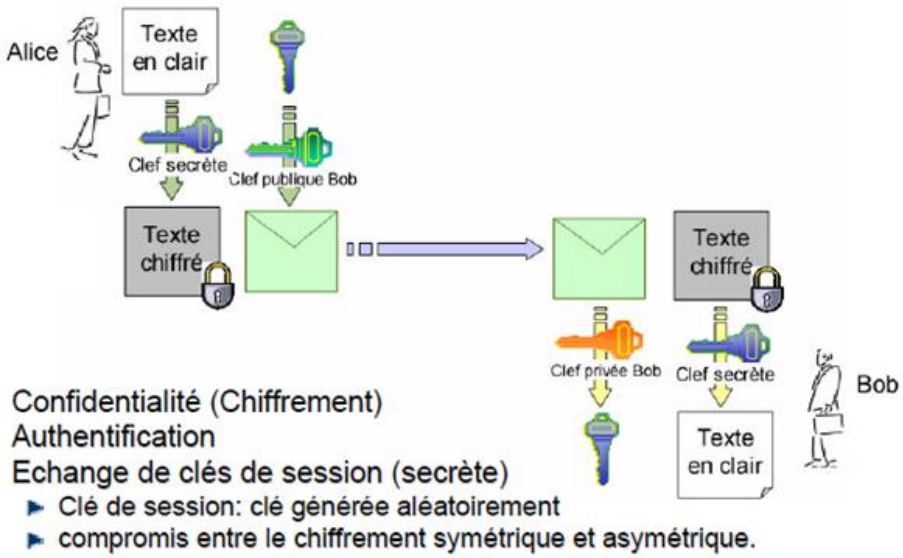


Figure 4.2 – Echange de clé de session + Envoi du message chiffré

2 Contrôle d'accès

- Toute organisation s'appuyant sur un système d'information a besoin de mécanismes sécurisés pour les connexions à son système.
- Le contrôle d'accès est la base de ces mécanismes. Il permet de garantir que les utilisateurs/les processus ne peuvent accéder aux informations et aux ressources auxquelles ils ne sont pas autorisés à accéder.
- Le contrôle d'accès est l'élément clé pour assurer :

– la confidentialité et l'intégrité des données : seules les personnes identifiées et authentifiées peuvent accéder aux ressources et les modifier s'ils sont habilités à le faire

– la non-répudiation : seules les entités identifiées et authentifiées peuvent être tenu responsables d'une certaine action (par exemple la preuve de l'origine d'un message ou d'une transaction bancaire…).

Un mécanisme de contrôle d'accès est basé sur 3 processus :

1. Identification
2. Authentification
3. Autorisation (permissions ou droits d'accès)

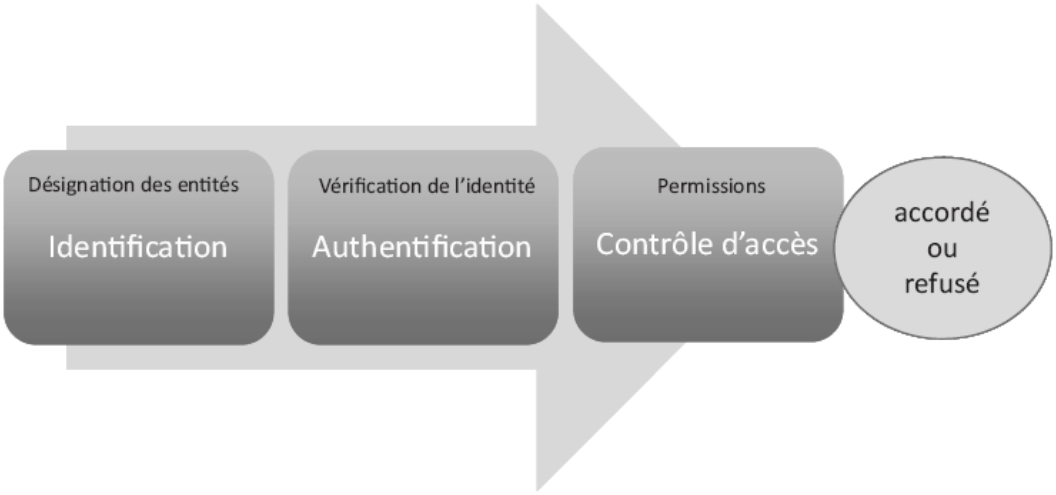


Figure 4.3 – Les mécanismes de contrôle d'accès

Sur la base d'une identification authentifiée, le mécanisme de contrôle d'accès accorde ou refuse, en fonction du profil de l'utilisateur, l'accès aux ressources sollicitées.

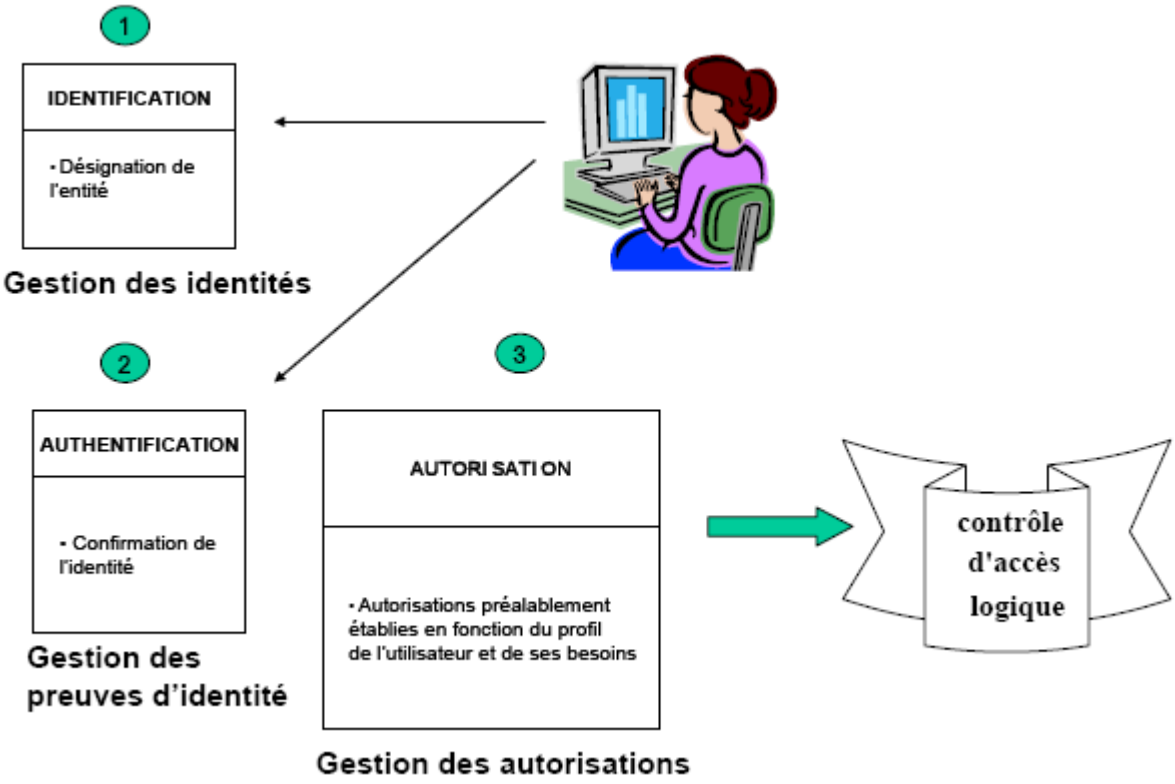


Figure 4.4 – Illustration des processus de contrôle d'accès

2.1 Identification

L'identification consiste à associer à une personne physique, un composant ou un élément logiciel une identité numérique.

Il s'agit de donner un nom à une entité (désignation de l'entité) qui relève donc de la gestion des identités.

Exemples :

- Personne physique : Login (nom d'utilisateur)
- Machine : Nom DNS d'un serveur, adresse IP
- Téléphone : Numéro de téléphone

….

S'identifier c'est déclarer son identité

2.2 Habilitations - Autorisation

Les habilitations correspondent aux droits d'accès associés à un utilisateur (accéder, créer, modifier, désactiver, révoquer, supprimer, signer, …)

Elles peuvent concerner une ou plusieurs ressources

Elles sont définies en fonction du profil de l'utilisateur:

- De son contexte (au sein de l'entreprise, mission, …)
- De sa fonction, son poste dans l'organisation
- Son rôle opérationnel (Exemple : le RSSI ou l'administrateur d'une application)

Le profil regroupe toutes les informations nécessaires aux décisions d'autorisation d'accès. Il doit être défini avec soin et résulte de la définition de la politique de gestion des accès.

Figure 4.5 – Mécanisme d'autorisation

2.3 Authentification

L'Authentification est la vérification d'informations relatives à une personne ou à un processus informatique. L'authentification complète le processus d'identification dans le sens où l'authentification permet de prouver une identité déclarée. Dans un serveur, un processus de contrôle valide l'identité et après authentification, donne l'accès aux données, applications, bases de données, fichiers ou sites Internet. Dans le cas contraire, l'accès est refusé.

Il existe une différence toute simple entre identification et authentification : c'est la preuve.

Une identification s'appuie sur une simple déclaration de l'identité. Cette identité n'est pas supposée secrète. C'est une donnée publique.

L'authentification s'appuie sur un élément de preuve comme une clé partagée ou une clé privée asymétrique. L'authentification permet de s'assurer avec un niveau de confiance raisonnable de l'identité de l'utilisateur.
S'authentifier c'est prouver son identité
L'Authentification est la vérification de l'identité
L'authentification peut se faire de différentes manières, et notamment par la vérification de :

1. Ce que l'utilisateur sait : un mot de passe par exemple
2. Ce que l'utilisateur sait faire : une signature manuscrite sur écran tactile
3. Ce que l'utilisateur est : une caractéristique physique comme une empreinte digitale
4. Ce que l'utilisateur possède : une carte à puce par exemple

Le choix de telle ou telle technique dépend en grande partie du contexte et d'application : authentification de l'expéditeur d'un email, authentification d'un utilisateur qui se connecte à distance, authentification d'un administrateur au système, authentification des parties lors d'une transaction,…

- Authentification multi-facteurs
- Les éléments précités (élément que l'on sait , que l'on sait faire, que l'on possède , ou que l'on est) sont appelés **des facteurs d'authentification**.
- Pour s'authentifier, un utilisateur fournit en général au moins 2 éléments :
- - son identifiant qui permet son identification.
- - et un ou plusieurs facteurs d'authentification.
- Dès que plusieurs facteurs d'authentification entrent en jeu, nous parlons d'authentification **multi-facteurs.**
- L'authentification **multi-facteurs** permet de renforcer le processus d'authentification, on parle alors d'authentification forte.

2.3.1 Méthodes d'authentification

Il existe plusieurs méthodes d'authentifications qui sont utilisées aujourd'hui :

- Mot de passe statique
- Mot de passe statique stocké dans une carte magnétique activée par code PIN
- Mot de passe dynamique généré par un outil logiciel
- Mot de passe dynamique généré par un outil matériel
- Certificat X.509 dans le navigateur de l'ordinateur
- Certificat X.509 dans une clé USB
- Certificat X.509 dans une carte à puce
- Biométrie

a) Authentification par mots de passe statiques

- Les mots de passe pris dans leur ensemble sont le moyen d'authentification le plus répandu à ce jour. On distingue deux catégories : les mots de passe statiques et les mots de passe dynamiques.

- Les mots de passe statiques sont des mots de passe qui restent identiques pour plusieurs connexions sur un même compte. Cette technique d'authentification est la plus utilisée dans les entreprises (coût, facilité de mise en œuvre et d'utilisation) mais aussi la moins robuste.

- Les entreprises devraient restreindre l'usage des mots de passe statiques à une authentification locale (pas sur réseau) car les attaques sur les mots de passe statiques sont nombreuses et faciles à mettre en pratique.

Menaces et parades

➢ **Crakage du mot de passe :**

- Attaque par force brute : essayer toutes les combinaisons possibles de symbole jusqu'à une certaine longueur

– Attaque par dictionnaire

Parades :

– Eviter les mots de passe évidents
– Augmenter la longueur du mot de passe
– Augmenter la complexité du mot de passe
– Renouvellement forcé de mots de passe (contrôle de réutilisation de mots de passe)
– Limitation des tentatives d'authentification
– Informer l'utilisateur (dernière authentification réussies, échecs)

➢ **Capture du mot de passe**

- Attaque par écoute du réseau (sniffing) : les mots de passe circulent en clair sur le réseau

- Attaque Man in the middle : L'attaquant se fait passer pour le serveur d'authentification

Exemple : l'attaquant a installé un programme simulant un écran de login; lorsqu'un utilisateur essaie de s'authentifier, le programme récupère le mot de passe, envoie un message d'erreur et interrompt son exécution.

Parades :

– Indiquer le nombre d'échecs d'authentification
–Authentification mutuelle : le système doit également s'authentifié auprès de l'utilisateur

➢ **Compromission du fichier de mot de passe**

- Les mots de passe des utilisateurs sont stockés dans un fichier

- Divulgation ou modification du contenu du fichier de mots de passe

Parades :

- Protection cryptographique :

– Fonction à sens unique (hachage)
– Mot de passes chiffrés

- Protection en écriture : sinon il suffit à l'attaquant de modifier le mot de passe

- Contrôle d'accès : seuls des utilisateurs privilégiés peuvent avoir accès à des fichiers de mots de passe

b) Mot de passe statique stocké dans une carte magnétique et protégé par code PIN (Personal Identification Number)

Ce système exige la présence d'un lecteur de carte magnétique installé sur le serveur d'authentification.
Le mot de passe statique est stocké sur la piste magnétique de la carte (sous forme chiffré par le code confidentiel : code PIN)

- le serveur demande le code confidentiel à l'utilisateur
- l'utilisateur tape son code
- le serveur décrypte le mot de passe chiffré
- si le mot de passe est valide, l'accès est autorisé

Figure 4.6 – Authentification par carte magnétique

Avantages :

- robustesse du mot de passe (possibilité de choisir un mot de passe aléatoire et comprenant des caractères spéciaux)
- pas de nécessité de mémoriser le mot de passe

Inconvénients :

- vol, perte ou oubli de la carte
- carte partageable
- durée de vie du mot de passe souvent trop longue : nécessité de renouveler régulièrement l'enregistrement dans la carte
- mot de passe rejouable (copie des mots de passe sur d'autres cartes + vol PIN)

c) Authentification Forte

Une authentification est considérée comme forte si elle associe au moins deux facteurs d'authentification
Il existe plusieurs méthodes d'authentification forte basées essentiellement sur :

1. La cryptographie symétrique (**mots de passe dynamiques)**
2. La cryptographie asymétrique (**certificats numériques**)
3. La biométrie

- **Mots de passe dynamiques**

Les mots de passe dynamiques sont des mots de passe à usage unique (**One Time Password : OTP**)
Ils ont une durée de vie limitée à quelques secondes. Il n'y a ainsi aucun intérêt à intercepter des mots de passe à usage unique : ils ne peuvent pas être réutilisés.

On distingue deux variantes:

- les OTP générés par outil matériel
- les OTP générés par outil logiciel

OTP générés par outil matériel :

Il y a principalement deux types :

1. Les cartes à puce cryptographique qui nécessitent un lecteur.
2. Les clés USB ou Tokens (avec puce) qui n'ont pas besoin de lecteur et peuvent se connecter directement au PC avec les pilotes appropriés.

Figure 4.7 – Authentification par carte à puce

OTP générés par outil logiciel :

Il s'agit ici de logiciels OTP embarqués sur des supports mobiles

(smartphone, PDA, Laptop,….)

Figure 4.8 – OTP : clés USB

La technologie OTP est fondée sur un secret partagé unique (cryptographie symétrique). La carte à puce/token contient le secret. Le serveur d'authentification contient le même secret.
Grâce au partage de ce secret commun il est alors possible de générer des mots de passe à usage unique (One Time Password).

Il existe deux modes de fonctionnement :

- le mode asynchrone et le mode synchrone

Mode asynchrone

Par asynchrone on entend le fait que le serveur d'authentification va commencer par envoyer à l'utilisateur un "challenge" (nombre aléatoire). L'utilisateur va communiquer ce challenge à son générateur d'OTP (logiciel ou matériel). En faisant entrer son code PIN => deux facteurs.
À partir de ce challenge, et du secret partagé, le générateur va calculer une réponse:
le mot de passe dynamique ou à usage unique (l'OTP), et l'envoyer au serveur pour vérification.

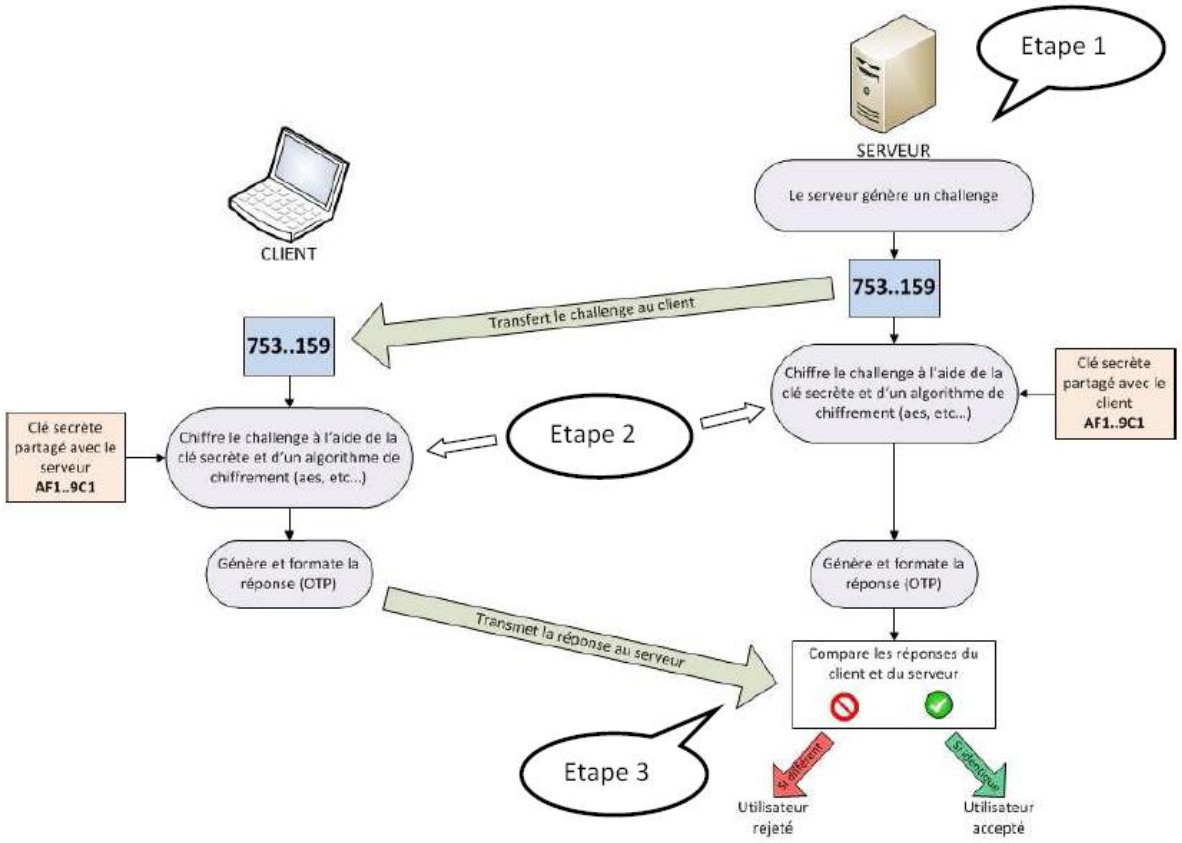


Figure 4.9 – OTP : fonctionnement du mode asynchrone

Problèmes associés

- Modifications nécessaires : Le fait d'utiliser une authentification asynchrone entraîne le besoin de fournir à l'utilisateur le challenge afin qu'il puisse calculer l'OTP correspondant. Ceci peut poser des problèmes pour certaines applications ou protocoles pour lesquels il n'est pas possible de demander un challenge. Pour utiliser un OTP asynchrone comme moyen d'authentification il est donc indispensable que :

- l'application soit modifiable pour afficher le challenge
- le protocole de communication utilisé soit capable de transporter le challenge au client
- l'application puisse être modifiée pour vérifier l'OTP une fois celui-ci saisi par l'utilisateur

Eviter de répéter le même challenge : Pour les OTP asynchrones le même challenge engendre toujours la même réponse. Si l'on suppose qu'un attaquant écoute tous les échanges pour essayer d'avoir accès au service à la place de l'utilisateur on comprend alors que si le même challenge est utilisé deux fois de suite l'attaquant va pouvoir la deuxième fois se faire passer pour l'utilisateur légitime. Une partie de la sécurité associée aux OTP asynchrones dépend donc de l'unicité du challenge. Ainsi un challenge qui serait trop court (donc qui reviendrait trop souvent) fera que le même OTP sera souvent utilisé ce qui donne plus de chances à l'attaquant.

Mode synchrone

Les OTP synchrones contrairement aux OTP asynchrones ne nécessitent pas de challenge. Ainsi il suffit de générer un OTP synchrone et de le fournir directement au serveur pour vérification.
L'OTP généré est toujours différent et peut être synchronisé selon différentes techniques :

- basé sur le temps
- basé sur un compteur
- basé sur le temps et un compteur

-basé sur le temps

Le dispositif utilisateur calcule l'OTP à partir de l'heure courante et du secret partagé (clé symétrique).
- Ce calcul se fait généralement après que l'utilisateur ait entré son code PIN => deux facteurs.

- L'algorithme utilisé doit être robuste et ne pas permettre de retrouver le secret à partir des OTP produits.
- L'OTP est envoyé ensuite au serveur d'authentification pour vérification. Il doit correspondre à celui généré par le serveur. Pour cela, le dispositif utilisateur et le serveur doivent être synchronisés.

-basé sur un compteur

Dans ce cas, on utilise un compteur et le secret partagé pour générer le mot de passe usage unique. Le compteur est incrémenté à chaque nouvelle génération d'OTP, ce qui aboutit à un OTP différent à chaque fois. Les deux parties (utilisateur et serveur) doivent se synchroniser sur le même compteur. Un code PIN est utilisé comme deuxième facteur.

-basé sur le temps et un compteur

Combinaison des deux techniques précédentes.

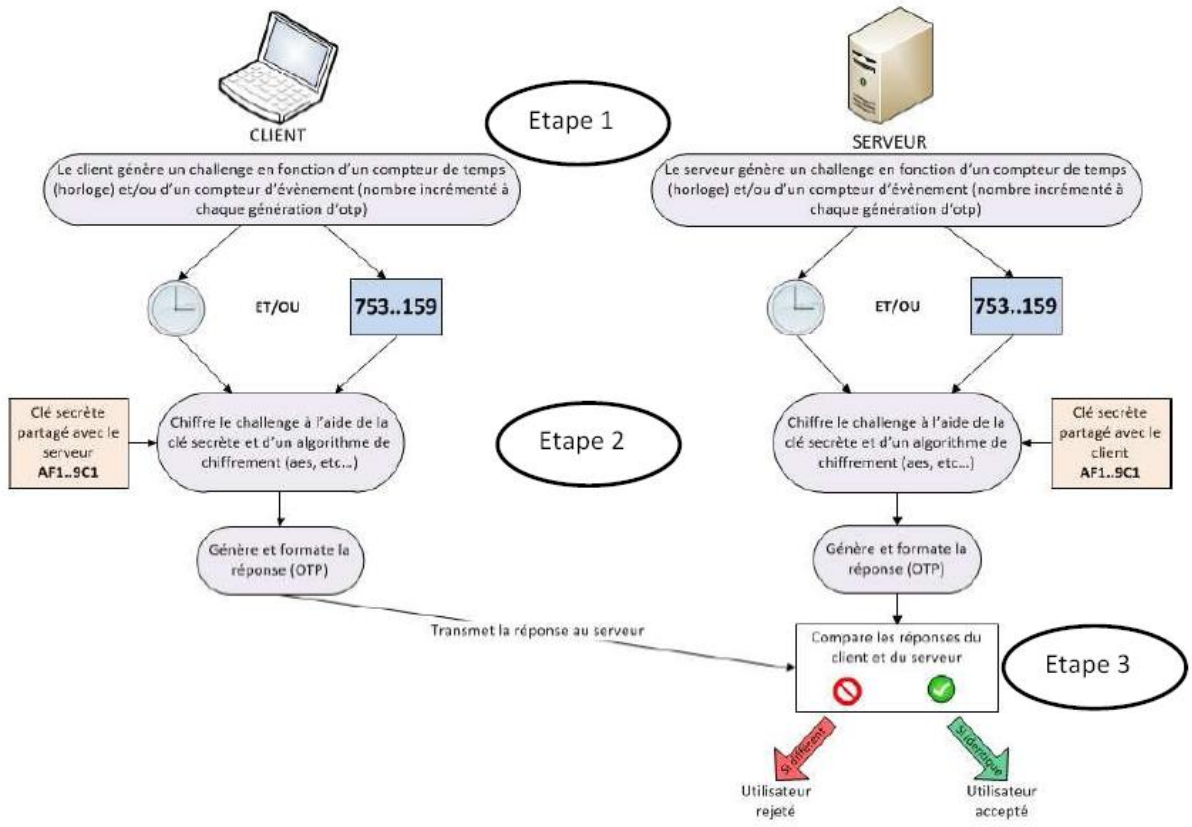


Figure 4.10 – OTP : fonctionnement du mode synchrone

- **Certificats numériques**

Un certificat numérique est une pièce d'identité électronique qui permet de lier l'identité de l'utilisateur /machine à sa clé publique.
Le certificat numérique est signé par une autorité de certification digne de confiance.

Processus de certification
Le processus de certification repose sur une infrastructure à clé publique (Public Key Infrastructure : **PKI**) faisant intervenir 3 acteurs :

- l'autorité de certification,
- l'autorité d'enregistrement
- et l'annuaire LDAP.

❑ **L'autorité de certification (AC)**

Une Autorité de Certification (Certification Autority, CA) est responsable de la délivrance, la révocation et le renouvellement de certificats numériques. Tous les certificats numériques émis par une CA sont signés avec la clé privée de cette CA afin d'assurer l'authenticité du certificat.

❑ **L'autorité d'enregistrement (AE)**

L'autorité d'enregistrement est responsable de la création de la paire de clés privée /publique de l'utilisateur

❑ **L'annuaire LDAP)**
L'annuaire LDAP permet de publier les certificats numériques valides et les listes de révocation (Certificate Revocation List : CRL)

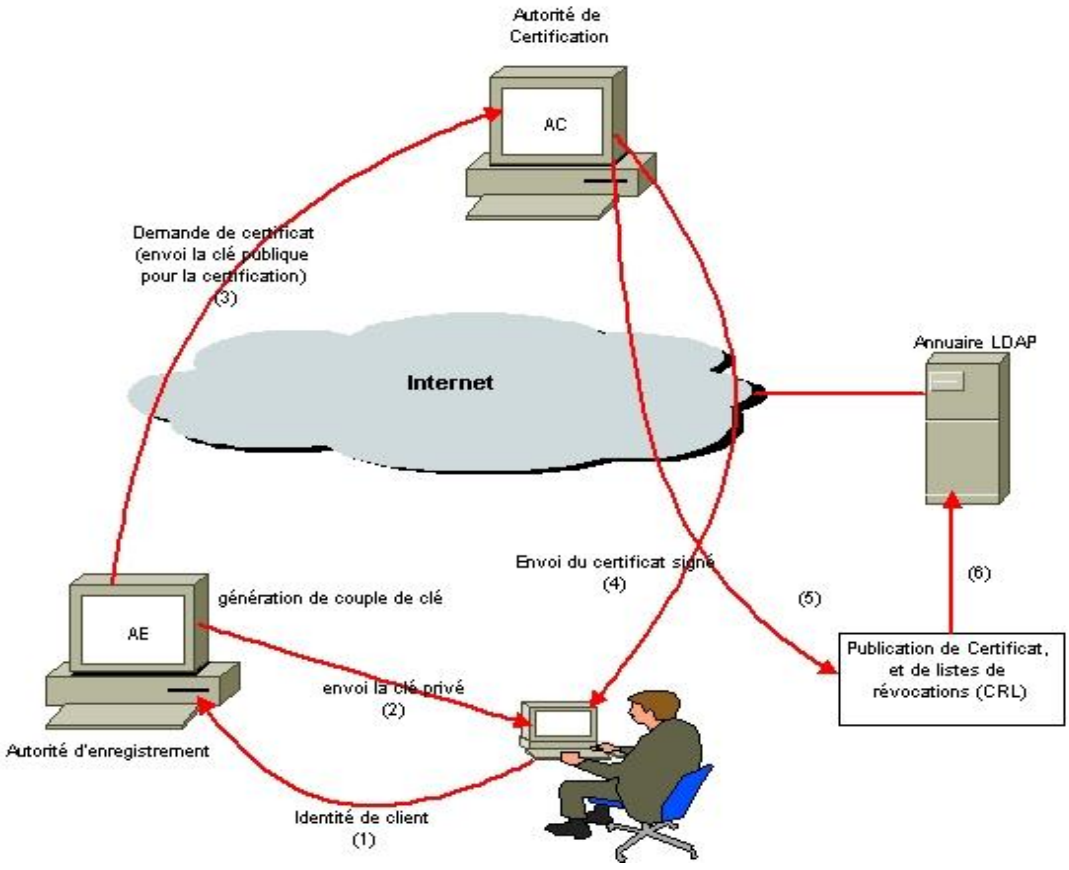


Figure 4.11 – Processus de certification

Le format standard d'un certificat est défini selon la norme X.509 :

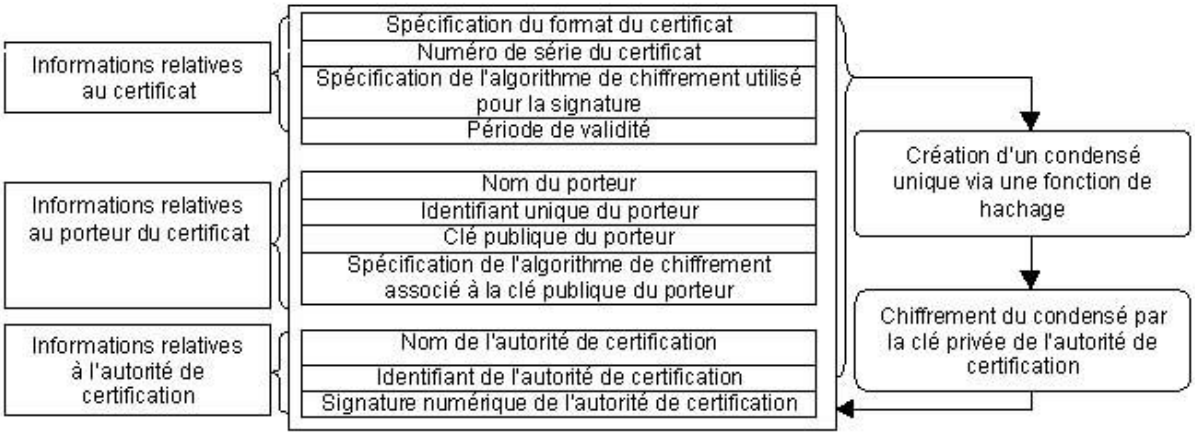


Figure 4.12 – Format d'un certificat X.509

L'authentification par certificats X.509 est utilisée sur support physique (carte à puce ou Token USB) ou support logiciel. C'est une authentification forte qui fait intervenir une signature d'un challenge avec la clé privée de l'utilisateur. La clé privée et le certificat de l'utilisateur sont stockés sur le support.
Le serveur d'authentification vérifie l''identité de l'utilisateur grâce au certificat , et il l'authentifie en vérifiant sa signature avec la clé publique présente dans le certificat.

- **Biométrie**

Les techniques d'identification par la biométrie servent principalement à des applications dans le domaine de la sécurité, comme le contrôle d'accès automatique, un tel dispositif étant qualifié de système de contrôle biométrique.

Un système de contrôle biométrique est donc un système automatique de mesure basé sur la reconnaissance de caractéristiques propres à une personne.

Figure 4.13 – Quelques exemples d'authentification forte

L'identification par la biométrie peut se pratiquer avec les empreintes digitales, l'iris, les réseaux veineux de la rétine, les réseaux veineux de la paume de la main, la morphologie de la main, le poids, ainsi qu'avec les traits du visage. Le moyen biométrique le plus utilisé actuellement est la lecture d'empreintes digitales (environ 50% des applications).

Principe

Exemple : Empreintes digitales

Un système de contrôle biométrique par reconnaissance d'empreintes digitales procède par comparaison entre les empreintes des personnes autorisées (stockées dans une base de données) et l'empreinte de l'individu se présentant au contrôle.

Figure 4.14 – Authentification par empreintes digitales

Cela suppose deux étapes dans la mise en œuvre :

Etape n°1 : l'enregistrement des empreintes des personnes autorisées
Capture d'image → Extraction des minuties et création d'un gabarit → Enregistrement dans la base de gabarits

Étape n°2 : l'**authentification** qui compare une empreinte lue sur le capteur avec une empreinte déjà Enregistrée
Capture d'image → Extraction des minuties → Comparaison avec la base de gabarits → Autorisation d'accès ou non

CHAPITRE 5

Pare-feu (Firewall)

1 Introduction

Chaque ordinateur connecté à Internet (ou sur n'importe quel réseau) est susceptible d'être victime d'intrusion pouvant compromettre l'intégrité du système ou des données. Les pirates informatiques ayant l'intention de s'introduire dans les systèmes recherchent, les failles de sécurités dans les protocoles, les systèmes d'exploitations et les applications. Ils scrutent les réseaux dans l'espoir de trouver un ordinateur mal protégé dans le quel, ils pourront s'introduire pour voler les données ou déposer des virus. Cette menace est encore plus importante si l'ordinateur est connecté en permanence à Internet.
Il est donc nécessaire, pour les entreprises, les établissements publics et les particuliers connectées à Internet, de se protéger des intrusions en installant un système pare-feu (firewall).

2 Définition

Un firewall (pare-feu), est un système physique (matériel) ou logique (logiciel) servant d'interface entre un ou plusieurs réseaux afin de contrôler et éventuellement bloquer la circulation des paquets de données, en analysant les informations contenues dans les couches 3, 4 et 7 du modèle OSI. Il s'agit donc d'une machine (machine spécifique dans le cas d'un firewall matériel ou d'un ordinateur sécurisé hébergeant une application particulière de pare-feu) comportant au minimum deux interfaces réseau :

- une interface pour le réseau à protéger (réseau interne)
- une interface pour le réseau externe

Les firewalls visent, au niveau de la sécurité, deux objectifs :

1) Contrôler et protéger les machines du réseau local :

- Contre la divulgation non autorisée d'informations sensibles.
- Contre les virus de toutes sortes.

- Contre les attaques par cheval de Troie.

2) Protéger les serveurs Internet :

- contre des commandes jugées dangereuses associées à des services du type Telnet et Sendmail.
- contre la modification ou la suppression non autorisée de fichiers vitaux pour le système.

Donc l'installation d'un firewall repose presque toujours sur la nécessité de protéger un réseau de l'intrusion. Dans la plupart des cas, il devra interdire l'accès aux ressources matérielles et logicielles aux utilisateurs non autorisés

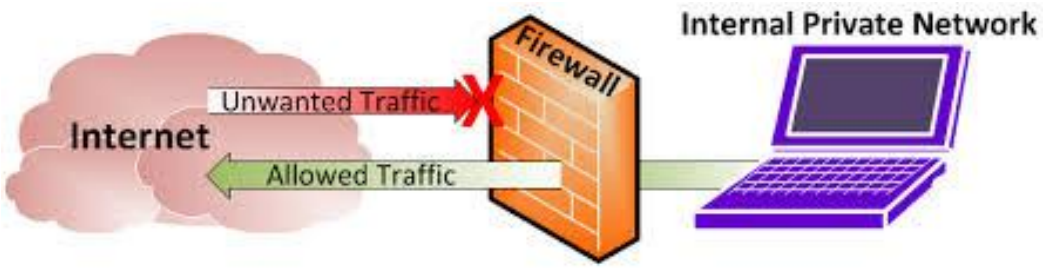


Figure 5.1 – Principe de fonctionnement d'un firewall

3 Fonctionnement d'un firewall

3.1 Principe de fonctionnement

Un système pare-feu contient un ensemble de règles prédéfinies permettant :

- D'autoriser la connexion (**allow**)
- De bloquer la connexion (**deny**)
- De rejeter la demande de connexion sans avertir l'émetteur (**drop**).

L'ensemble de ces règles permet de mettre en oeuvre une méthode de filtrage dépendant de la politique de sécurité adoptée par l'entité. On distingue habituellement deux types de politiques de sécurité permettant :

- soit d'autoriser uniquement les communications ayant été explicitement autorisées : « tout ce qui n'est pas explicitement autorisé est interdit » , il s'agit d'une stratégie dangereuse car il faut modifier la configuration du firewall à chaque fois qu'un nouveau service est crée .

- soit d'empêcher les échanges qui ont été explicitement interdits :

« tout ce qui n'est pas explicitement interdit est autorisé » , cette stratégie est la plus souvent choisie, mais impose toutefois une définition précise et contraignante des besoins en communication.

3.2 Types de filtrage

La plupart des firewalls travaillent au niveau des couches 4 (TCP, UDP...), 3 (IP...) et 2 (Ethernet...). Ils opèrent par filtrage de paquets et ne comprennent rien aux protocoles au dessus (ils sont incapables de filtrer HTTP, SMTP, POP3...).

Certains firewalls sont capables de travailler au niveau de la couche 7 (filtrage applicatif). Ils sont généralement plus lents, plus lourds et plus complexes à configurer mais permettent de filtrer certains protocoles comme HTTP, SMTP, POP3, FTP…

Par exemple, c'est utile pour bloquer le téléchargement de virus, interdire certains sites, filtrer les cookies...

- **Le filtrage de paquets :** dans ce cas, la fonction du firewall est basé sur le principe du filtrage de paquets IP, c'est-à-dire sur l'analyse des en-têtes des paquets IP échangés entre deux machines. Lorsqu'une machine de l'extérieur se connecte à une machine du réseau local, et vice-versa, les paquets de données passant par le firewall contiennent les en-têtes suivants, qui sont analysés par le firewall:

  - L'adresse IP de la machine émettrice
  - L'adresse IP de la machine réceptrice
  - Le type de paquet (TCP, UDP, ...)
  - Le numéro de port (un port est un numéro associé à un service ou une application réseau)

  Les adresses IP contenues dans les paquets permettent d'identifier la machine émettrice et la machine cible, tandis que le type de paquet et le numéro de port donnent une indication sur le type de service utilisé.

  Lorsque le filtrage est basé sur les adresses IP on parle de filtrage par adresse, tandis que le terme de filtrage par protocole est utilisé lorsque le type de paquets et le port sont analysés.
  Certains ports sont associés à des services courants (les ports 25 et 110 sont généralement associés au courrier électronique, et le port 80 au Web) et ne sont généralement pas bloqués. Toutefois, il est recommandé de bloquer tous les ports qui ne sont pas indispensables (selon la politique de sécurité retenue).
  Le port 23 par exemple est critique car il correspond au service Telnet qui permet d'émuler un accès par terminal à une machine du réseau de manière à pouvoir exécuter des commandes saisies au clavier à distance...

  En résumé, le filtrage de paquets présente des avantages en terme de :
  - gain de temps pour la mise en place
  - coût généralement faible
  - performances assez bonnes
  - transparence aux utilisateurs et aux applications.

  Cependant, le besoin croissant de sécurité et de contrôle du contenu informationnel des échanges met en évidence les limites de ce type de firewall :
  - L'élaboration de règles de filtrage peut être une opération pénible à partir du moment où l'administrateur réseau doit avoir une compréhension détaillée des différents services Internet et du format des en-têtes des paquets. Si des règles complexes de filtrage doivent être mises en place, c'est un processus long, lourd et difficile à faire évoluer et à comprendre. De plus, une mauvaise configuration peut conduire le site à rester vulnérable à certaines attaques.
  - Le firewall filtre de paquets ne protège pas contre les applications du type Cheval de Troie car il n'analyse pas le contenu des paquets. Il est donc possible de tenter une attaque par tunneling de protocole, c'est à dire passer par les protocoles autorisés pour en atteindre d'autres interdits.
  - Plus le nombre de règles à appliquer est grand, plus les performances du firewall diminuent, diminuant d'autant les performances de tout le système. On doit alors faire ici un choix parfois crucial entre performance et sécurité.

Le tableau ci-dessous montre un exemple de configuration des règles d'un firewall **:**

Règle	Action	IP source	IP destination	Protocol	Port source	Port destination
1	Accept	176.64.1.33	164.47.28.6	TCP	Any	25
2	Accept	Any	176.64.1.33	TCP	Any	80
3	Accept	192.168.20.0/50	Any	TCP	Any	80
4	Deny	Any	Any	Any	Any	Any

- **Le filtrage applicatif :** Le filtrage applicatif permet, comme son nom l'indique, de filtrer les communications application par application, ce qui signifie qu'il travaille au niveau de la couche 7 du modèle OSI. Le filtrage applicatif suppose donc une connaissance de l'application, et notamment de la manière de laquelle elle structure les données échangées.
  Un firewall effectuant un filtrage applicatif est appelé **passerelle applicative** car il permet de relayer des informations entre deux réseaux en effectuant un filtrage fin au niveau du contenu des paquets échangés. Il s'agit donc d'un dispositif performant assurant une bonne protection du réseau, pour peu qu'il soit correctement administré. En contrepartie une analyse fine des données applicatives requiert une grande puissance de calcul et se traduit donc souvent par un ralentissement des communications.

  Avantages du filtrage applicatifs:
  Les avantages des Firewalls filtrant au niveau application sont les suivants :
  - Les filtres au niveau application donnent à l'administrateur un contrôle complet sur chaque service par la restriction sur les commandes utilisables et sur l'accès aux hôtes internes par ces services.
  - Les filtres au niveau application permettent l'installation de procédures d'authentification extrêmement poussées.
  - Les règles d'un firewall filtrant au niveau application sont bien plus faciles à configurer et à tester que pour un firewall filtre de paquets.

  Limitations des filtres au niveau application:
  En contrepartie, les limitations des filtres au niveau application sont :
  - Les filtres au niveau application augmentent considérablement le coût du firewall, ce coût étant principalement lié à celui de la plate-forme matérielle de la passerelle, des services proxy et du temps et des connaissances requises pour configurer cette passerelle.
  - Les filtres au niveau application ont tendance à réduire la qualité du service offert aux utilisateurs tout en diminuant la transparence du système.
  - Une analyse fine des données applicatives requiert une grande puissance de calcul et se traduit donc souvent par un ralentissement des communications.

- **Le filtrage dynamique :** Quand une application TCP (orientée connexion) crée une session avec une machine d'un réseau distant, un port est également loué au niveau du système expéditeur pour recevoir le trafic réseau envoyé par le système destinataire. Suivant le protocole TCP, ce port source au niveau du client sera d'un numéro supérieur à 1023 et inférieur à 16384, tandis que

le port de l'hôte destinataire sera fort probablement d'un numéro inférieur à 1023 (25 pour le service SMTP, 80 pour HTTP, etc.).

Une fois installés, les Firewalls classiques doivent donc permettre à tous le trafic réseau arrivant par un port d'un numéro supérieur à 1023 de passer, afin de permettre à de tels applications de s'exécuter correctement, cependant l'ouverture de ce nombre important de port augmente considérablement le risque d'intrusion dans le réseau privée.

Le filtrage dynamique (stateful inspection Firewall) permet de résoudre le problème en utilisant une table d'état (state table) contenant la liste de toutes les connections établies ainsi que les numéros de port correspondants, cette table est par la suite employée pour valider l'utilisation des règles de filtrage au moment de l'analyse des paquets du trafic entrant au réseau privé. Bien que les « stateful inspection Firewalls » ont donné un niveau de sécurité plus élevé que les Firewalls de base classique, en considérant d'autres aspects de la couche transport du modèle OSI, ils ont hérité en même temps plusieurs faiblesses connues dans les types antécédents, notamment ceux liées à la rapidité de traitement et la vulnérabilités aux plusieurs types d'attaques.

4 Types de firewalls

Il existe deux grandes familles de firewalls *:* les firewalls **matériels** qui protègent tout un réseau local et les firewalls **logiciels** qui ne protègent que la machine sur laquelle ils sont installés. Laquelle des deux solutions il faut adopter. La réponse est très simple : opter pour un firewall logiciel pour une machine, mais adopter la deuxième solution pour le réseau .

L'usage de 2 firewalls s'appelle une défense en profondeur. Elle s'est imposée avec l'arrivée des réseaux sans fil. Le firewall matériel constitue un premier bouclier entre Internet et le réseau personnel (qu'il soit sans fil ou non). Tandis que le firewall logiciel se charge de protéger chaque machine d'attaques plus complexes et d'attaques menées depuis l'intérieur.

- **Firewalls logiciels :** Ils sont destinés aux particuliers qui n'ont pas les moyens d'acheter une machine séparée uniquement pour faire un firewall. Au lieu d'être une machine séparée, le firewall personnel est un logiciel qui fonctionne directement sur l'ordinateur à protéger.

  Ils ont quelques inconvenients :

  - Si un hacker parvient à s'introduire sur l'ordinateur, il pourra facilement désactiver le firewall (C'est beaucoup plus difficile quand le firewall est un ordinateur séparé)
  - Comme il fonctionne en même que les autres logiciels, il ralentit légèrement l'ordinateur.

  Mais il a aussi un avantage: il est capable de contrôler quels logiciels vont se connecter sur Internet (ce que ne sait pas faire un firewall séparé).

  Parmi les firewalls "personnels", citons : Privatefirewall, Sygate , ZoneAlarm, Agnitum Outpost, Look'n Stop...

- **Firewalls matériels :** Ils sont sous la forme de boîtiers compacts. En réalité, se sont de simples ordinateurs possédant leur propre système d'exploitation et leur propre logiciel firewall. Mais ils n'ont rien de différent d'un simple firewall installé sur un ordinateur.Les Firewalls de cette catégorie présentent l'avantage de pouvoir protéger de nombreux ordinateurs à la fois et d'être indépendant des systèmes d'exploitation. Mais ils coûtent généralement très cher.

La mise en oeuvre d'un firewall matériel est un peu moins simple que celle d'un firewall logiciel. Mais on trouve à l'heure actuelle des modem ADSL - routeur sans fil - firewall très bon marché dont l'ergonomie a été spécifiquement étudiée pour le grand public (par exemple Netgear DG834G, Linksys Wag54G, DLink DSL 604+, …)

On peut parfaitement imaginer d'avoir les 2 types de firewalls en même temps: un firewall séparé pour protéger le réseau, et un firewall personnel pour protéger chaque ordinateur.

5 Limites des firewalls

Le fait d'installer un firewall n'est bien évidemment pas signe de sécurité absolue.
Les firewalls ne protège en effet que des communications passant à travers eux. Ainsi, les accès au réseau extérieur non réalisés au travers du firewall sont autant de failles de sécurité.
La mise en place d'un firewall doit donc se faire en accord avec une véritable politique de sécurité. D'autre part la mise en place d'un système pare-feu n'exempt pas de se tenir au courant des failles de sécurité et d'essayer de les minimiser...
Ils ne protègent pas très bien des virus car il y a trop de manières différentes de coder des fichiers pour les transférer. En d'autres termes, un firewall ne pourra pas remplacer l'attention et la conscience des utilisateurs qui doivent respecter un certain nombre de règles pour éviter les problèmes... La première étant bien évidemment de ne jamais ouvrir un fichier attaché à un mail sans être sûr de sa provenance. Il faut prendre des mesures globales et importantes contre les virus. Avant de traquer les virus à l'entrée du réseau, il faut s'assurer que chaque poste de travail dispose d'un anti-virus. Les virus passent également très facilement par disquette... Les virus sur Internet son bien moins importants que les virus sur disquette.
Il est impossible de confirmer la source des données qui traversent un garde-barrière, ainsi ce dernier ne peut ni en assurer l'intégrité ni protéger leur source. Il ne fait que transférer les données d'un réseau à l'autre et est incapable de détecter si elles ont été modifiées en transit.
Un firewall est incapable d'assurer la confidentialité des données. Une fois qu'il a permis l'accès à des données, il ne peut en maintenir le contrôle. De cette manière, les messages électroniques ou mots de passes d'authentification en clair peuvent être lus pendant leur traversée du réseau. Enfin, les firewalls ne protègent pas les réseaux des techniques de sniffing.

CHAPITRE 6

Réseaux VPN et Protocoles de sécurité

1 Définition

Un réseau VPN (Virtual Private Network) permet d'établir un lien sécurisé pour accéder depuis n'importe où aux ressources et aux services d'un réseau d'entreprise.

Donc un VPN dispose de la même fonctionnalité qu'un réseau privé (utilisant des lignes spécialisées) mais utilise Internet pour créer des lignes louées virtuelles qui passent par le réseau public.

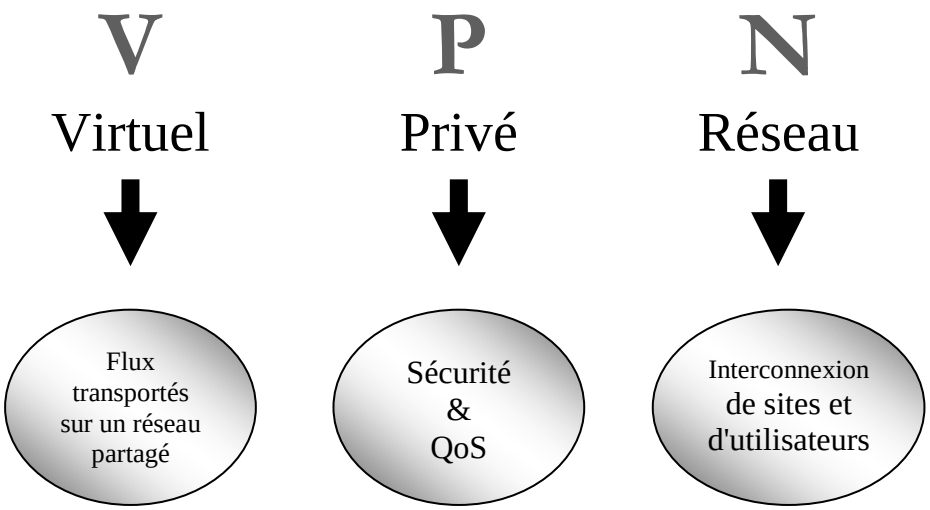


Figure 6.1 – Objectifs d'un VPN

Il est constitué de liaisons virtuelles sur Internet entre des sites distants appartenant à une même société ou à un même organisme. En chiffrant les données, tout se passe exactement comme si la connexion se faisait en dehors d'Internet comme l'illustre la figure suivante :

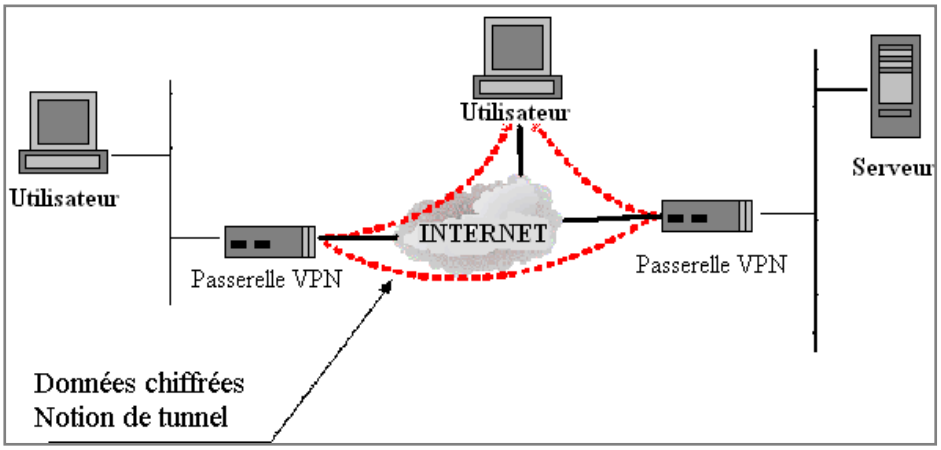


Figure 6.2 – Principe d'un VPN

Ce réseau est dit virtuel car il relie deux réseaux physiques de LAN par une liaison Internet de type WAN, et privé où seuls les ordinateurs des réseaux locaux faisant partie du VPN peuvent accéder aux données. Il permet le raccordement de travailleurs mobiles et l'interconnexion de sites distants.

2 Principe de fonctionnement

Un réseau VPN est réalisé avec des mécanismes de chiffrement et d'authentification. En chiffrant les données, tout se déroule comme si la machine se trouvait directement sur son réseau privé, sans qu'aucune personne extérieure à ce réseau puisse accéder aux données qu'elle envoie et reçoit.

Les utilisateurs du réseau peuvent se connecter en toute confidentialité et partager les applications et les informations de l'entreprise.

On peut réaliser un VPN à l'aide de matériels spécifiques (cartes réseaux, routeurs, . . .), de logiciels ou d'une combinaison des deux (hardware/software).

Tunneling : Le processus d'encapsulation, de routage et de décapsulation est désigné dans son ensemble par le terme tunneling. Le tunneling encapsule le paquet d'origine dans un nouveau paquet. Ce nouveau paquet peut comporter de nouvelles informations d'adressage et de routage, qui lui permettent de parcourir les réseaux WAN. Lorsque le tunneling est lié à une certaine confidentialité, les données du paquet d'origine (ainsi que la source et la destination d'origine) ne sont pas révélées à ceux qui écoutent le trafic sur le réseau. Une fois que les paquets encapsulés ont atteint leur destination, l'en-tête d'encapsulation est retiré et l'en-tête du paquet d'origine est utilisé pour transférer le paquet vers sa destination finale.

Le tunnel à proprement parler est le chemin de données logique que parcourent les paquets encapsulés. Entre la source d'origine et son homologue de destination, le tunnel est généralement transparent et ressemble à n'importe quelle autre connexion point à point du chemin d'accès réseau. Lorsque le tunneling est lié à une certaine confidentialité, il permet de mettre au point des réseaux privés virtuels (VPN).

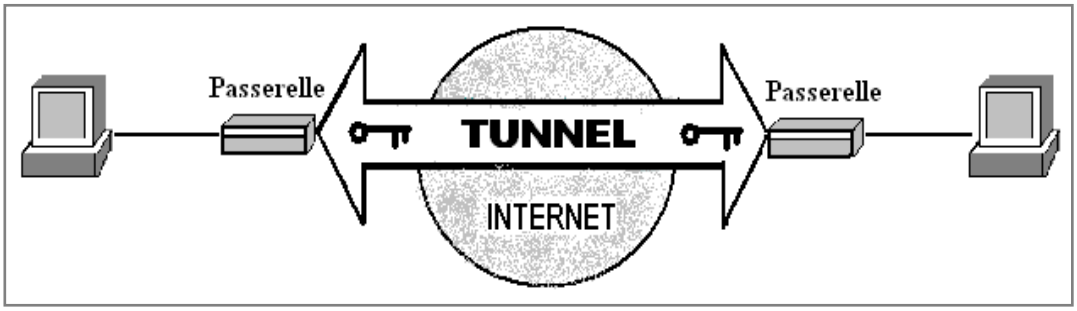


Figure 6.3 – Tunnel VPN

Un réseau VPN est utilisé :

- Pour permettre le lien entre deux ou plusieurs sites appartenant à la m
- ême d'une entreprise. Par exemple entre le siège et les filiales. Dans ce cas, le tunnel est établi en général de manière permanente entre deux équipements qui ont un rôle de passerelle. (Architecture LAN TO LAN)

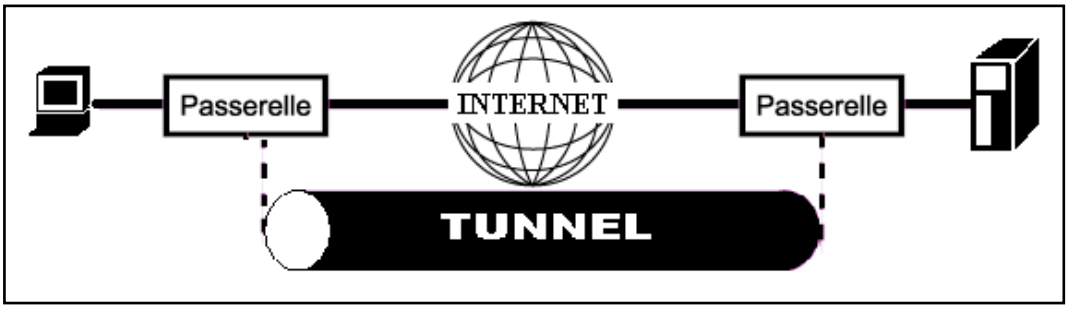


Figure 6.4 –Architecture LAN TO LAN

- Pour permettre le lien entre une entreprise et ses collaborateurs travaillant à distance (Architecture HOST TO LAN).

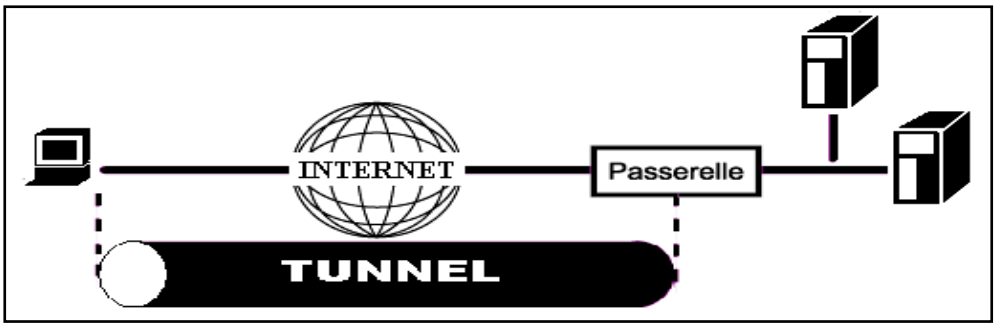


Figure 6.5 –Architecture HOST TO LAN

3 Architectures VPN

Les architectures VPN sont de 3 types :

3.1 VPN Intranet

C'est un réseau VPN entre les différents services d'une entreprise, les bureaux des filiales, les bureaux situés à l'étranger, etc. Sans VPN, les entreprises seraient forcées d'utiliser des lignes dédiées entre leurs filiales (procédé très onéreux, surtout lorsqu'il s'agit de lignes internationales). Avec les réseaux VPN, ces mêmes communications peuvent passer par l'Internet sans souci de confidentialité ou d'intégrité des transferts, et c'est pour un coût bien moindre.

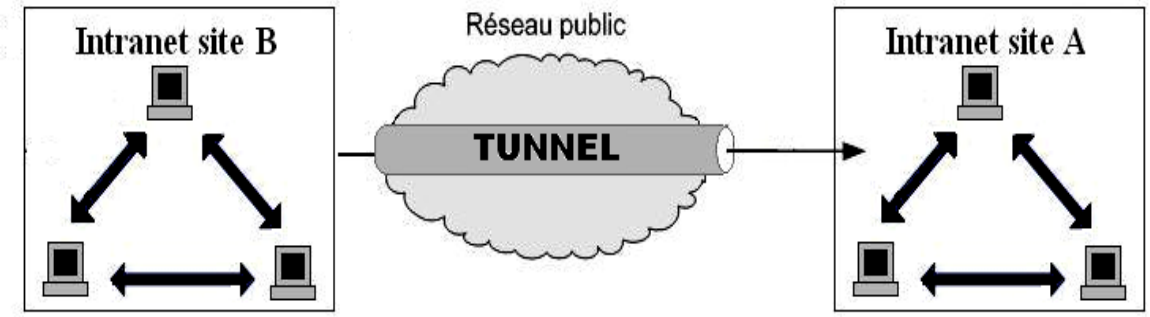


Figure 6.6 –VPN INTRANET

3.2 VPN Extranet

C'est l'extension du VPN Intranet et qui permet de connecter les utilisateurs ne faisant pas partie de l'entreprise (partenaires, fournisseurs, clients ou autres interlocuteurs) aux bureaux de l'entreprise, pour accéder à certaines données. Presque tous les sites « e-commerce » ainsi que les banques offrent ce type de connexion sécurisée à leurs clients.

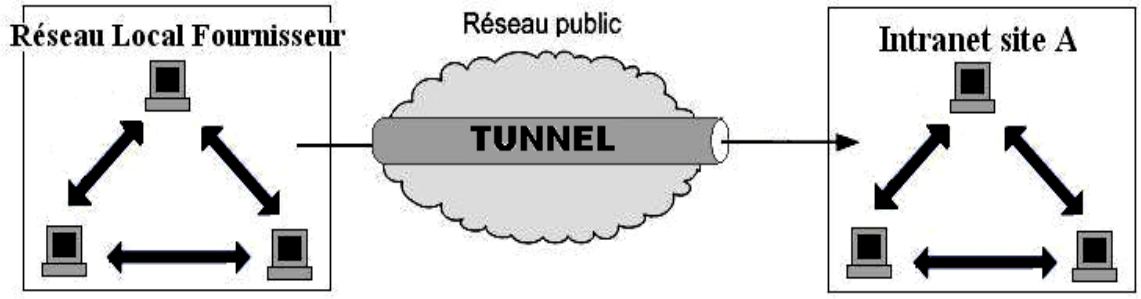


Figure 6.7 – VPN EXTRANET

3.3 VPN pour l'accès à distance

C'est l'extension du VPN Intranet, qui permet de connecter les utilisateurs Nomades aux bureaux de l'entreprise. Ce type de réseaux VPN peut être utilisé pour accéder à certaines ressources prédéfinies d'une entreprise sans y être physiquement présent. Cette opportunité peut ainsi être très utile au commercial ou au cadre qui souhaite se connecter au réseau de son entreprise lors d'un déplacement. En général, l'utilisateur de ce type de VPN possède un accès Internet chez un fournisseur d'accès standard (ISP).

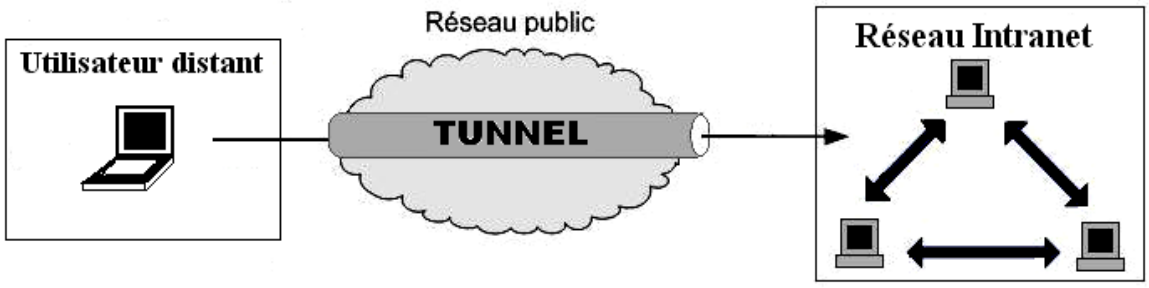
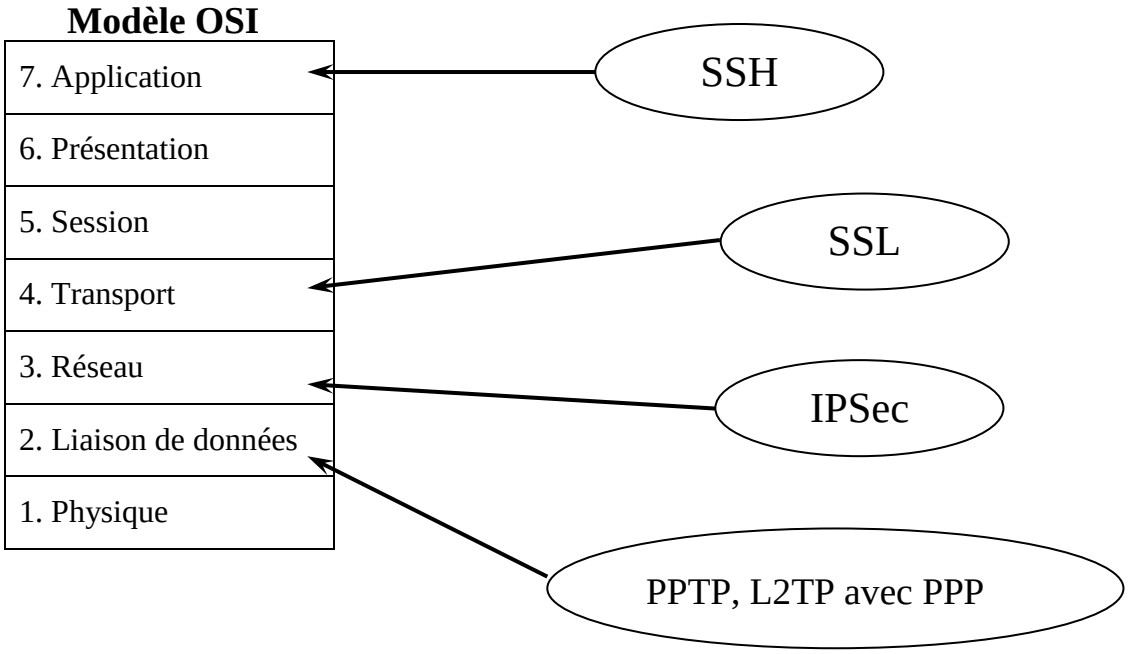


Figure 6.8 –VPN POUR l'ACCES A DISTANCE

4 Protocoles sécurises pour VPN

Nous pouvons classer les protocoles que nous allons étudier en trois catégories:

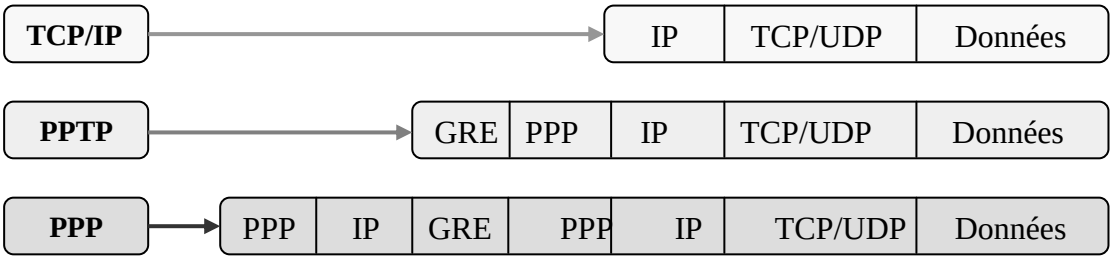
- Les protocoles de niveau 2 comme PPTP et L2TP.
- Les protocoles de niveau 3 comme IPSec.
- Le protocole de niveau 4 : SSL.
- Le protocole de niveau 7 : SSH.



4.1 Protocole PPTP

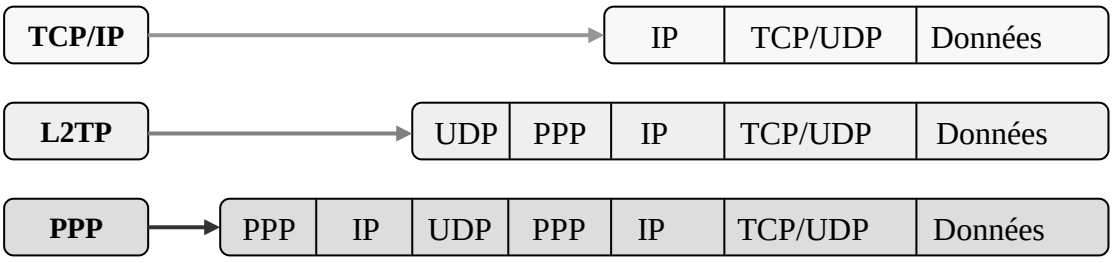
Le principe du protocole PPTP (Point To Point Tunneling Protocol) est de créer des trames sous le protocole PPP et de les encapsuler dans un datagramme IP.

Ainsi, dans ce mode de connexion, les machines distantes des deux réseaux locaux sont connectés par une connexion point à point (comprenant un système de chiffrement et d'authentification, et le paquet transite au sein d'un datagramme IP.



4.2 Protocole L2TP

Le protocole **L2TP** (Layer Two Tunneling Protocol) est un protocole standard de tunnelisation très proche de PPTP. Il permet l'encapsulation des paquets PPP au niveau des couches 2 (Frame Relay et ATM) et 3 (IP). Lorsqu'il est configuré pour transporter les données sur IP, L2TP peut être utilisé pour faire du tunnelling sur Internet. L2TP repose sur deux concepts : les concentrateurs d'accès L2TP (LAC : L2TP Access Concentrator) et les serveurs réseau L2TP (LNS : L2TP Network Server). L2TP n'intègre pas directement de protocole pour le chiffrement des données. C'est pourquoi L'IETF préconise l'utilisation conjointe d'IPSec et L2TP.



4.3 Protocole IPSec

IP Security IPSec (Internet Protocol Security) a été développé par l'IETF pour fournir des services cryptographiques de sécurité de la couche Réseau (niveau 3) :

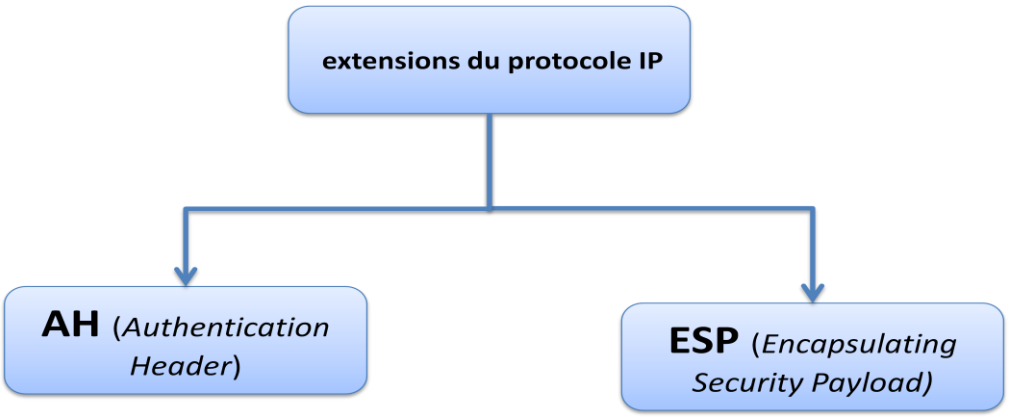
Les services de sécurité offerts par IPsec :

- Confidentialité des données:
- Authentification des extrémités :

- Authenticité des données et contrôle d'accès continu:
- Intégrité des données échangées en mode non connecté :
- Protection contre les écoutes et analyses de trafic :
- Protection contre le rejeu :

Ces services sont basés sur des **mécanismes cryptographiques** qui leur confèrent un niveau de sécurité élevé lorsqu'ils sont utilisés avec des algorithmes forts.

IPSec étend le protocole IP natif avec les services de sécurité nécessaires à la création d'un réseau VPN. Ces extensions se composent de deux nouveaux en-têtes ajoutés (soit séparément, soit ensemble) au paquet IP



L'en-tête AH est utilisé pour fournir l'authentification de l'origine, l'intégrité des données et facultativement la protection contre la ré-exécution.

L'en-tête ESP est utilisée pour fournir la confidentialité des données et facultativement tous les services de l'en-tête AH L'en-tête ESP est généralement utilisé en "mode tunnel", mode dans lequel la totalité du data gramme IP est encapsulée (et chiffrée), ce qui permet d'obtenir une certaine forme de protection de la confidentialité du flux de données.

En complément, le groupe de travail IETF a défini le protocole IKE Internet Key Exchange comme le protocole clé standard associé à IPSec. Il fournit les mécanismes nécessaires pour négocier entre les entités les paramètres de sécurité et les éléments de chiffrement qui seront utilisés avec les en-têtes AH et ESP.

Un système conforme à IPSec peut :

- Choisir des protocoles de sécurité
- Déterminer les algorithmes à utiliser pour les services
- Utiliser les clés cryptographiques et les certificats
-

4.3.1 Modes de fonctionnement

Afin d'assurer l'intégrité, l'authentification et la confidentialité des données transmises, ainsi qu'optionnellement la protection contre le rejeu, IPSec utilise deux protocoles distincts : AH (Authentification Header) et ESP (Encapsulating Security Payload).

Ces protocoles peuvent être appliqués dans deux modes différentes, selon le type de sécurisation et de connexion utilisés (mode transport, mode tunnel).

Le mode transport :

Il s'intercale entre le protocole réseau (IP) et le protocole de transport, seules les données des protocoles de niveau supérieur (TCP, UDP…) transportées par le datagramme IP, sont protégées sans toucher à l'en-tête. Il est réservé aux connexions entre équipements terminaux (postes clients, serveurs). Il peut être utile dans le cas d'une communication sécurisée au sein d'un LAN, par exemple.

L'inconvénient de ce mode réside dans le fait que l'en-tête extérieur est produit par la couche IP c'est-à-dire sans masquage d'adresse. De plus, le fait de terminer les traitements par la couche IP ne permet pas de garantir la non-utilisation des options IP potentiellement dangereuses. L'intérêt de ce mode réside dans une relative facilitée de mise en oeuvre.

Le mode tunnel :

Il permet la création de tunnels par encapsulation de chaque paquet IP dans un nouveau paquet. Ainsi, la protection porte sur tous les champs des paquets IP arrivant à l'entrée d'un tunnel, y compris sur les champs des en-têtes (adresses source et destination par exemple). Donc l'en-tête IP est également protégé et remplacé par un nouvel en-tête qui permet juste au paquet de traverser le tunnel de sécurité, à la sortie duquel l'en-tête original est rétabli. Ce mode offre une protection plus importante contre l'analyse du trafic, car il permet le masquage des adresses de l'expéditeur et du destinataire final.

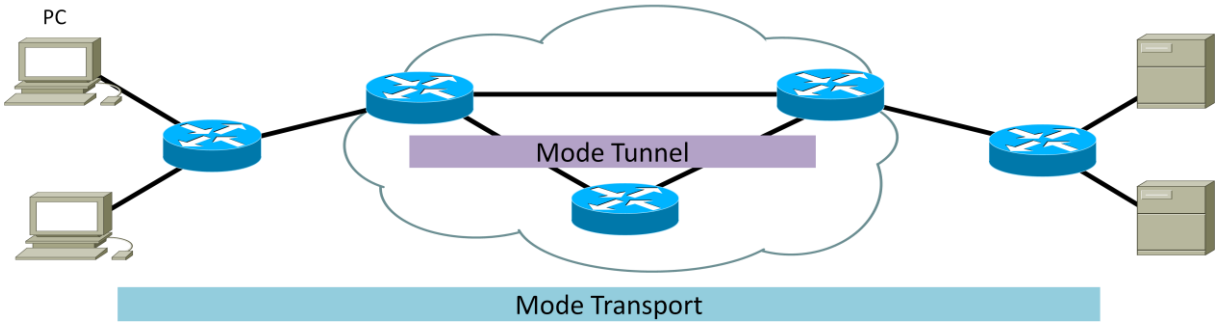


Figure 6.9 –Modes de fonctionnement d'un VPN :
Mode Tunnel et Mode Transport

4.3.2 Protocoles AH et ESP

Donc comme on a déjà dit, le protocole IPsec utilise deux sous protocoles pour la protection des datagrammes : AH et ESP.

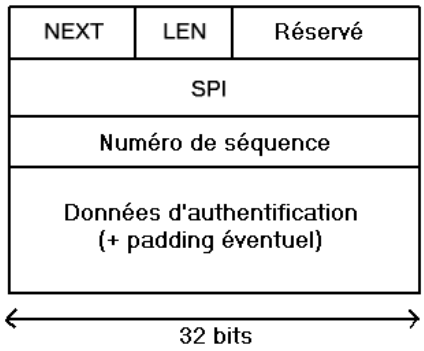
AH permet l'authentification, le contrôle d'intégrité et l'anti-rejeu, ESP offre les mêmes fonctionnalités plus la confidentialité.

a) Le protocole d'authentification AH (Authentication Header)

Il est conçu pour assurer l'intégrité et l'authentification des datagrammes IP sans chiffrement des données (i.e. sans confidentialité).

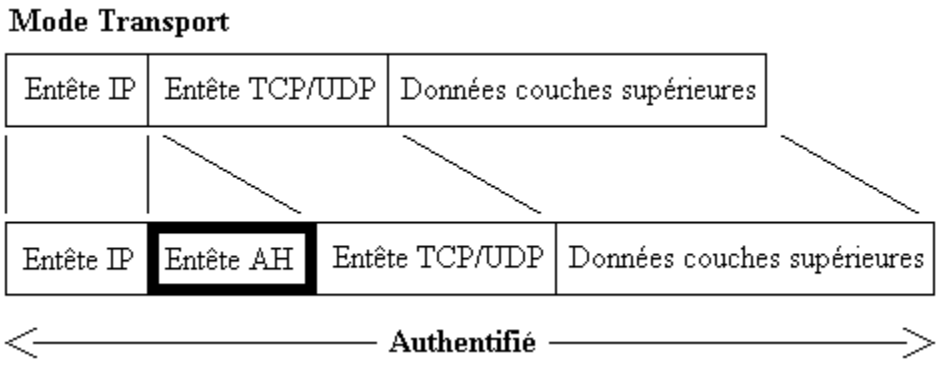
Le principe de AH est d'adjoindre au datagramme IP classique un champ supplémentaire permettant à la réception de vérifier l'authenticité des données incluses dans le datagramme. Ce bloc de données est appelé "valeur de vérification d'intégrité" (Intégrity Check Value, ICV). La protection contre le rejeu se fait grâce à un numéro de séquence.

L'en-tête AH se compose de 6 champs comme le décrit la figure suivante :

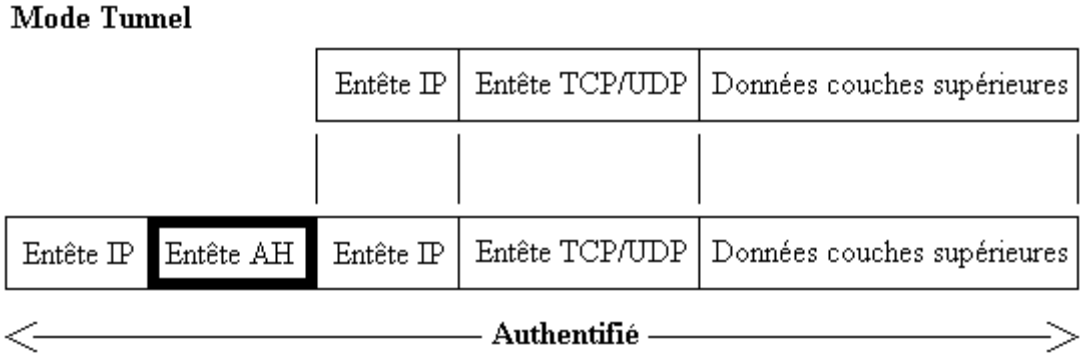


Les transformations AH peuvent se faire en mode transport ou en mode tunnel.

- En mode transport, l'entête original du paquet IP sera celle du paquet sécurisé :



- En mode tunnel, un nouvel entête IP est créé pour chaque paquet sécurisé. C'est ce mode qui est typiquement utilisé dans les logiciels ou boîtiers VPN pour connecter deux réseaux par exemple.



AH permet de se protéger contre les attaques de type :

- spoofing et autres modifications des paquets IP
- rejeu
- DoS quand ils sont basés sur la charge impliquée par les calculs cryptographiques (la vérification d'intégrité n'intervient qu'après)

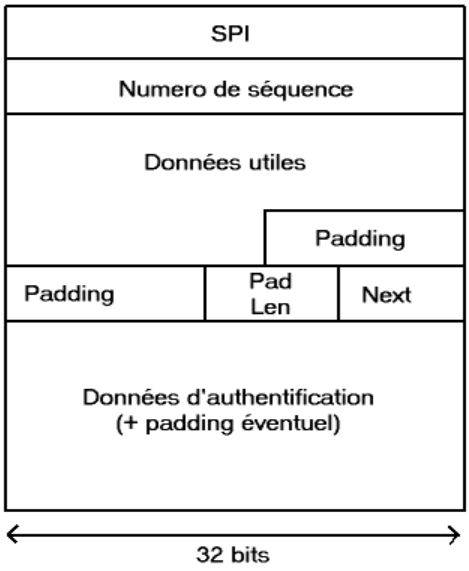
b) Le protocole de confidentialité ESP (Encapsulating Security Payload)

Il fournit les services de sécurité suivants :

- Confidentialité
- Protection contre de l'analyse de trafic
- Intégrité en mode non-connecté (comme AH)
- Authentification des données (comme AH)
- Anti-rejeu (comme AH)

On peut voir tout de suite qu'ESP couvre les services offerts par AH, et on peut se demander pourquoi AH est utilisé et pourquoi l'on complique ainsi un protocole qui l'est déjà bien assez. C'est en effet une question qui est d'actualité mais néanmoins il faut souligner le fait qu'AH offre des services plus complets qu'ESP car il est le seul à protéger l'en-tête du paquet (en-tête original en mode Transport, en-tête IPSec en mode Tunnel). ESP ne protège que les données (c'est-à-dire tout au plus l'en-tête original en mode Tunnel).

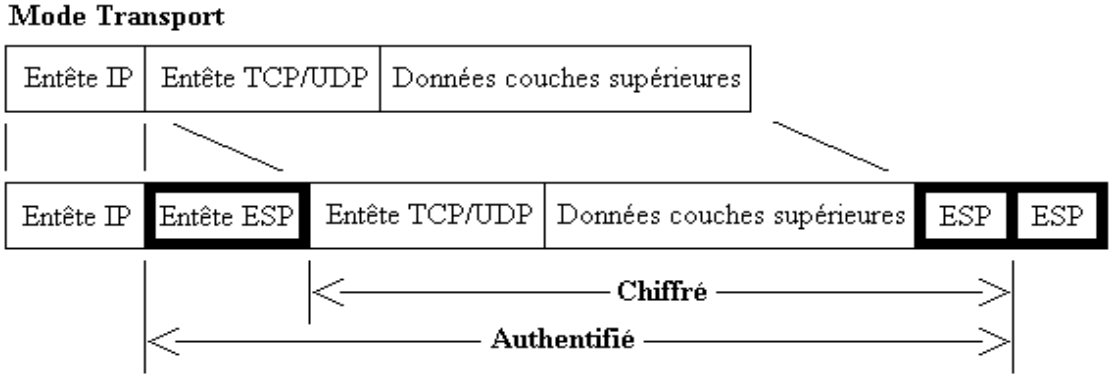
L'en-tête ESP se compose de 7 champs (dont 2 optionnels) comme le décrit la figure suivante:



Nous ne rentrerons pas dans les détails de chaque champ ici. Mais très brièvement, le chiffrement sera appliqué aux données utiles jusqu'au champ NEXT inclus (ce champ contient un identifiant du protocole supérieur, au niveau 4). Comme les données utiles ne sont pas prédéfinies, leur longueur peut varier grandement et un padding assez important est requis. Il est obligatoire et sa longueur est explicitée dans le champ PAD LEN.

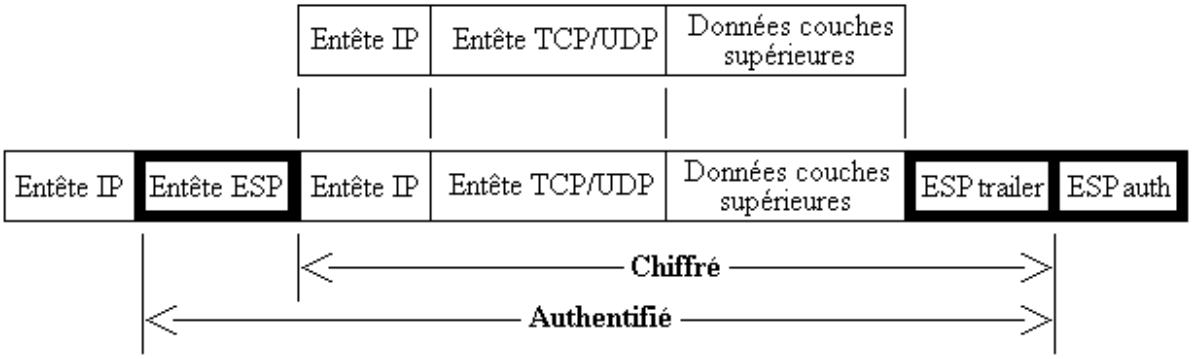
Les données d'authentification protègent les données du champ SPI au champ NEXT inclus; en cas d'utilisation des 2 mécanismes simultanément, le chiffrement est effectué en premier suivi du calcul des données d'intégrité. Cela a pour résultat d'éviter des attaques par déni de service au déchiffrement, ainsi que de permettre d'effectuer les 2 opérations en parallèle (à la réception).

- Tout comme AH, l'ESP peut se décliner en mode transport. L'entête IP du datagramme original est conservé et les autres champs sont sécurisés et précédés d'une entête ESP :



- En mode tunnel, une nouvelle entête IP est générée, précédent le datagramme sécurisé, et ne contenant aucune option IP, même si l'entête initiale en contenait. Ce mode est typiquement utilisé pour les communications d'hôte à hôte (gateway-to-gateway), où il permet de masquer les adresses IP des expéditeurs et destinataires originaux :

Mode Tunnel



ESP permet de protéger contre les attaques de type :

- espionnage et autres divulgations d'informations
- rejeu (optionnel)
- analyse de trafic (optionnel)

4.3.3 Security Association (SA)

Une association de sécurité IPsec (SA) est une connexion qui fournit des services de sécurité au trafic qu'elle transporte. Il s'agit d'une structure de données qui regroupe l'ensemble des paramètres de sécurité associés à une communication donnée.

Les services de sécurité sont fournis par l'utilisation soit de AH soit de ESP. Le rôle d'une SA est donc de consigner, pour chaque adresse IP avec laquelle l'implémentation IPsec peut communiquer, les informations suivantes :

- index de la SA appelé SPI (Security Parameter Index) choisi par le récepteur
- un numéro de séquence, indicateur utilisé pour le service d'anti-rejeu
- une fenêtre d'anti-rejeu : compteur 32 bits
- dépassement de séquence
- paramètres d'authentification (algorithmes et clés)
- paramètres de chiffrement (idem)
- temps de vie de la SA
- mode du protocole IPsec (tunnel ou transport)

Chaque association est identifiée de manière unique à l'aide d'un triplet composé de :

- l'adresse de destination des paquets
- l'identifiant du protocole de sécurité (AH ou ESP)
- le SPI (lors de la négociation d'une SA, un nombre de 32 bits appelé Security Parameters Index (SPI) est attribué pour la désigner).

- **SAD :**

Pour gérer les associations de sécurité actives, on utilise une base de données des associations de sécurité SAD (Security Association Database). Elle contient tous les paramètres relatifs à chaque SA et sera consultée pour savoir comment traiter chaque paquet reçu ou à émettre.

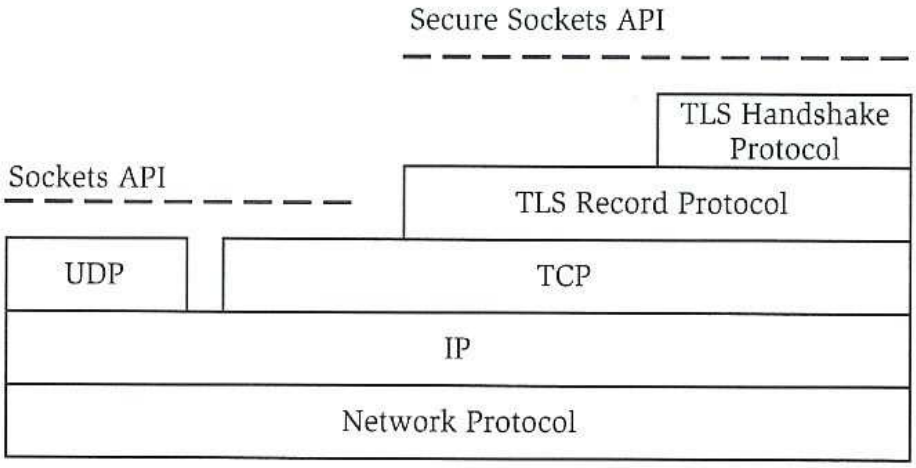
- **SPD :**

Les protections offertes par IPsec sont basées sur des choix définis dans une base de données de politique de sécurité SPD (Security Policy Databas). Cette base de données est établie et maintenue par un utilisateur, un administrateur ou une application mise en place par ceux-ci. Elle permet de décider, pour chaque paquet, s'il se verra apporter des services de sécurité, s'il sera autorisé à passer ou s'il sera rejeté.

4.4 Protocole SSL

SSL/TLS est un protocole qui prend place entre le protocole de la couche de transport et la couche application.

Le protocole SSL supporte l'utilisation d'un grand nombre d'algorithmes de cryptographie, pour des opérations telles que l'authentification mutuelle d'un serveur et d'un client, la transmission de certificat, et l'établissement d'une clef de session. Les clients et les serveurs peuvent supporter différentes suites de chiffrement, c'est à dire différents ensembles d'algorithmes, selon la version de SSL qu'ils intègrent, le protocole de négociation SSL détermine comment serveur et client choisissent l'algorithme de chiffrement utilisé pour s'authentifier l'un à l'autre, pour transmettre des certificats et pour établir les clefs de session.



Les données qui vont et viennent entre le client et le serveur sont chiffrées à l'aide d'un algorithme symétrique tel que DES ou RC4. Un algorithme de clé publique généralement RSA est utilisé pour l'échange des clés de chiffrement et pour les signatures numériques. L'algorithme utilise la clé publique du certificat numérique du serveur. Ce dernier permet également au client de vérifier l'identité du serveur. Les versions 1 et 2 du protocole SSL offrent uniquement l'authentification serveur. La version 3 inclut l'authentification client, utilisant les certificats numériques du client et du serveur.

4.4.1 Principe de fonctionnement de SSL (Secure Socket Layer) :

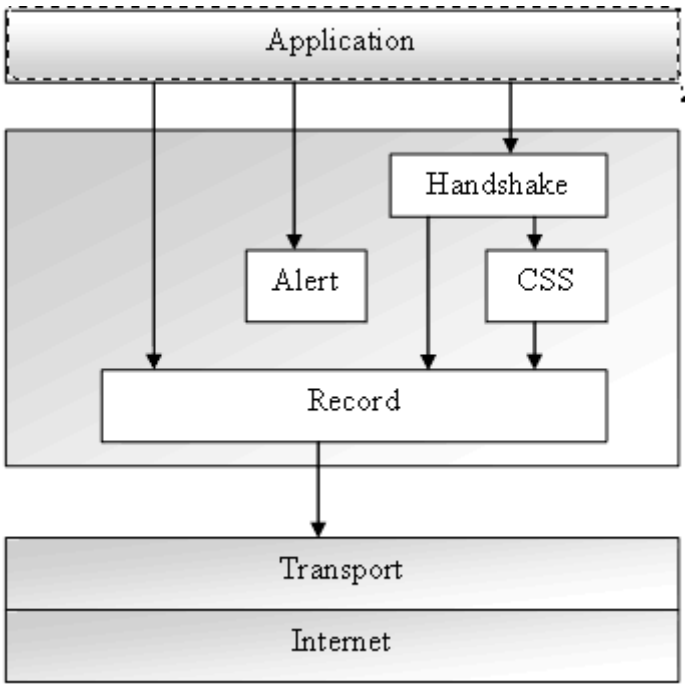
Une session SSL commence lorsqu'une URL spéciale commençant par https est demandée, le client se connecte alors sur le port 443. Cette opération s'effectue de la façon suivante :

L'authentification du serveur SSL permet au client de confirmer l'identité du serveur : le logiciel côté client utilise la technique standard de cryptographie à clé publique pour vérifier que le certificat du serveur et son identifiant public, ont été produits par un organisme certificateur agréé. Cette confirmation est importante pour le client si par exemple, il envoie des informations de sa carte bancaire à travers le réseau et veut s'assurer de l'identité du serveur récepteur.

L'authentification du client SSL permet au serveur de confirmer l'identité du client : c'est le même procédé que celui utilisé pour l'authentification du serveur. Cela peut être important dans le cas où le serveur est une banque, qui veut transmettre une information financière à son client et donc vérifier l'identité du destinataire.

Une connexion cryptée SSL implique que toutes les données échangées entre un client et un serveur soient cryptées par le logiciel expéditeur et décryptées par le logiciel destinataire, ce qui introduit un degré élevé de sécurité. Toutes les données envoyées au moyen d'une connexion cryptée SSL sont protégées par un mécanisme contre les manipulations, qui garantit que les données n'ont pas été modifiées dans leur transit. Le protocole SSL/TLS se décompose en quatre modules :

- Le protocole SSL handshake
- Le protocole SSL Change Cipher Spec
- Le protocole SSL Alert
- Le protocole SSL Record



Le protocole SSL inclut deux sous protocoles :

- Le protocole SSL handshake : choisit la version de SSL et TLS qui sera utilisée, réalise l'authentification par l'échange de certificats et permet la négociation entre le client et le serveur d'un niveau de sécurité au travers du choix des algorithmes de cryptage. C'est le protocole de configuration de la transaction.

- Le protocole SSL record : encapsule et fragmente les données. C'est le protocole de transmission des données.

Le protocole SSL record définit le format utilisé pour transmettre les données.

Le protocole SSL handshake implique l'utilisation du protocole SSL record pour échanger une série de messages entre le serveur et le client lorsqu'ils établissent leur première connexion.

Cet échange de message sert à mettre en place les actions suivantes :

- Authentification du serveur par le client
- Permettre au client et au serveur de choisir les algorithmes de chiffrements qu'ils supportent conjointement
- Authentification du client par le serveur (optionnel)
- Utiliser les techniques de cryptage à clef publiques pour générer des secrets partagés
- Etablir une connexion SSL cryptée

Le chiffrement avec SSL

Le protocole SSL supporte l'utilisation d'un grand nombre d'algorithmes de cryptage différents, notamment pour les opérations d'authentification, de transmission de certificats et d'établissement de clés de session.

Le client et le serveur peuvent supporter différents niveaux de chiffrements qui dépendent de la version SSL, des polices d'assurance qui imposent un niveau de cryptage acceptable et des dispositions législatives de chaque pays concernant l'exportation.

En tenant compte de ces informations, le protocole SSL détermine comment le serveur et le client vont négocier l'algorithme de chiffrement qu'ils vont utiliser pour s'authentifier mutuellement, transmettre les certificats et établir leur clé de session.

Les chiffrements qui suivent font référence aux algorithmes suivants :

- DES : Data Encryption Standard,
- DSA : Digital Signature Algorithm, algorithme utilise pour la signature numérique.
- KEA : Key Exchange Algorithme, pour échange de clés.
- MD5 : Message Digest algorithm developed by Rivest.
- RC2 and RC4 : Algorithmes de chiffrement.
- RSA : Algorithme de clés publiques pour cryptage et authentification.
- RSA Key Exchange : Algorithme d'échange de clés pour SSL basé sur RSA.
- SHA-1 : Secure Hash Algorithm, fonction de hachage.
- SKIPJACK : Algorithme classifié de clés symétriques
- Triple-DES : DES appliqué trois fois.

Les algorithmes d'échange de clés tels que RSA Key Exchange régissent la façon dont le serveur et le client vont déterminer les clés symétriques qu'ils vont utiliser durant la session SSL.

Lors de l'échange d'information pendant le protocole SSL handshake, le client et le serveur identifient le niveau de chiffrement le plus élevé commun, c'est celui qu'ils utiliseront pour la session SSL.

Le niveau de chiffrement qu'une société met en œuvre dépend de plusieurs facteurs :

- la sensibilité des donnés impliquées

- la rapidité du chiffrement.
- l'application des lois à l'exportation.

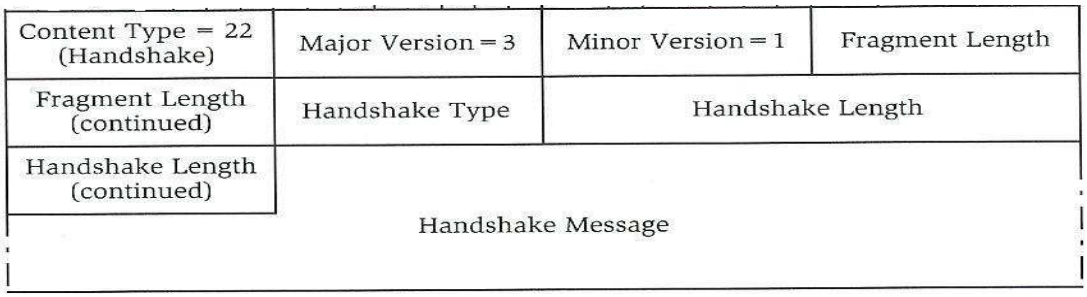
Une connexion SSL est définie par les paramètres suivants partagés entre le client et le serveur :

- Session identifié : un octet fixé par le serveur pour identifier la session
- Peer certificat : un certificat pour le serveur, éventuellement un autre pour le client
- Cipher Spec : définit l'algorithme de chiffrement symétrique et l'algorithme de condensation
- Master secret : clé de 48 octets négociée entre le serveur et le client
- Compression method : NULL pour l'instant (SSL 3.0/TLS)
- Is resumable : flag qui indique si de nouvelles connexions peuvent être créées à partir de cette session
- Server and client random : des octets aléatoires déterminés par le client et le serveur pour chaque connexion
- Server write (send) MAC secret : clé secrète utilisée par le serveur pour faire les MAC
- Client write (send) MAC secret : clé secrète utilisée par le client pour faire les MAC
- Server write (send) key : clé symétrique utilisée par le serveur pour chiffrer les données
- Client write (send) key : clé symétrique utilisée par le client pour chiffrer les données
- Initialization vectors : pour un algorithme de chiffrement par bloc en mode CBC. Le premier est fixé lors du handshake, les suivants sont les derniers blocs des précédents fragments chiffrés
- Sequence number : chaque message est numéroté de part et d'autre

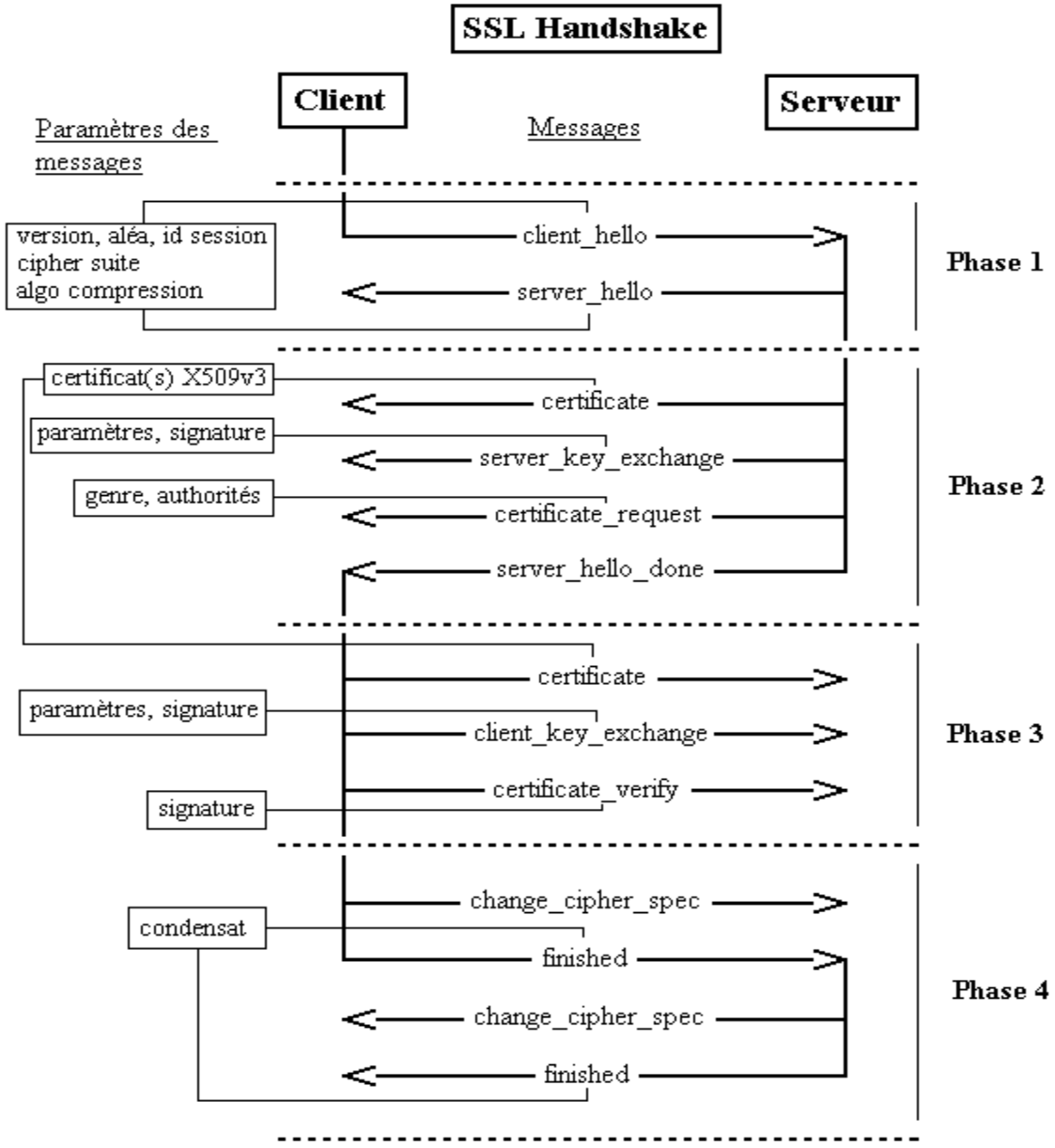
Le protocole SSL handshake : (protocole de négociation)

Ce protocole permet au client et au serveur de s'authentifier mutuellement, de négocie les algorithmes de chiffrement, de négocier les algorithmes de MAC et enfin de négocier les clés symétriques qui vont servir au chiffrement. Chaque message échangé entre le client et le serveur contient trois champs:

Type : indique l'objet du message
Lenght : indique la longueur du message
Content : contient les données transmises



Le protocole handshake perme d'établir une négociation entre le client et le serveur par l'étape suivant :



La négociation s'effectue de la façon suivante :

1. La phase de négociation est initiée par le client qui envoie un message du type «ClientHello». Ceci indique que le client désire établir une session sur sa connexion TCP. Il envoie un nombre aléatoire (qui sera utilise pour la détermination des clés), un identificateur de session et la liste des algorithmes et de compression qu'il est capable d'utiliser.

Les trois paramètres négociables sont la version de SSL/TLS, les algorithmes de sécurité et la méthode de compression.

2. Le serveur répond par une série des messages qui définissent les paramètres de sécurité du serveur :

- Il commence par un message du type Server Hello. Celui-ci fait savoir au client que le serveur a bien reçu son Client Hello (en envoyant le même numéro de session), envoie également un nombre

aléatoire et indique l'algorithme de compression et l'algorithme de sécurité qu'il a choisi parmi ceux proposés par le client.

- Le serveur envoie ensuite un message du type « Certificate » pour s'identifier si le client l'a demandé.

- Le serveur envoie alors un message du type « Server Key Exchange » pour transmettre les clés de sécurité si le client l'a demandé.

- Le serveur envoie éventuellement un message du type « Certificate Request », s'il désire que le client s'identifie.

- Le serveur indique enfin qu'il a termine sa série de messages en envoyant un message du type « Server Hello Done ».

Ceci permet au client de savoir qu'il n'a pas à attendre un message du type « Certificate Request ».

3. Le client s'embarque alors dans une série de messages qui définissent les paramètres de sécurité du client :

- Le message « Certificate » identifie le client, uniquement en réponse à un « Certificate Request ».

- Le message « Client Key exchange » permet de transmettre les clés de sécurité.

- Le client confirme qu'il a accepte le certificat envoyé par le serveur (si c'est le cas) en envoyant un message du type « Certificate Verify », qui devrait bien sur plutôt s'appeler « Certificate Verified ».

- Le client envoie alors un message du type « Change Cipher Specification » pour indiquer que les messages suivants seront cryptés.

- Le client termine par un message (crypté) du type « Finished ».

4. Le serveur répond en envoyant :

- un message du type « Change Cipher Specification ».

- suivi d'un message du type « Finished ».

Maintenant les données peuvent être échangées.

L'ordre des messages est imposé. La suite de message montrée à la figure ci-dessus peut être réinitialisée durant la session TCP sécurisée pour renégocier les paramètres de sécurité.

EXERCICES

Exercice 1

Donner le chiffré du message «the big bang theory» en utilisant les systèmes de chiffrement suivants :

a) César
b) Chiffrement mono-alphabétique à clé (clé : TELECOM)
c) Vigenère (clé :TELECOM). Donner la clé si le chiffrement du même message donne comme chiffré : ZHPBFEHAYGQFKOCY
d) Chiffrement par transposition à clé (clé : TELECOM)
e) Affine y= ax+b (mod26), clé :(a, b) = (17,1). Quelle est la clé utilisée si A est chiffré par G et E est chiffré par A ?

On donne la table de multiplication modulo26 :

×	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
2	4	6	8	10	12	14	16	18	20	22	24	0	2	4	6	8	10	12	14	16	18	20	22	24
3	6	9	12	15	18	21	24	1	4	7	10	13	16	19	22	25	2	5	8	11	14	17	20	23
4	8	12	16	20	24	2	6	10	14	18	22	0	4	8	12	16	20	24	2	6	10	14	18	22
5	10	15	20	25	4	9	14	19	24	3	8	13	18	23	2	7	12	17	22	1	6	11	16	21
6	12	18	24	4	10	16	22	2	8	14	20	0	6	12	18	24	4	10	16	22	2	8	14	20
7	14	21	2	9	16	23	4	11	18	25	6	13	20	1	8	15	22	3	10	17	24	5	12	19
8	16	24	6	14	22	4	12	20	2	10	18	0	8	16	24	6	14	22	4	12	20	2	10	18
9	18	1	10	19	2	11	20	3	12	21	4	13	22	5	14	23	6	15	24	7	16	25	8	17
10	20	4	14	24	8	18	2	12	22	6	16	0	10	20	4	14	24	8	18	2	12	22	6	16
11	22	7	18	3	14	25	10	21	6	17	2	13	24	9	20	5	16	1	12	23	8	19	4	15
12	24	10	22	8	20	6	18	4	16	2	14	0	12	24	10	22	8	20	6	18	4	16	2	14
13	0	13	0	13	0	13	0	13	0	13	0	13	0	13	0	13	0	13	0	13	0	13	0	13
14	2	16	4	18	6	20	8	22	10	24	12	0	14	2	16	4	18	6	20	8	22	10	24	12
15	4	19	8	23	12	1	16	5	20	9	24	13	2	17	6	21	10	25	14	3	18	7	22	11
16	6	22	12	2	18	8	24	14	4	20	10	0	16	6	22	12	2	18	8	24	14	4	20	10
17	8	25	16	7	24	15	6	23	14	5	22	13	4	21	12	3	20	11	2	19	10	1	18	9
18	10	2	20	12	4	22	14	6	24	16	8	0	18	10	2	20	12	4	22	14	6	24	16	8
19	12	5	24	17	10	3	22	15	8	1	20	13	6	25	18	11	4	23	16	9	2	21	14	7
20	14	8	2	22	16	10	4	24	18	12	6	0	20	14	8	2	22	16	10	4	24	18	12	6
21	16	11	6	1	22	17	12	7	2	23	18	13	8	3	24	19	14	9	4	25	20	15	10	5
22	18	14	10	6	2	24	20	16	12	8	4	0	22	18	14	10	6	2	24	20	16	12	8	4
23	20	17	14	11	8	5	2	25	22	19	16	13	10	7	4	1	24	21	18	15	12	9	6	3
24	22	20	18	16	14	12	10	8	6	4	2	0	24	22	20	18	16	14	12	10	8	6	4	2
25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1

Exercice 2

On code numériquement l'alphabet comme éléments 0, 1. . . , 25. (A=0, B=1, …).
Et soit la transformations affine définie par : f(x) = ax +b modulo26 , qui à toute lettre en clair x associé la lettre chiffrée f(x).

1. Montrer que, pour des valeurs particulières de a et b, ce chiffrement est un chiffrement de César.
2. Donner la méthode de déchiffrement de ce système pour a et b quelconque.
3. En déduire, pour une clé (a,b)=(3,5), le chiffrement du message en clair « MASTER TELECOM », et le déchiffrement du message chiffré « RWFLF ».
4. Déterminer la clé (a,b) si E est chiffrée par G, et T par H.

Exercice 3

Chiffrer le message : « Good Morning Veitnam», avec les méthodes suivantes:

- César
- Rot13
- Chiffrement par substitution mono-alphabétique à clé (clé=topsecret)
- Vigenère (clé=topsecret)
- Zig Zag de profondeur 3
- Transposition à grille (clé=topsecret)
- ADFGVX (clé=topsecret)

Exercice 4

On considère que les lettres de l'alphabet sont numérotées de 0 à 25 (A=0, B=1 ...).

1. Montrer, à travers un exemple, que la transformation lettre par lettre dans le chiffrement de Vigenère se formalise mathématiquement, par :

   Lettre codée = (Lettre en clair + Clé) modulo 26

2. En déduire l'équation de déchiffrement.

Exercice 5

Le chiffrement de HILL correspond à un chiffrement affine à deux dimensions. On chiffre deux lettres à la fois.
Les lettres P_k et P_{k+1} du texte clair seront chiffrées respectivement C_k et C_{k+1} , avec la formule ci-dessous:

$$\begin{pmatrix} C_k \\ C_{k+1} \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} P_k \\ P_{k+1} \end{pmatrix} (\text{mod} 26)$$

1. Chiffrer le message « DZ » en utilisant la matrice clé suivante :

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix} = \begin{pmatrix} 3 & 2 \\ 1 & 3 \end{pmatrix}$$

2. Donner l'équation de déchiffrement

Exercice 6

Soit E_k une fonction de chiffrement binaire par bloc de taille fixe (4 bits) tel que :
A tout message en clair m_i , on associe un chiffré $c_i = E_k(m_i)$.
Le tableau suivant donne la correspondance entre les messages m_i et leurs chiffrés c_i :

m_i	c_i
0000	0001
0001	1001
0010	0000
0011	1000
0100	0011
0101	1011
0110	0010
0111	1010
1000	0101
1001	1101
1010	0100
1011	1100
1100	0111
1101	1111
1110	0110
1111	1110

1. Donner le chiffré du message *M* suivant : *M*= 10110001, avec les modes d'opérations :
    a) ECB
    b) CBC (valeur initiale IV= 1010)
    c) CFB (valeur initiale IV= 1010)
    d) OFB (valeur initiale IV= 1010)
    e) CTR (valeur initiale T= 1010)

2. Déchiffrer le message chiffré « 11111111 » par CBC (valeur initiale IV= 1010)

Exercice 7

Le chiffrement de Vigenère se formalise mathématiquement, par :

Lettre codée = (Lettre en clair + Clé) modulo 26

Où : les lettres de l'alphabet sont numérotées de 0 à 25 (A=0, B=1 ...).

3. Donner le chiffré du message « MASTER GT M UN» (clé=TOPSECRET)
4. Quelle est l'équation de déchiffrement.
5. Un message m est chiffré par le chiffrement ADFGVX avec une clé K_0 .
   Cette clé est à son tour chiffrée par Vigenère avec la clé : OR.
   On donne
   - le chiffré de m : VVDAAFV ADADAAX FVADGFD XFXAAVV
   - et le chiffré de la clé K_0 : XRBLOIM

   Déchiffrer m.

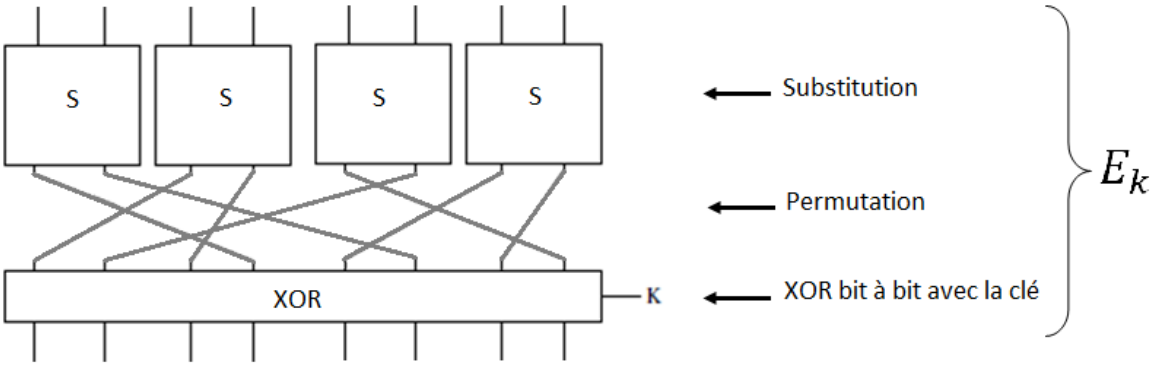
Exercice 8

Le mode d'opération CBC est donné par :

$$c_i = E_k(m_i \oplus c_{i-1})$$

Où E_k est une fonction de chiffrement binaire par bloc en mode EBC.

1. Donner l'équation de déchiffrement du mode CBC
2. Soit le chiffrement E_k , travaillant sur des blocs de 8 bits, représenté sur la figure suivante :



Où : les **S** boxes sont telles que :

X	00	01	10	11
$S(X)$	10	11	00	01

Et la clé est égal à : 11001001

- En utilisant le mode CBC (IV=00111100), Chiffrer le message suivant : M= 000011110101010110100110

Exercice 9

1. L'algorithme AES est un chiffrement par bloc itératif qui repose sur 4 opérations, les quelles ?
2. Donner le résultat de l'opération « SubBytes » de : 3d (code hexadécimal)
3. Quel est le résultat de l'opération « ShiftRows » sur le bloc suivant ?

04	e0	48	28
66	cb	f8	06
81	19	d3	26
e5	9a	7a	4c

4. Calculer dans le corps AES , les quantités suivantes (code hexadécimal) :
   - 66 + fa
   - 02 * bf
   - 11* de + 02 * bf

On donne

- le polynôme de Rinjdael : $R(x) = x^8 + x^4 + x^3 + x + 1$
- La table de substitution

hex	y															
	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
5	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
8	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
a	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
b	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
d	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
f	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

x

Exercice 10

Expliquer le principe de chiffrement/déchiffrement RSA à travers un exemple : on prendra n=33, p=11, q=3 et le message à chiffrer M=9.

Exercice 11

1. Le standard DSA est basé sur un problème calculatoire difficile, lequel ?
2. Soit un message m dont le haché est H(m)=2, générer la signature DSA du message m. On prendra (p, q, g)=(11,5,2), et comme clé privée la valeur 3.

Exercice 12

1. Donner la différence :

a) entre vulnérabilité et menace b) entre menace et attaque

2. Quelle est la différence des vulnérabilités dans la conception et dans l'implémentation ?
3. Quelle est la différence entre dommage et altération d'un système ?
4. Quelles sont les trois principaux éléments à sécuriser dans un système d'information ?
5. Quels sont les endroits où il faudra protéger l'information ?

Exercice 13

Classer les événements suivants en vulnérabilités et menaces, puis les menaces en accidentelles et intentionnelles, enfin les menaces intentionnelles en passives et actives :

• Faibles mots de passe • Interception d'émissions • Réseau ouvert (sans authentification) • Défaut de conception dans l'architecture réseau • Panne du matériel (hardware failure)	• Sabotage • Absence de redondance (des serveurs, des systèmes de communications, …) • Effacement de la mémoire • Panne de disque (disk failure) • Corruptions des données (data

• Envoi d'un script malveillant attaché à une page web • Défaillance du logiciel (software crash) • Erreur d'opérateur (operator error) • Action malveillante intérieure (malicious inside action) • Absence de sauvegarde (backup) • Logiciel malveillant (malware) • Ports non standards ouverts • Incendie, explosion, inondation, séisme (fire, explosion, flood, earthquake) • Espionnage	corruption) • Absence de contrôle d'accès • Données inexactes (inaccurate data) • Accès non autorisé (unauthorized access) • Prise de contrôle d'un site web • Cracking • Communications défaillantes ou corrompues • Réseau wifi mal configuré • Coupure d'électricité (power loss) • Inondation de messages

Exercice 14

1. Comment appelle-t-on l'attaque permettant d'utiliser gratuitement un réseau téléphonique ?
2. Donner la différence entre les attaques : spoofing, sniffing et scanning
3. Expliquer les différences entre trojan, spam, ver (worm), bombe logique, virus, spyware.
4. Qu'est ce qu'un déni de service ? Comment peut-on réaliser une attaque de type déni de service distribué ?

Exercice 15

1. Définir ce qui est une PSSI.
2. Donner les objectifs de sécurité et indiquer par quels moyens ils peuvent être assurés.

Exercice 16

Expliquer à l'aide de schémas comment deux utilisateurs peuvent-ils échanger des informations chiffrées et authentifiées (chiffrement et signature). On utilisera par exemple les algorithmes AES, RSA et SHA.

Exercice 17

1. Définir ce qui est un contrôle d'accès physique et logique.
2. Le contrôle d'accès logique repose sur 3 processus, lesquels ?

Exercice 18

1. Qu'est ce que la gestion des identités ?
2. Que signifie l'identification, aux yeux de l'utilisateur ? aux yeux du système de contrôle d'accès ?

3. Quelle est le processus permettant de vérifier l'identité d'un utilisateur ?

Exercice 19

1. Sur quels éléments se basent les décisions d'autorisation d'accès ?
2. Quelle est la différence entre habilitations et autorisations ?
3. Quelles conditions permettent de déclencher le processus d'autorisation ?

Exercice 20

1. Quels sont les facteurs d'authentifications ? donner des exemples.
2. Quels sont les éléments que doit fournir un utilisateur pour s'authentifier auprès d'un système de contrôle d'accès ?
3. Citer quelques méthodes d'authentification.

Exercice 21

1. Quels sont les avantages et inconvénients d'utilisation d'une authentification par mot de passe statique ?
2. Enumérer les menaces possibles sur le mécanisme d'authentification par mot de passe statique.
3. Dans un processus d'authentification par mot de passe statique, comment se fait la vérification du mot de passe par le serveur d'authentification ?

Exercice 22

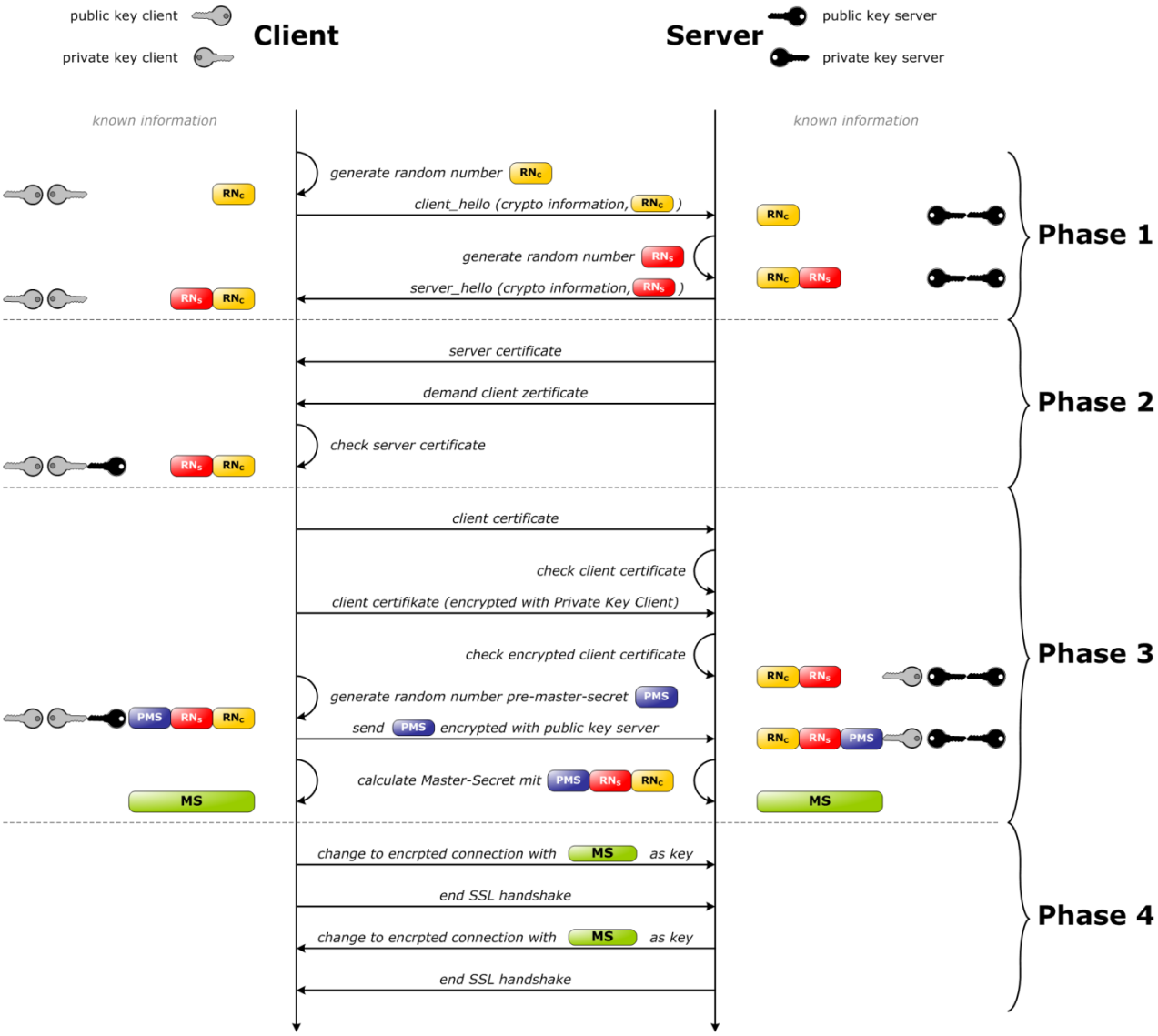
A quelle fin utilise-t-on les fonctions de hachage ? Donner les différents standards.

Exercice 23

1. Donner les différents mécanismes d'IPSec ?
2. Quel est le rôle du protocole IKE dans IPSec ?

Exercice 24

A quoi sert un certificat électronique? Expliquer l'authentification par certificat dans le fonctionnement de SSL (Figure 1 ci-dessous).


public key client
private key client
Client
Server
public key server
private key server
known information
known information
generate random number RNc
client_hello (crypto information, RNc)
generate random number RNs
server_hello (crypto information, RNs)
Phase 1
server certificate
demand client zertificate
check server certificate
Phase 2
client certificate
check client certificate
client certifikate (encrypted with Private Key Client)
check encrypted client certificate
generate random number pre-master-secret PMS
send PMS encrypted with public key server
calculate Master-Secret mit PMS RNs RNc
MS
MS
Phase 3
change to encrpted connection with MS as key
end SSL handshake
change to encrpted connection with MS as key
end SSL handshake
Phase 4

CORRIGES DES EXERCICES

Exercice 1

a) Chiffrement de César (avec décalage de 3 lettres)

Le message chiffré est : WKHELJEDQJWKHRUB

b) Chiffrement mono-alphabétique à clé :

Clé : TELECOM

La clé nettoyée est :TELCOM

Le message chiffré est : DPO EQN ETWN DPOXAJ

c) Chiffrement de vigenère

Clé : TELECOM

Le message chiffré est : MLPFKUNTRRXJSAKC

La clé utilisé est : GALAXY

d) Chiffrement par transposition

Clé :TELECOM

Le message chiffré est : IHHNYBTEGBOGETAR

e) Chiffrement Affine

y=ax+b(mod26) , clé : (a,b)=(17,1)

On a :

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

Exemple:

Pour chiffrer T , on a:

$(17\times19+1) \bmod 26 = 324 \bmod 26 = 12 \rightarrow M$

De la même manière, le chiffré du message s'en déduit facilement.

- Déchiffrement affine

$x=a^{-1}(y-b) \bmod 26$

a^{-1}est l'inverse de a (mod 26)

$(a.\ a^{-1}= 1 \bmod 26)$

Exemple : l'inverse de 17 est 23 car $17.\ 23 = 391= 1 \bmod 26$

Comment déterminer la clé :

Si A→G

E→A

On a :

$a.0+b= 6 \bmod 26 \Rightarrow b=6$

$4+b= 0 \bmod 26 \Rightarrow 4.a+6=0 \bmod 26 \Rightarrow 4a= -6 \bmod 26 =20 \bmod 26 \Rightarrow a=5$

Un autre exemple :

Si B→T

C→K

On a :

$a.1+b= 19 \bmod 26$

$a.2+b= 10 \bmod 26$

$$\Rightarrow \begin{vmatrix} a+b=19 \\ 2a+b=10 \end{vmatrix} \Rightarrow \begin{vmatrix} b=19-a \\ a+19=10 \end{vmatrix}$$

$$\Rightarrow \begin{vmatrix} b=19-a \\ a=-9\ mod26 \end{vmatrix} \Rightarrow \begin{vmatrix} b=2\ mod26 \\ a=17\ mod26 \end{vmatrix}$$

Exercice 5

Chiffrement de Hill : $\begin{bmatrix} c_k \\ c_{k+1} \end{bmatrix} = \begin{bmatrix} a & c \\ b & d \end{bmatrix} \begin{bmatrix} P_K \\ P_{K+1} \end{bmatrix} \bmod 26$

Les lettres en clair : $P_K et\ P_{K+1}$

Les lettres chiffrées : $c_k et\ c_{k+1}$

Calcul matriciel :

$c_k = aP_K + bP_{K+1}$(mod26)

$c_{k+1} = cP_K + dP_{K+1}$ (mod26)

1. Chiffrement de DZ avec la clé : $\begin{bmatrix}3 & 2\\1 & 3\end{bmatrix}$

D→3

Z→25

On a: 3×3+2×25 = 59 mod26 = 7

1×3+3×25 = 78 mod26 = 0

Donc DZ est chiffrée par HA

2. Equation de déchiffrement de Hill

Il faut déterminer $P_K\ et\ P_{K+1}$en fonction de $c_k et\ c_{k+1}$

Il faut pour cela inverser la matrice $\begin{bmatrix}a & b\\c & d\end{bmatrix}$

Donc :

$$\begin{bmatrix}P_K\\P_{K+1}\end{bmatrix}=\begin{bmatrix}a & b\\c & d\end{bmatrix}^{-1}\begin{bmatrix}c_k\\c_{k+1}\end{bmatrix}$$

$$\begin{bmatrix}a & c\\b & d\end{bmatrix}^{-1} = \frac{1}{ad-bc}\begin{bmatrix}d & -b\\-c & a\end{bmatrix} \text{mod26}$$

Exemple :

$\begin{bmatrix}3 & 2\\1 & 3\end{bmatrix}$ → matrice inverse $\frac{1}{7}\begin{bmatrix}3 & -2\\-1 & 3\end{bmatrix}$mod26

Soit : $\begin{bmatrix}3 & 2\\1 & 3\end{bmatrix}^{-1} = 7^{-1}\begin{bmatrix}3 & -2\\-1 & 3\end{bmatrix}$mod26

7^{-1} est l'inverse de 7 mod 26

$7^{-1} = 15$

D'où $\begin{bmatrix}3 & 2\\1 & 3\end{bmatrix}^{-1} = 15\begin{bmatrix}3 & -2\\-1 & 3\end{bmatrix} mod\ 26 = \begin{bmatrix}19 & 22\\11 & 19\end{bmatrix}$

Déchiffrement

Exemple : déchiffrer HA

c_k= H $\rightarrow$ 7

c_{k+1} = A $\rightarrow$ 0

$$\begin{bmatrix}P_K\\P_{K+1}\end{bmatrix} = \begin{bmatrix}19 & 22\\11 & 19\end{bmatrix}\begin{bmatrix}7\\0\end{bmatrix}$$

On a :

19×7+22×0 = 133 mod26 = 3 $\rightarrow$D

11×7+19×0 =77 mod26 = 25 $\rightarrow$Z

Exercice 6

M= 10110001 donc m_1=1011, m_2=0001

a) Mode ECB

$c_i=E_k(m_i)$

donc : $c_1 = E_k(m_1) = E_k(1011) = 1100$

$c_2 = E_k(m_2) = E_k(0001) = 1001$

b) Mode CBC (c_0=1010)

$c_i= (m_i \oplus c_{i-1})$

donc : $c_1= E_k(m_1 \oplus c_0)$

$= E_k[(1011) \oplus (1010)]$

$= E_k\ (0001)$

= 1001

$c_2 = E_k(m_2 \oplus c_1)$

$= E_k[(0001) \oplus (1001)]$

$= E_k(1000)$

$= 0101$

c) Mode CFB (c_0=1010)

$ci = m_i \oplus E_k(ci_{-1})$

donc : $c_1 = m_1 \oplus E_k(c_0)$

$= (1011) \oplus E_k(1010)$

$= (1011) \oplus (0100)$

$= 1111$

$c_2 = m_2 \oplus E_k(c_1)$

$= (0001) \oplus E_k(1111)$

$= (0001) \oplus (1110)$

$= 1111$

d) Mode OFB (c_0=1010)

$Z_O = c_0$; $Z_i = E_k(Zi_{-1})$, $c_i = m_i \oplus Z_i$

donc : $Z_O = c_0 = 1010$

$Z_1 = E_k(Z_0) = E_k(1010) = 0100$

$c_1 = m_1 \oplus Z_1$

$= (1011) \oplus (0100)$

$= 1111$

$Z_2 = E_k(Z_1) = E_k(0100) = 0011$

$c_2 = m_2 \oplus Z_2$

$= (0001) \oplus (0011)$

$= 0010$

e) Mode CTR (T=1010)

$ci = m_i \oplus E_k (T_{i+1})$

donc :

$c_1 = m_1 \oplus E_k(T+1)$

$= (1011) \oplus E_k(1010+1)$

$= (1011) \oplus E_k(1011)$

$= (1011) \oplus (1100)$

$= 0111$

Exercice 7

1. FOHLITXXFNB
2. Lettre en clair = (Lettre codée - Clé) modulo 26
3. Clé K_0 : JANUARY

   Message **m** : HAPPY YEAR 2016

Exercice 8

1. $m_i = c_{i-1} \oplus {E_k}^{-1}(c_i)$
2. $M = 000011110101010110100110 = m_1 m_2 m_3$

   $m_1 = 00001111 \rightarrow c_1 = 11111010$

   $m_2 = 01010101 \rightarrow c_2 = 10001011$

   $m_3 = 10100110 \rightarrow c_3 = 10010011$

Exercice 9

1. SubBytes, ShiftRows, MixColumns, AddRoundKey
2. 27
3.

04	e0	48	28
cb	f8	06	66
d3	26	81	19
4c	e5	9a	7a

4. 66 + fa = 9c
   02 * bf = 65
   11* de + 02 * bf = f4

Exercice 10

- On choisit deux nombres premiers p et q
- On calcule n = p*q
- On calcule la fonction d'Euler Ø(n) = (q-1)*(p-1)
- On choisit aléatoirement un nombre d premier avec Ø(n)
- On calcul e l'inverse de d modulo Ø(n)
- La clé privée est le couple (d, n)
- La clé publique est le couple (e, n)

Pour chiffrer un message M, on doit le représenter sous forme de nombre compris entre 0 et n-1

Et puis on calcule $C=M^e$ modulo (n)

Pour décrypter :

$M=C^d$ modulo (n)

Dans l'exercice on a : n=33, p=11, q=3 , soit : Ø(n)= 20, on prendra par exemple d=13.

Exercice 12

1.

a) Une vulnérabilité est une faille du système qui peut être exploité par un attaquant

Une menace est une violation potentielle de la sécurité, c'est-à-dire un signe qui laisse prévoir un danger.

Une menace peut provenir de l'environnement naturel, physique ou peut être le résultat d'actions humaines.

Les actions humaines peuvent être préméditées ou non, malveillantes ou non.

L'administrateur de la sécurité doit prendre en compte toutes les menaces possibles.

b) menace ≠ attaque

Une attaque est une réalisation d'une menace (un exploit). Une menace peut être réalisée par différentes attaques.

2. Une vulnérabilité dans la conception signifie une faille (un défaut) dans la spécification soit du matériel ou du logiciel

   Une vulnérabilité dans l'implémentation est une faille (un défaut) dans la réalisation de ce matériel ou de ce logiciel
3. Dommage : rendre totalement le système hors service

   Altération : provoquer partiellement des dégâts sur le système
4. Les informations

   Les systèmes d'exploitation

   Les réseaux
5. Sur support de stockage et sur support de transmission

Exercice 13

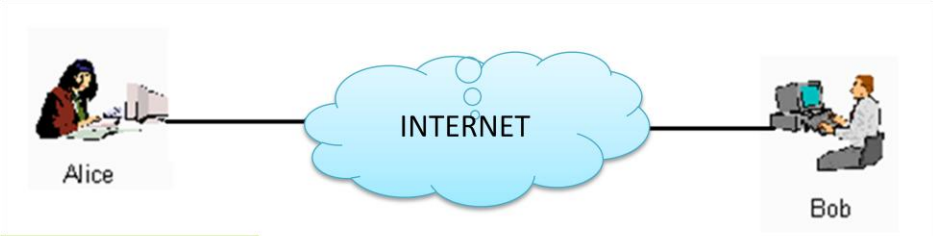
<table>
<tr><th>Vulnérabilités</th><th colspan="3">Menaces</th></tr>
<tr><td rowspan="3">• Faibles mots de passe
• Réseau ouvert (sans authentification)
• Défaut de conception dans l’architecture réseau
• Absence de sauvegarde (backup)
• Ports non standards ouverts
• Absence de redondance (des serveurs, des systèmes de communications, …)
• Absence de contrôle d’accès
• Réseau wifi mal configuré</td><td>Accidentelles</td><td colspan="2">• Panne du matériel (hardware failure)
• Défaillance du logiciel (software crash)
• Erreur d'opérateur (operator error)
• Incendie, explosion, inondation, séisme (fire, explosion, flood, earthquake)
• Panne de disque (disk failure)
• Données inexactes (inaccurate data)
• Coupure d'électricité (power loss)
• Communications défaillantes</td></tr>
<tr><td rowspan="2">Intentionnelles</td><td>Passives</td><td>Actives</td></tr>
<tr><td>• Interception d'émissions
• Espionnage</td><td>• Envoi d’un script malveillant attaché à une page web
• Action malveillante intérieure (malicious inside action)
• Logiciel malveillant (malware)
• Sabotage
• Effacement de la mémoire
• Corruptions des données (data corruption)
• Accès non autorisé (unauthorized access)
• Prise de contrôle d’un site web
• Cracking
• Communications corrompues
• Inondation de messages</td></tr>
</table>

Exercice 15

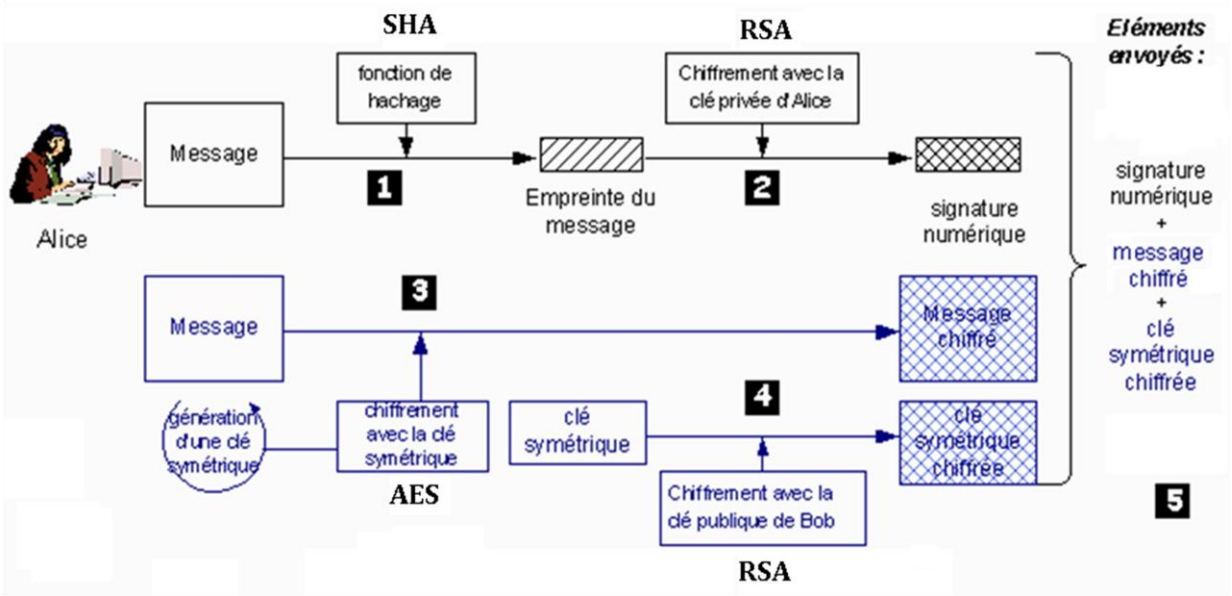
1. Une PSSI signifie une Politique de Sécurité du Système d'Information. Elle indique l'ensemble des mesures techniques, organisationnelles ou juridiques permettant d'atteindre des objectifs de sécurité prédéfinis.
2.

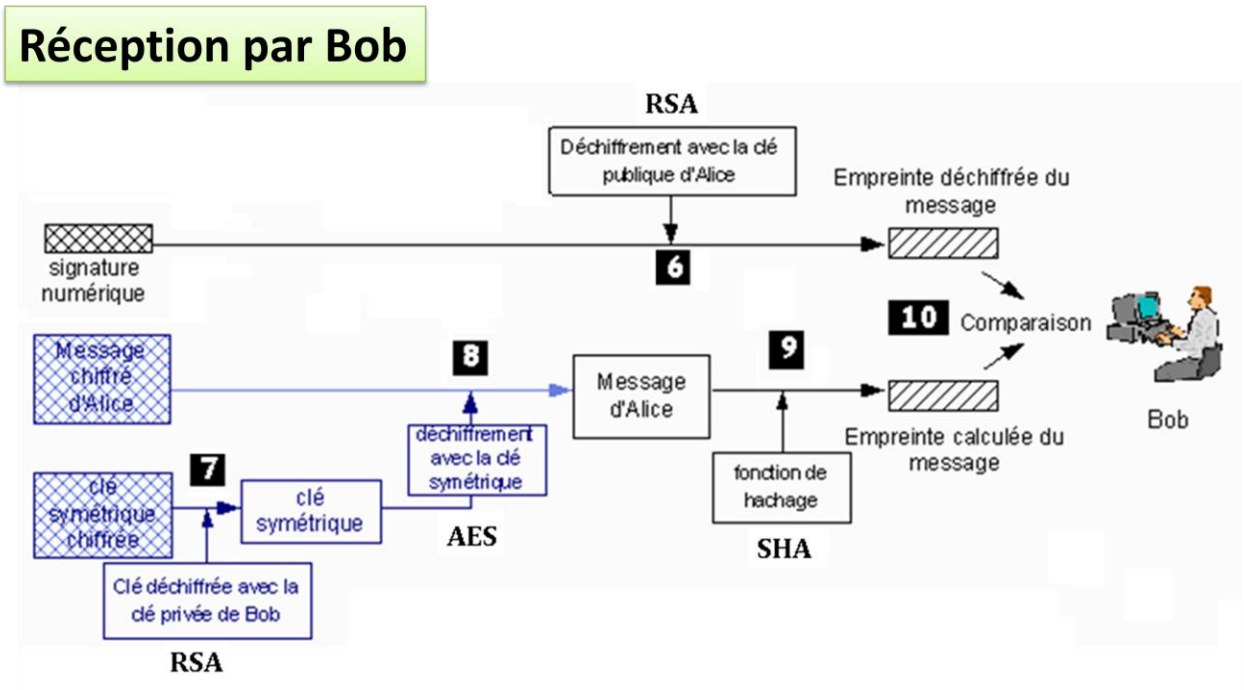
- Confidentialité (par le chiffrement)
- Intégrité (par la fonction de hachage)
- Authenticité (par l'authentification dans le cas de l'authenticité des utilisateurs, et par la signature dans le cas de l'authenticité des messages)
- Disponibilité (il n y a pas de mesure universelle, en général il faut combiner plusieurs solutions : firewall, contrôle d'accès, sauvegarde, redondance,…)
- Non-répudiation (par la signature)
- Traçabilité (par fichier log, caméras de surveillance, pointage, …)

Exercice 16



Emission par Alice





Exercice 17

1. Il existe deux types de contrôle d'accès : physique et logique.

Le contrôle d'accès physique permet de limiter l'accès physique aux infrastructures de l'entreprise par badges, carte magnétique, code d'accès, etc , aux seules personnes autorisées.

Le contrôle d'accès logique permet de limiter l'accès aux ressources informatiques de l'entreprise.

2. L'identification, l'authentification et l'autorisation

Exercice 18

1. C'est la désignation des identités des entités de l'entreprise (personnes, composants, logiciels, …)
2. Aux yeux de l'utilisateur, c'est déclarer son identité. Aux yeux du système de contrôle d'accès, c'est acquérir cette identité pour vérifier ses droits.
3. L'authentification.

Exercice 19

1. sur le profil des utilisateurs
2. les habilitations sont les droits d'accès associés à un utilisateur (par exemple sur un fichier, lire, écrire, supprimer, modifier les attributs, …). L'autorisation est la décision donnant l'accès aux ressources sollicitées par l'utilisateur.
3. L'identification et l'authentification réussies.

Exercice 20

1. Il existe 4 facteurs :

   - élément que l'on sait : un secret (mot de passe)

   - élément que l'on sait faire : déchiffrer un message

   - élément que l'on possède : carte à puce

   - élément que l'on est : empreinte digitale

2. Son identité et un ou plusieurs facteurs d'authentification.

3.

- Mot de passe statique

- Mot de passe statique stocké dans une carte magnétique activée par code PIN

- Mot de passe dynamique généré par un outil logiciel

- Mot de passe dynamique généré par un outil matériel

- Certificat X.509 dans le navigateur de l'ordinateur

- Certificat X.509 dans une clé USB

- Certificat X.509 dans une carte a puce

- Biométrie

Exercice 21

1. Avantages : coût, facilité de mise en œuvre et facilité d'utilisation

Inconvénients : perte, oubli, attaques faciles à mettre en pratique.

2.

- Crakage du mot de passe (par force brute, par dictionnaire)
- Capture du mot de passe (capture de trames sur le réseau, attaque man in the middle)
- Divulgation ou modification du fichier de mot de passe