RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE

MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE Dr. TAHAR MOULAY SAIDA

FACULTE : TECHNOLOGIE

DEPARTEMENT : INFORMATIQUE

Cryptologie

Présenté par :

Houacine Abdelkrim

Maître de conférences « B » en Informatique

Novembre 2021

Avant propos

Ce polycopié est une introduction à la cryptographie moderne utilisée dans la transmission et le stockage sécurisé de données. Il constitue un manuel de cours et d'exercices sur une partie du domaine de la cryptographie moderne destiné aux étudiants niveau master option sécurité informatique et cryptographie. L'accent mis sur les principes et les outils de bases mises en œuvre pour établir des schémas cryptographiques utilisés dans la pratique, les fondements mathématiques (mathématiques pour la cryptographie) sont décrits dans d'autres ressources dont on a fait références dans la bibliographie.

Table des matières

Sigles et Abréviations

ABE : Attributs Based Encryption.

AES : Advanced Encryption Standard.

CA : Certification Athority

CBC : Cipher Bloc Chaining

CFB : Cipher FeedBack.

DEA : Data Encryption Algorithm.

DES : Data Encryption Standard.

DoS : Déni de service.

ECB : Electronic Code-Book.

ECDH : Diffie–Hellman Elliptic Curve.

GPA : Générateur Pseudo-Aléatoire de symboles.

GSM : Global System for Mobile Communications.

IBE : Identite Based Encryption.

ICP : Infrastructure à clef publique.

IV : Valeur Initiale.

KDC : Centre de distribution de clés.

LFSR : Linear Feedback Shift Registers.

OFB : Output FeedBack..

PKI : Infrastructure à clef publique.

RSA : Ronald Rivest(R), Adi Shamir(S) et Leonard Adleman (A).

ZKP : Zero-knowledge proofs.

Introduction générale

Le but de ce polycopié est de fournir une présentation des différents concepts de base de la cryptographie moderne que nous avons trouvés de la plus grande utilité pratique dans le domaine de la sécurité informatique, pour fournir une base solide aux étudiants qui apprennent le sujet.

Ce polycopié est structuré en huit chapitres comme suit : Dans le premier chapitre, nous avons introduit la cryptographie, ainsi que la cryptographie ancienne ou on a donnés des exemples. Le deuxième chapitre décrit le domaine de la théorie d'information et les probabilités, ainsi que leur relation avec la sécurité. Le troisième chapitre décrit un type de chiffrement dit chiffrement par flot, ou on a présenté: les générateurs pseudos aléatoire et quelques exemples d'algorithme de ce type. Le quatrième chapitre est consacré à un autre type de chiffrement dit chiffrement symétrique ou à clé secrète à savoir les cryptosystèmes : DES(Data Encryption Standard) et AES(Advanced Encryption Standard). Le cinquième chapitre décrit le chiffrement asymétrique ou à clé publique en présentant : l'algorithme RSA, Le protocole d'échange de clés de Diffie-Hellman et le cryptosystème d'El Gamal. Le sixième chapitre décrit une application de la cryptographie à savoir les fonctions de hachage et la signature numérique. Le septième chapitre décrit le concept de l'autorité de certification ainsi que les protocoles d'authentification. Enfin, dans le huitième chapitre nous avons traité d'autres nouveaux schémas cryptographiques à clé publique à savoir les courbes elliptiques, Chiffrement basé identité (IBE- Identite Based Encryption) et Chiffrement basé attributs (ABE- Attributs Based Encryption). Nous trouverons en fin de ce polycopié une liste de références bibliographiques.

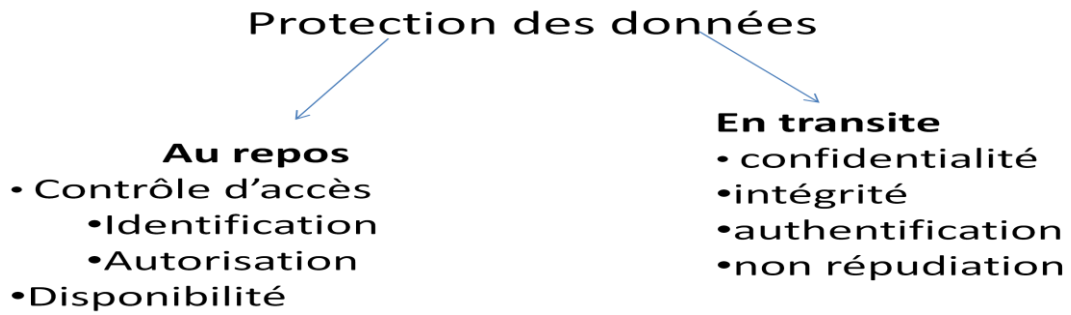
Chapitre 1

Introduction

1. Introduction

La sécurité informatique consiste à la protection des systèmes, de l'information et des services contre les menaces (accidentelles ou délibérées) atteignant leur confidentialité, intégrité ou disponibilité.

a. **disponibilité :** demande que l'information sur le système soit *disponible* aux personnes autorisées.
b. **Confidentialité :** demande que l'information sur le système ne puisse être *lue* que par les personnes autorisées.
c. **Intégrité :** demande que l'information sur le système ne puisse être *modifiée* que par les personnes autorisées.



La non-répudiation de l'origine fournit au récepteur une preuve empêchant l'émetteur de contester l'envoi d'un message ou le contenu d'un message effectivement reçu.

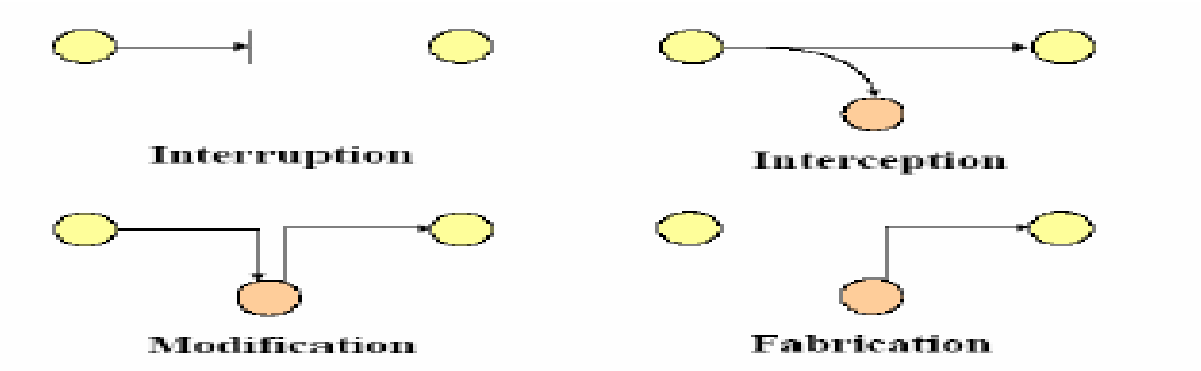


Figure 1.1 : Types des attaques

1.1 Principales attaques

Virus , Déni de service (DoS), Écoute du réseau (sniffer), Intrusion, Cheval de Troie, « social engeneering , L'homme du milieu, Espiogiciels (spyware)…

1.2 Protection

Formation Des Utilisateurs, Antivirus, Pare-Feu (Fire Wall), Authentification (Login, Mot De Passe, Biométrie (Empreinte), Clés USB, Carte à Puce) Et la **cryptologie**.

2. Cryptologie

La cryptographie est l'étude des méthodes d'envoi de messages codés de telle sorte que seul le destinataire puisse le décoder. Le message qu'on veut envoyer s'appelle le texte clair et le message codé, ou encrypté, s'appelle aussi cryptogramme.

Le processus de conversion d'un texte clair en message codé s'appelle chiffrement, ou codage, et le processus inverse s'appelle déchiffrement, ou décodage. Pour effectuer un codage, on suit une méthode précise appelée système de codage, ou système cryptographique, ou même encore cryptosystème. Un codage se fait donc à l'aide d'un système cryptographique, et celui-ci nécessite très souvent l'utilisation d'une clé de codage. Cette clé (un mot, un nombre, une grille) est nécessaire pour décoder le message chiffré. En d'autres termes, la clé modifie le comportement du mécanisme de codage et de décodage.

Ce qu'il faut retenir :

Cryptologie : Science du secret avec deux composantes complémentaires : [01]

a. Cryptographie : étude et conception des procédés de chiffrement des informations.

b. Cryptanalyse : analyse des textes chiffrés pour retrouver les informations dissimulées.

Un **cryptogramme** est un message chiffré.

2.1 La cryptographie ancienne

Il y a deux grandes familles de chiffrements classiques:

- Les chiffrements à répertoire
- Les chiffrements à clefs secrètes qui se subdivisent en deux familles
  – les chiffrements de transposition ou de permutation qui sont des chiffrements par blocs.
  – les chiffrements de substitution qui peuvent être des chiffrements par blocs ou par flots

2.2 Les chiffrements à répertoire

Ils consistent en un dictionnaire qui permet de remplacer certains mots par des mots différents. Ils sont très anciens et ont été utilisés intensivement jusqu'au début du 20-ième siècle.

On peut par exemple créer le dictionnaire suivant:

Rendez-vous ↔ 175 ; midi ↔ à vendre ; demain ↔ oiseaux ; Villetaneuse ↔ au marché

Le chiffrement de : « RENDEZ-VOUS DEMAIN MIDI VILLETANEUSE »

Donne :

175 OISEAUX A VENDRE AU MARCHE

Ces codes manquent de souplesse :

- ils ne permettent pas de chiffrer des mots nouveaux sans un accord préalable entre l'expéditeur et le destinataire.
- Il faut qu'ils échangent des documents ce qui accroît le risque d'interception du code.
- Ils ne sont pas adaptés à des usages intensifs entre de nombreux correspondants.
- Ils ne sont pratiquement plus utilisés pour les usages publics.

2.3 Les chiffrements de transposition ou de permutation

Dans les chiffrements de permutation on partage le texte en blocs, on garde le même alphabet mais on change la place des lettres à l'intérieur d'un bloc (on les permute).

Exemple : On veut envoyer le message suivant

RENDEZ VOUS DEMAIN MIDI VILLETANEUSE

L'expéditeur et le destinataire du message se mettent d'accord sur une grille de largeur fixée à l'avance (ici une grille de 6 cases de large).

L'expéditeur écrit le message dans la grille en remplaçant les espaces entre les mots par le symbole □. Il obtient:

R	E	N	D	E	Z
□	V	O	U	S	□
D	E	M	A	I	N
□	M	I	D	I	□
V	I	L	L	E	T
A	N	E	U	S	E

Il lit le texte en colonne et obtient ainsi le message crypté:

R□D□VAEVEMINNOMILEDUADLUESIIESZ□N□TEC

Pour augmenter la sécurité les deux interlocuteurs peuvent décider l'ajout d'une clef secrète constituée par l'ordre de lecture des colonnes.

Exemple 1.1

On choisit la clé: **CAPTER**

On numérote les colonnes en fonction du rang des lettres du mot CAPTER dans l'alphabet c'est à dire

2, 1, 4, 6, 3, 5

On obtient

E	R	D	Z	N	E
V	□	U	□	O	S
E	D	A	N	M	I
M	□	D	□	I	I
I	V	L	T	L	E
N	A	U	E	E	S

et on lit les colonnes dans l'ordre indiqué :

EVEMINR□D□VADUADLUZ□N□TENOMILEESIIES

On a 6! Codes différents.

Pour déchiffrer le message précédent on range en colonne sur la grille en suivant l'ordre des colonnes donné par la clé :

R	E	N	D	E	Z
□	V	O	U	S	□
D	E	M	A	I	N
□	M	I	D	I	□
V	I	L	L	E	T
A	N	E	U	S	E

C'est un chiffrement à clé secrète ou chiffrement symétrique car la même clé sert pour chiffrer et déchiffrer.

Pour éviter le chiffrement de la totalité du message avant de commencer la transmission on fragmente le message en blocs de taille m=*k* x *l* ou *k* est la largeur de la grille et *l* sa hauteur.

2.4 Les chiffrements de substitution

Dans les chiffrements de substitution par flots ou par blocs l'ordre des lettres est conservé mais on les remplace par des symboles d'un nouvel alphabet suivant un algorithme précis.

Dans ce qui suit nous allons voir deux exemples de chiffrements, le chiffre de César qui est un chiffrement mono-alphabétique et le chiffre de Vigenère poly-alphabétique.

2.4.1 Le Chiffre de César (100 av. J.-C. - 44 av. J.-C.)

Le chiffre de César est la méthode de cryptographie la plus ancienne. Il consiste en une **substitution mono-alphabétique** : chaque lettre est remplacée("substitution") par une *seule* autre("mono-alphabétique"), selon un certain décalage dans l'alphabet ou de façon arbitraire exemple : avec un décalage de 3 lettres : **A** devient **D**, **B** devient **E**, **C** devient **F**, etc.

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
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

« MASTER SIC « devient « PDVWHU VLF »

- Niveau sécurité : le chiffre de César n'est pas fiable du tout, **attaques exhaustives** (tester toutes les décalages un à un), ne demanderaient que très peu de temps.

2.4.2- Le chiffre de Vigenère (1523-1596)

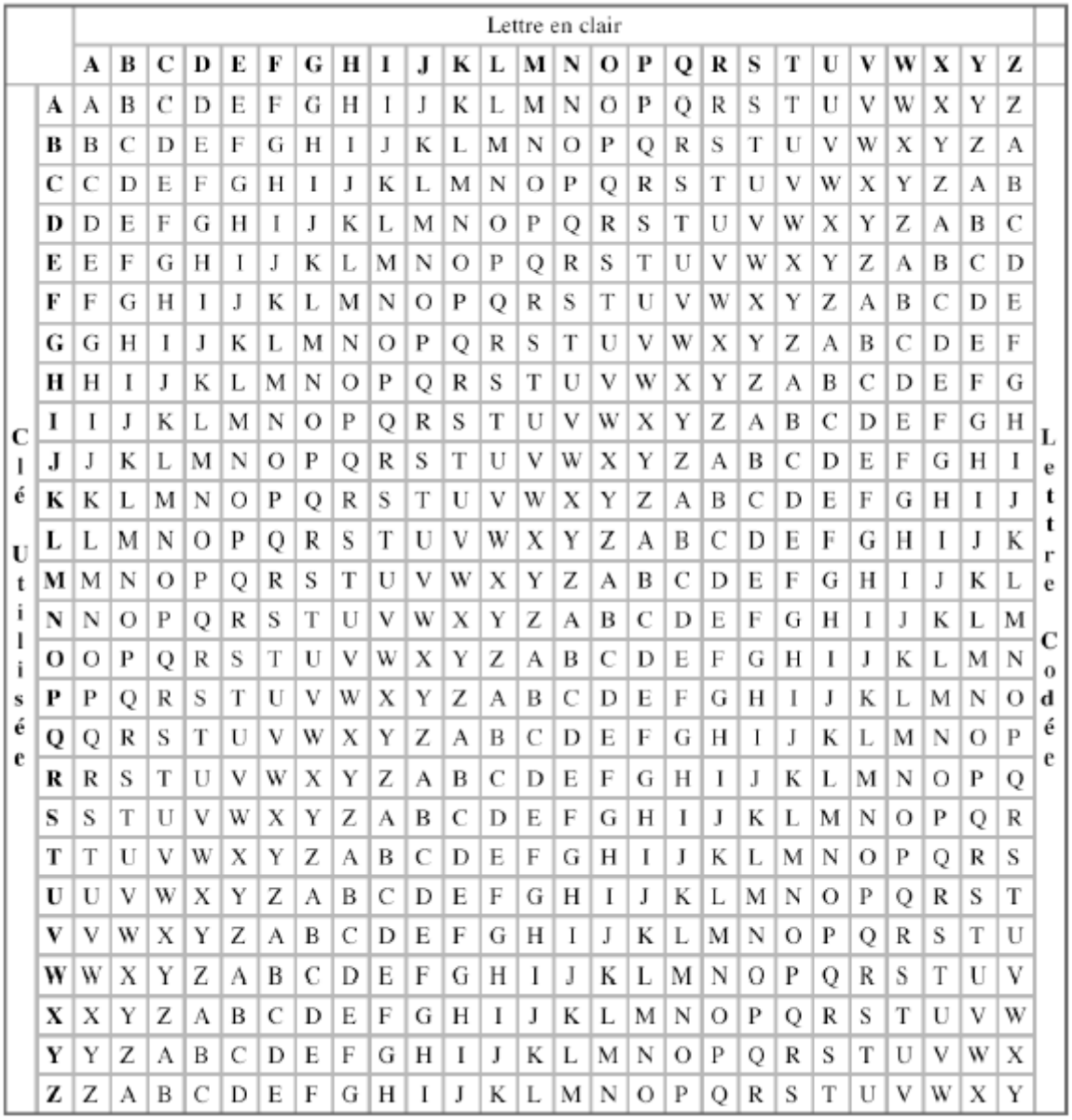
Le **chiffre de Vigenère** est une amélioration décisive du chiffre de César. C'est un chiffrement par translation polyalphabetique. Sa force réside dans l'utilisation non pas d'un, mais de 26 alphabets décalés pour chiffrer un message. On peut résumer ces décalages avec un carré de Vigenère figure 1.2 .

Ce chiffre utilise une **clef** qui définit le décalage pour chaque lettre du message.

Le chiffrement

- Pour chaque lettre en clair, on sélectionne la colonne correspondante
- pour une lettre de la clé on sélectionne la ligne adéquate,
- puis au croisement de la ligne et de la colonne on trouve la lettre codée.

La lettre de la clé est à prendre dans l'ordre dans laquelle elle se présente et on répète la clé en boucle autant que nécessaire.



Clé Utilisée \ Lettre en 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 1.2 : Le Carrée de Vigenère

Exemple 1.2

Chiffrons le texte "CHIFFRE DE VIGENERE" avec la clef "BACHELIER" (cette clef est éventuellement répétée plusieurs fois pour être aussi longue que le texte clair).

Clair	C	H	I	F	F	R	E	D	E	V	I	G	E	N	E	R	E
Clé	B	A	C	H	E	L	I	E	R	B	A	C	H	E	L	I	E
Décalage	1	0	2	7	4	11	8	4	17	1	0	2	7	4	11	8	4
Chiffré	D	H	K	M	J	C	M	H	V	W	I	I	L	R	P	Z	I

Exemple1.3

clé : MUSIQUE

texte : j'adore ecouter la radio toute la journee

Texte en clair : j'adore ecouter la radio toute la journee

Clé répétée : m usiqu emusiqu em usiqu emusi qu emusiqu

Colonne O, ligne I: on obtient la lettre W.

Colonne D, ligne S: on obtient la lettre V.

Colonne A, ligne U: on obtient la lettre U.

Colonne J, ligne M: on obtient la lettre V.

Le texte chiffré est alors :

V'UVWHY IOIMBUL PM LSLYI XAOLM BU NAOJVUY.

Pour déchiffrer ce texte :

- on regarde pour chaque lettre de la clé répétée la ligne correspondante,
- on y cherche la lettre codée.
- La première lettre de la colonne que l'on trouve ainsi est la lettre décodée.

Texte codé : V'UVWHY IOIMBUL PM LSLYI XAOLM BU NAOJVUY

Clé répétée : M USIQU EMUSIQU EM USIQU EMUSI QU EMUSIQU

Ligne I, on cherche W: on trouve la colonne O.

Ligne S, on cherche V: on trouve la colonne D.

Ligne U, on cherche U: on trouve la colonne A.

Ligne M, on cherche V: on trouve la colonne J.

2.4.4 Cryptanalyse de chiffrement par substitution

a. **A force brute** : en essayant les 26 ! permutations possibles de l'alphabet.
b. **L'analyse fréquentielle**, ou *analyse de fréquences*, est une méthode de cryptanalyse qui consiste à examiner la fréquence des lettres employées dans un message chiffré. Cette méthode est fréquemment utilisée pour décoder des messages chiffrés par substitution, dont un exemple très simple est le chiffre de César.

3. Exercices

Exercice 1.1

Chiffrez le texte suivant à l'aide de la méthode de Vigenère et de la clef « salut »:

« ilétaitunpetitnavire »

Exercice 1.2

On considère l'alphabet privé du W, soit 25 lettres. Polybe (200-125 av J.C) a proposé le mécanisme suivant : on range les lettres dans un tableau 5X5, en commençant par le mot clé (en supprimant les doublons), puis on contenue avec les lettres restantes de l'alphabet, dans l'ordre.

Par exemple : avec le mot-clé MYSTERE, on construit le tableau suivant :

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

Le chiffrement s'effectue alors en remplaçant chaque lettre par les deux chiffres : ligne+colonne qui indiquent sa position dans la graille, par exemple F est chiffré par 31.

Q1- Expliquer comment on peut cryptanalyser un tel système : par une attaque claire connu, puis attaque simple (seulement un chiffré).

Q2- chiffrez le message : « rendez-vous a deux heures département informatique ».

Chapitre 2

Théorie de l'information et la sécurité parfaite

1. Introduction

Comment mesurer la sécurité apportée par un système de cryptographie? C'est le mathématicien Claude Shannon qui en 1947 a répondu à cette question en développant la théorie de l'information, qui est une théorie probabiliste permettant de quantifier le contenu moyen en information d'un ensemble de messages, dont le codage informatique satisfait une distribution statistique précise. Dans ce chapitre on commence par un rappel sur les probabilités et on verra comment la théorie de l'information peut mesurer la sécurité d'un système de cryptographie.

2. Expérience aléatoire

Une expérience aléatoire est une activité, pas nécessairement scientifique (mais précise) qui produit des résultats qui dépendent du hasard, et dont on sait précisément décrire l'ensemble des résultats possibles. Cet ensemble de résultats possibles s'appelle l'espace échantillonal, et on le note ici Ω. Ainsi, tirer au hasard une lettre dans la phrase suivant constitue une expérience aléatoire.

« *toutes tes attitudes tentent de tromper ta tante et ses tiers* »

L'espace échantillonal de cette expérience est l'ensemble {a,e,i,m,n,o,p,r,s,t,u}

2.1 Evénement

Un événement est tout simplement un autre nom pour un sous-ensemble quelconque de l'espace échantillonal Ω. Pour notre exemple ci-haut, on peut considérer l'événement qui correspond à tirer une voyelle :

V = {a, e, i, o, u}

Si on tire une lettre au hasard dans la phrase (1), on peut obtenir une voyelle ou non. Si c'est le cas, on dit aussi que l'événement V s'est produit. Les événements sont désignés ici par des lettres majuscules A,B, . . .

2.2 Probabilité

La probabilité P(A) (notre chance de succès) qu'un événement se produise dans une expérience aléatoire, ou tous les résultats ont la même chance de se produire, on calcule P(A) comme : $P(A)= \frac{Card\ (A)}{Card\ (\Omega)}$. Le résultat est donc toujours un nombre entre 0 et 1.

2.3 Variables aléatoires

Une variable aléatoire discrète X, est constitué d'un ensemble fini X et une distribution de probabilité définie sur X. la probabilité que la variable aléatoire X prend la valeur x est notée P (X = x); ou P (x) si la variable aléatoire X est fixée. et on a : 0 <= P (x) pour tout x $\in$ X et $\sum_{x \epsilon X} P(x) = 1$.

Exemple2.1

On lance un dé. Soit X le résultat obtenu, les valeurs possibles : X={1,2,3,4,5,6} .

P(X=1)= P(X=2)= P(X=3)= P(X=4)= P(X=5)= P(X=6)=1/6.

2.4 Espérance d'une variable aléatoire

L'espérance d'une variable aléatoire est la valeur moyenne prise par cette variable notée E(X).

Si l'ensemble des valeurs que peut prendre X est $\{x_1, x_2, \ldots, x_k\}$, alors

$$E(X) = x_1P(X = x_1) + x_2P(X = x_2) + \cdots + x_kP(X = x_k).$$

L'interprétation de cette espérance est la suivante : Si l'on répète l'expérience aléatoire un grand nombre de fois, et qu'on calcule la moyenne des valeurs x_i obtenues, alors nous devrions (espérer) obtenir un résultat moyen prés de E(X).

2.5 Probabilité conditionnelle

Pour des événements quelconques A et B, on dénote P(A|B), la probabilité conditionnelle de A, étant donné B. Elle se calcule simplement comme le rapport :

$$P(A|B) = \frac{P(A \cap B)}{P(B)} \qquad (2.1)$$

Ou A ∩ B est l'intersection des événements A et B.

2.6 Indépendance

L'événement A est indépendant de B lorsque P(A) = P(A|B). De (1) on déduit que s'il ya indépendance entre A et B alors

$$P(A \cap B) = P(A) \cdot P(B) \qquad (2.2)$$

2.7 Entropie

Cette fonction introduite en 1948 par Claude Shannon dans son célèbre article « A mathematical theory of Communication »possède plusieurs interprétations équivalentes. Notamment, de façon informelle, H(X) correspond à une mesure moyenne de l'incertitude sur la valeur de X. Pour illustrer, considérons les expériences aléatoires qui consistent à :

A – Lancer (idéalement) une pièce de monnaie,

B – Lancer un dé (non pipé) `a 6 faces,

C – Prendre une carte (parfaitement au hasard) dans un paquet de 52 cartes.

Clairement, on a plus de chances de prédire correctement le résultat de l'expérience A, que de prédire celui des expériences B ou C. Nous dirons qu'il y a plus d'incertitude sur le résultat des expériences B ou C, que sur le résultat de l'expérience A. C'est cette incertitude que l'entropie va permettre de mesurer.

Si X prend ses valeurs dans un ensemble à n éléments, on peut encoder de façon naïve chaque élément sur $\log_2 n$ bits, H(X) fournit sur une échelle graduée de 0 à $\log_2 n$, une valeur traduisant l'incertitude moyenne sur X.

Exemple 2.2

Si H(X) = 10, ceci signifie qu'il suffit en moyenne de connaître 10 bits pour déterminer entièrement les valeurs prises par X. Ainsi ces valeurs peuvent être codées sur 10 bits au lieu de $\log_2 n$.

Définition 2.1 : soit X une variable aléatoire discrète dont les issues possibles x_i ont chacune pour probabilité p_i (1<= i <=n). L'entropie de X est définie par $H(X) = \sum_{i=1}^{n} p_i \log p_i$

Définition 2.2 : l'entropie conditionnelle de X sachant Y, et on note H(X|Y), la valeur qui mesure l'incertitude moyenne sur X connaissant Y.

Théorème 2.1 : soit X une variable aléatoire dont les n issues ont pour probabilités p_i pour 1<=i<=n :
alors on a H(X)<=log(n).de plus il y a égalité si et seulement si chaque pi vaut 1/n.

Théorème2.2 : soient X, Y deux variables aléatoires, On a H(X,Y)=H(X|Y)+H(Y)= H(Y|X)+H(X) .

Ou H(X,Y) est l'entropie de paires de variables aléatoires.

et $H(X|Y) = p_1 H(X|Y = a_1) + p_2 H(X|Y = a_2) + \ldots + p_k H(X|Y = a_k)$ (2.3)

ou p_i est la probabilité que la valeur de Y soit a_i et on a :

$$H(X|Y = a) = q_1 \log_2 1/q1 + \cdots + q_n \log_2 1/q_n \quad (2.4)$$

Avec q_j égal à la probabilité que $X = f_j$ sachant que $Y = a$. Avec

$$q_j = P(X = fj|Y = a).$$

Dans le cas ou X et Y sont indépendantes alors

$$H(X/Y) = H(X) \quad (2.5)$$

On interprète l'entropie conditionnelle comme mesurant le gain d'information obtenu en moyenne lorsque, connaissant la valeur de la variable aléatoire Y, on apprend qu'elle est la valeur de la variable aléatoire X.

On a $H(X, Y) \leq H(X) + H(Y)$ avec égalité exactement lorsque X et Y sont des variables aléatoires indépendantes.

2.8 Entropie d'un langage

Dans une suite aléatoire de lettre prises dans l'alphabet usuel, l'information portée par chacune des lettres est log(26)=4,7 bits, en tenant compte des inégalités de fréquences des différents lettres on obtient $H(Z_{26})$ =4.19 bits en Anglais et 3.97 en français.

Définition 2.3 : soit L un langage naturel sur un alphabet A son entropie est définie par

$H(L) = \lim_{n\to\infty}\left(\frac{H(A^n)}{n}\right)$ ou n est n-grammes (longueur du texte)

3. Systèmes cryptographiquement sûrs

Nous supposerons que nous étudions un système cryptographique constitué de :
– un ensemble fini M de textes clairs,
– un ensemble fini C de textes chiffrés,
– un ensemble fini K de clés,
– pour chaque clé $k \in K$ une fonction injective de chiffrement e_k de M dans C et une fonction injective de déchiffrement d_k de e_k (M) dans M de telle sorte que $d_k \circ e_k = IdM$. Nous supposerons que ce système est utilisé de la manière suivante : chaque nouveau chiffrement d'un texte clair utilise une nouvelle clé choisie aléatoirement dans l'ensemble des clés conformément à sa loi de probabilité.

Définition 2.4 : Un système cryptographique, du type décrit précédemment (en particulier pour lequel on choisit une nouvelle clé pour chaque nouveau message), est parfaitement sûr si : ∀x

∈M, ∀y∈C, P(x|y) =P(x) < la probabilité d'un texte clair x sachant que le texte chiffré est y est la même que la probabilité de x. Le texte chiffré dans ce cas n'apporte aucune information sur le texte clair.

Théorème 2.3 : Si on suppose que ∀ y∈C on a P(y)> 0 et que le système est parfaitement sûr, alors :

|K|≥|C|≥|M|

.

Théorème 2.4 : Soit un système cryptographique vérifiant : |K|=|C|=|M|, ainsi que P(y)>0 pour tout y ∈C. Il est à sécurité parfaite si et seulement si les deux conditions suivantes sont réalisées :

a) toutes les clés sont équiprobables,

b) pour chaque x ∈M et chaque y ∈C il existe une unique clé k vérifiant ek(x) =y.

Soient M, C et K les variables aléatoires discrètes associées aux choix d'un message m , d'un cryptogramme c et d'une clé k, on peut alors donner une définition de la confidentialité parfaite en termes d'entropie.

Définition 2.5 : On dit qu'un système de chiffrement à clé secrète est à confidentialité parfaite si H(M|C) =H(M). < ceci signifie que la connaissance d'un cryptogramme c n'apporte aucune information sur le texte clair m>

Exemple 2.3

Le one-time-pad vérifie très exactement les conditions du théorème (2.3). C'est un système parfaitement sûr.

le one-time-pad, ou encore chiffrement de Vernam, ou encore masque jetable:

Le cryptosystème de Vernam utilise une clé secrète très longue qui devrait de manière idéale représenter une suite aléatoire de bits où chaque bit est indépendant des autres et a une probabilité ½ d'être 0 et bien entendu une probabilité ½ d'être 1.

Si on a un message m de n bits à chiffrer, on considère les n premiers bits de la clé qui constituent un mot K et on calcule le « ou exclusif bit à bit »entre le message et cette partie de la clé, c'est-à-dire que le texte chiffré s'écrit sous la forme c =m⊕K .

La novelle clé

0	1	1	0	1	1	0	0	0	1	1	1	0	1	0	1	0	1	0	1	1	0	1	0			

0	1	1	1	0	1	0	0

Texte clair

0	0	0	1	1	0	0	0

Texte chiffré c=m XOR k

Ainsi la partie K de la clé sert de masque. Le destinataire qui partage la même clé extrait de la même façon la partie K et récupère alors le texte clair m en calculant m = c⊕K. Les deux interlocuteurs jettent la partie K utilisée et peuvent effectuer une nouvelle transaction en procédant de même avec le reste de la clé.

Ce système réalise évidemment les conditions du théorème de Shannon sur la sécurité parfaite : une nouvelle clé est tirée à chaque chiffrement, la clé est aussi longue que le texte clair, l'ensemble des clés a une probabilité équirépartie.

Problème pratique :

- Construire une clé aussi longue et qui de plus soit une suite aléatoire de bits n'est pas chose facile.
- Il est possible de réaliser approximativement un tel système à partir d'un générateur pseudo-aléatoire et d'un germe. Bien entendu dans ce cas, le germe est la véritable clé secrète du système, et les conditions de Shannon ne sont plus satisfaites.

4. Le paradoxe de la cryptographie à clé publique

En cryptographie à clé publique l'attaquant dispose du cryptogramme et de la valeur de la clé publique k. On s'intéresse donc à la quantité H(M|C, K).

Étant donné qu'un message clair est entièrement déterminé par un cryptogramme c et une clé publique k, un résultat classique sur la fonction d'entropie permet d'affirmer que H(M|C, K) = 0. Ce résultat « surprenant », stipule qu'il n'existe pas de système de chiffrement à clé publique à confidentialité parfaite et que de plus il y a suffisamment d'information dans un cryptogramme c et dans la clé publique k pour déterminer le message clair correspondant. Cependant ce résultat ne nous dit pas comment utiliser cette information, ni même si cette dernière peut être exploitée en temps raisonnable. C'est là tout le paradoxe de la cryptographie à clé publique.

5. Exercices

Exercice 2.1

Soit X la variable aléatoire donnée par

X =

A	Avec probabilité ½
B	Avec probabilité ¼
C	Avec probabilité 1/8
D	Avec probabilité 1/8

Calculer l'entropie de X.

Exercice 2.2

Un dé équitable est lancé en même temps qu'une pièce équitable est lancée. Soit A le nombre sur la surface supérieure du dé et B décrit le résultat du tirage au sort, où B est égal à 1 si le résultat est pile et il est égal à 0 si le résultat est face. Les variables aléatoires X et Y sont données par X = A + B et Y = A - B, respectivement.

- Calculer les entropies H(X) et H(Y), les entropies conditionnelles H(X/Y) et H(Y/X) et l'entropie conjointe H(X,Y).

Chapitre 3

Le chiffrement par flot - STREAM CIPHER

1. Introduction

Dans le Chiffrement Symétrique La clé de chiffrement est la même que la clé de déchiffrement comme dans la figure 3.1

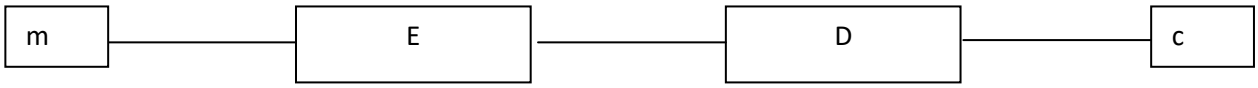


Figure 3.1 : le Chiffrement Symétrique

Dans ce type de chiffrement on trouve deux grandes familles, le chiffrement symétrique par blocs et par flots.

Dans le Chiffrement par blocs les messages sont découpés en blocs par exemple DES en blocs de 64 bits, clés de 56 bits, AES en blocs de 128 bits.

Dans le chiffrement par flots les données sont traitées en flux, on n'a pas besoin de lire le message ni d'avoir sa longueur pour commencer à chiffrer, traitement bit par bit ou octet par octet.

2. Le principe de fonctionnement du chiffrement par flot

L'émetteur et le récepteur du message partagent la connaissance d'une clé secrète K. L'émetteur choisit une valeur aléatoire IV, et utilise un algorithme public (appelé générateur le pseudo-aléatoire cryptographique) lui permettant de dériver une suite S de nombres arbitrairement longue à partir de la clé K et IV. Cette suite ressemble à une suite aléatoire. En particulier, sans la connaissance de la clé, on ne doit pas pouvoir retrouver cette suite, même en en connaissant une partie. L'émetteur utilise cette suite de nombres comme clé de chiffrement, et applique un chiffrement de Vernam sur son message clair.

Il envoie IV et le message chiffré.

Le récepteur connaît K et a reçu IV, il peut donc en déduire S.

Connaissant S, il peut effectuer l'opération correspondant au déchiffrement de Vernam pour retrouver le message clair (La figure 3.2).

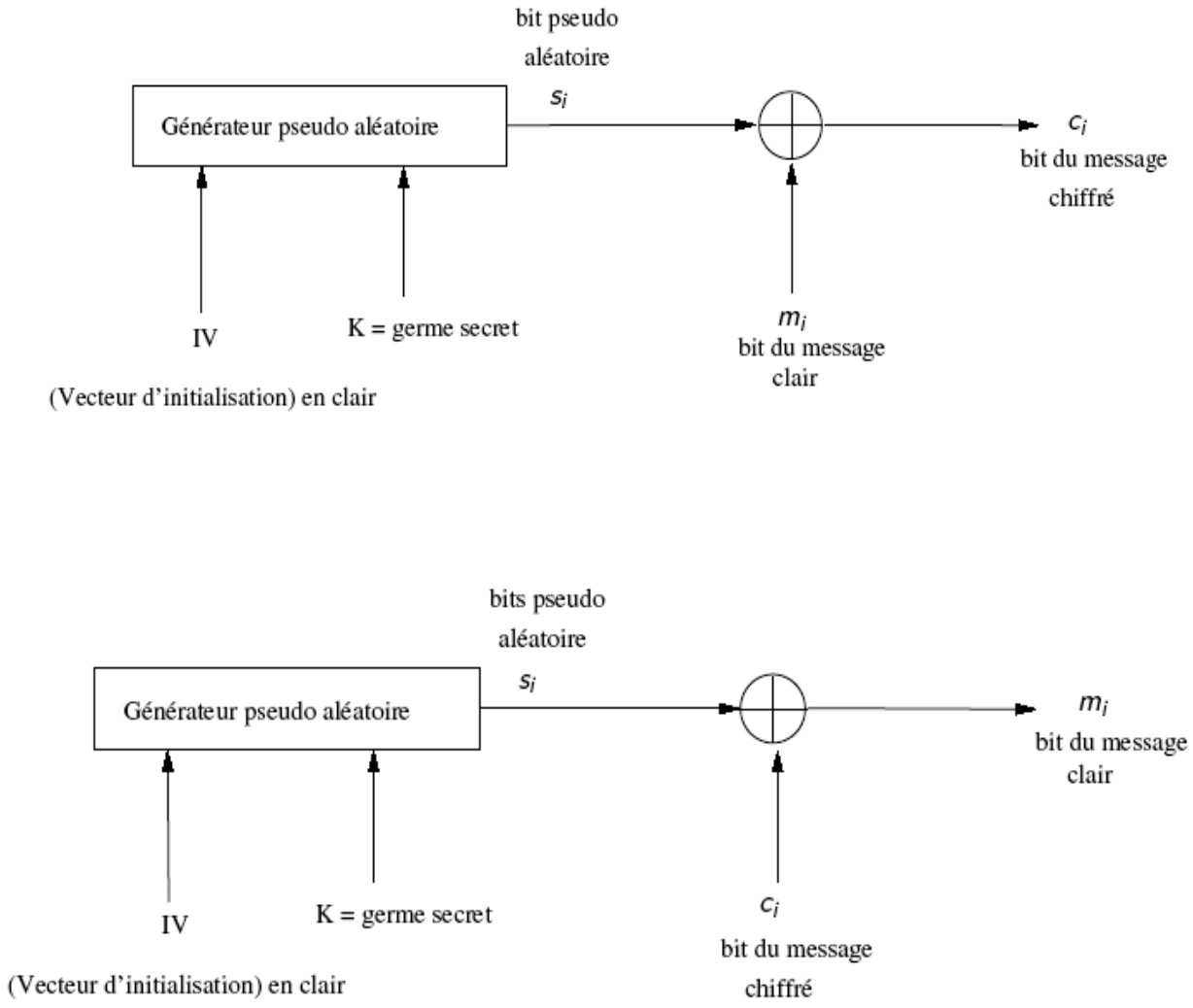


Figure 3.2 : Le chiffrement par flots

Ce type de chiffrement est utilisé essentiellement pour des communications sans fil (GSM, Wifi, Bluetooth,...) à cause du besoin de débit élevé.

3. Générateur aléatoire cryptographique

Un générateur cryptographique utilisable pour la cryptographie doit:

a. Générer des suites de bits satisfaisant les caractéristiques statistiques de suites vraiment aléatoires.

b. Garantir que si un attaquant connait tout ou une partie de la suit chiffrante s0, s1….. si ….. il est difficile, d'un point de vu quantité de calcul, de trouver la clef K ayant servi de germe.

3.1 Générateur pseudo aléatoire GPA

Un générateur pseudo-aléatoire de symboles (GPA) est un automate a nombre fini d'états qui a partir de la donnée d'un nombre fi ni de symboles, que l'on appelle graine ou germe (seed en anglais) produit une suite potentiellement illimité de symboles qui a l'apparence d'une suite aléatoire.

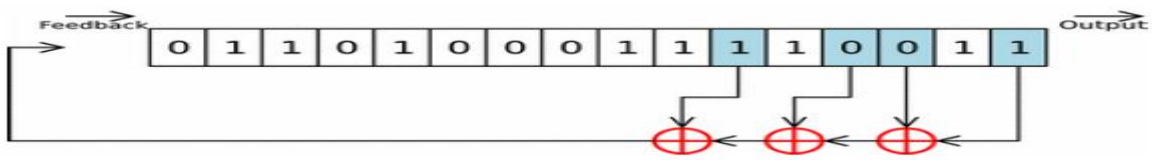
3.2 Registre à décalage (Linear Feedback Shift Registers- LFSR)

un registre à décalage de longueur L est constitué de L cases mémoire R_0, R1,…,R_{l-1} chacune pouvant contenir 0 ou 1(un bit) et ayant une sortie et une entrée, ainsi que d'une horloge contrôlant le mouvement des données.

A chaque top d'horloge :

- Un bit de rétroaction est calculé par combinaison de bits des cases 0 à L-1 ;
- Le contenu de la case 0 sort du registre pour former la séquence de sortie ;
- Le contenu de la case i passe dans la case i-1 pour 1<=i<=L-1 ;
- La case L-1 est remplacée par le bit de rétroaction.

Si la combinaison est un simple XOR on dit que la rétroaction est linéaire (LFSR) , dans le cas d'une rétroaction linéaire donnée par des coefficients binaires a_i, si l'on appelle s_i la séquence de sortie on a alors la relation de récurrence suivante entre les bits de sortie : $s_i=a_1S_{i-l}+a_2S_{i-2}+\ldots+a_LS_{i-L}$ pour i>= L la séquence (s0,s1,…,sl-1) est appelée initialisation, le polynôme $C(x)=1+a_1X+\ldots+a_LX^L$ est appelé polynome de rétroaction linéaire du générateur.



Exemple 3.1

Soit L=3 et $C(x)=1+x+x^3$ avec une initialisation (s_0, s1, s2, s3)= (0, 1, 1)

011-111-110-101-010-100-001-011-111-110-101-010-100-001-011

LSFR de période 7

Le résultat

011101001110100

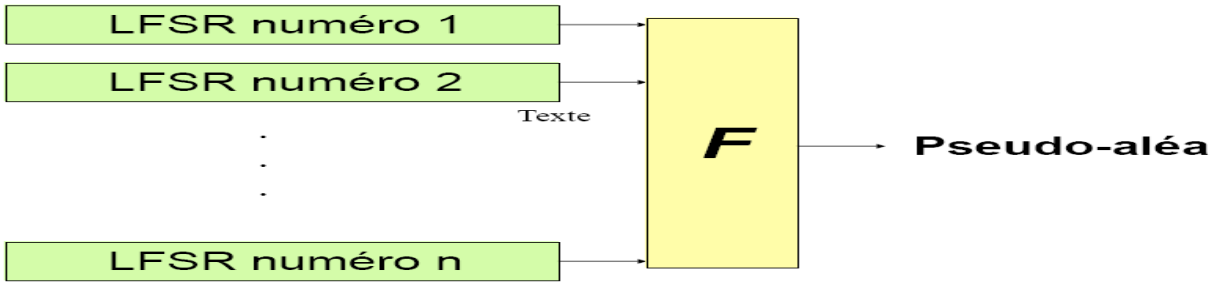
Lemme 3.1

Soit R un registre à rétroaction de longueur L. Si son polynome de rétroaction C(x) est de dgré L, alors la séquence est périodique de période au plus 2^L-1.

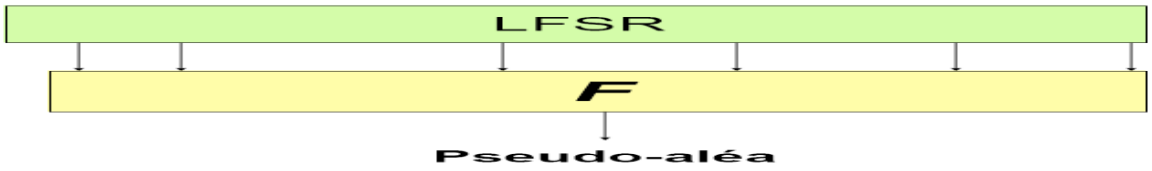
4. Constructions

Pour construire un générateur pseudo-aléatoire Avec des LFSR, on a trois techniques : Combinaison, Filtrage et Contrôle de l'horloge.

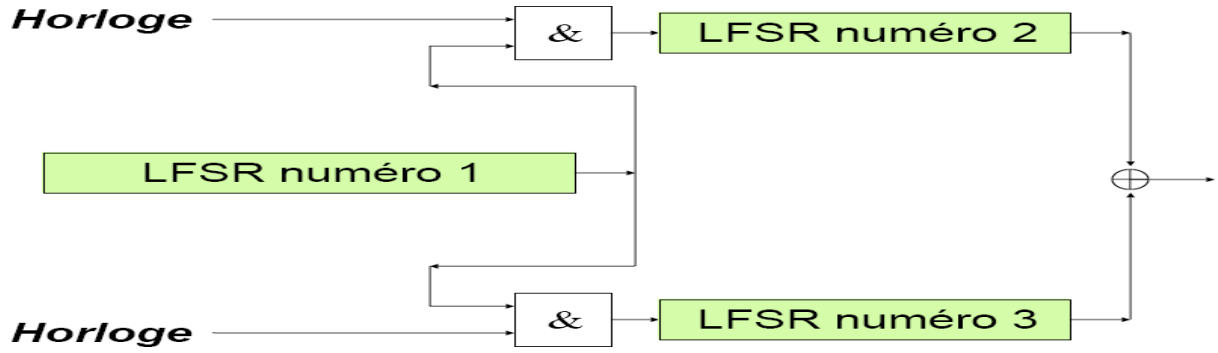
1. Combinaison



2. Filtre



3. Contrôle de l'horloge



5. Les principaux algorithmes de chiffrement à flot utilisés dans les applications

5.1- A5/1(A5/2) : est l'algorithme de chiffrement à flot utilisé pour protéger la confidentialité des communications hertziennes pour les téléphones mobiles dans la norme GSM, c'est-à-dire pour protéger les échanges entre téléphone mobile et station de base.

Le générateur pseudo-aléatoire associé est composé de trois LFSR combinés

Description. Le générateur pseudo-aléatoire de A5/1 est composé de trois LFSR de longueurs respectives 19, 22 et 33. Leurs polynômes caractéristiques sont $X^{19}+X^5+X^2+X+1$, $X^{22}+X+1$ et $X^{23}+X^{15}+X^2+X+1$.

Sécurité :A5/1 est vulnérable aux attaques par compromis temps-mémoire et à certaines attaques par corrélation rapide.

5.2- RC4 : est un algorithme de chiffrement à flot destiné aux applications logicielles. Il a été conçu par R. Rivest en 1987 pour les laboratoires RSA

Il est employé par exemple dans SSL/TLS, protocole permettant d'assurer la confidentialité des transactions Web, dans la norme de chiffrement WEP (Wired Equivalent Privacy) pour les réseaux sans fil, IEEE 802.11b...

Description. RC4 est composé d'une phase d'initialisation, qui consiste à engendrer une permutation des mots de n bits à partir de la clef secrète, Puis, à chaque instant, on modifie le tableau initial en permutant deux de ses éléments, et on produit un mot de n bits de suite chiffrante, qui correspond à un élément du tableau.

Clef secrète.

K, composé de k mots de n bits, K[0], …, K[k−1].

Initialisation.

//addition modulo 2^n

- Pour i de 0 à 2^n-1, S[i] ← i.
- j ← 0
- Pour i de 0 à 2^n-1 faire
  - j ← j + S[j] + K[i mod k]
  - échanger S[i] et S[j]

Génération de la suite chiffrante.

i ← 0, j ← 0

- Répéter
  - i ← i+1
  - j ← j+S[i]
  - échanger S[i] et S[j].

Sécurité : La plupart des attaques connues sur RC4 exploitent des faiblesses de la phase d'initialisation.

Remarque : il existe d'autres algorithmes : F8, MUGI, SNOW 2.0, E0 …

6. Exercices

Exercice 3.1

Soit L=4 et $C(x)=1+x+x^2+x^4$ avec une initialisation $(s_0, s_1, s_2, s_3)= (0, 1, 1,0)$

Générer la suite pseudo-aléatoire par cette LSFR.

Chapitre 4

Le chiffrement à clé secrète –DES (Data Encryption Standard)

1. Introduction

- Le 15 mai 1973 le NBS (*National Bureau of Standards)* a lancé un appel dans le Federal Register (Journal Officiel) pour la création d'un algorithme de chiffrement répondant aux critères suivants :
  - posséder un haut niveau de sécurité lié à une clé de petite taille servant au chiffrement et au déchiffrement
  - être compréhensible
  - ne pas dépendre de la confidentialité de l'algorithme
  - être adaptable et économique
  - être efficace et exportable
- Fin 1974, IBM propose « Lucifer », qui, grâce à la NSA (National Security Agency), est modifié le 23 novembre 1976 pour donner le **DES** (*Data Encryption Standard*).
- Le DES a finalement été approuvé en 1978 par le NBS. Connus sous la dénomination *DEA* (*Data Encryption Algorithm*).

2. Chiffrement de Feistel

Chiffrement appliquant itérativement la transformation suivante à un mot coupé en deux (dont les deux moitiés sont respectivement appelées L_i et R_i) :

$(L_0,R_0) \rightarrow (L_k,R_k)$ Avec : $L_i = R_{i-1}$ et $R_i = L_{i-1}$ XOR $\mathbf{f}(R_{i-1},K_i)$ (K_i dérivée de K)

3. Principe du DES (Data Encryption Standard)

DES utilise un chiffrement Feistel avec :

• Groupement du texte en blocs de 64 bits

• Clé 56 bits (notée K)

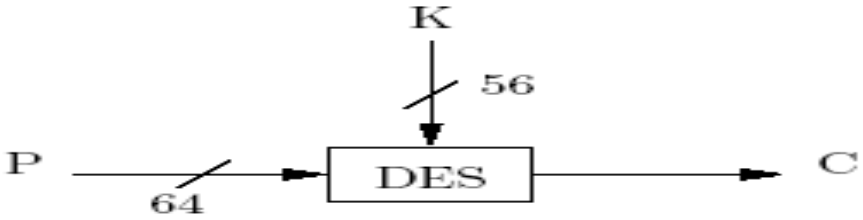


Figure 4.1 : Chiffrement DES

Ou P est le texte d'origine et C est le mot chiffré.

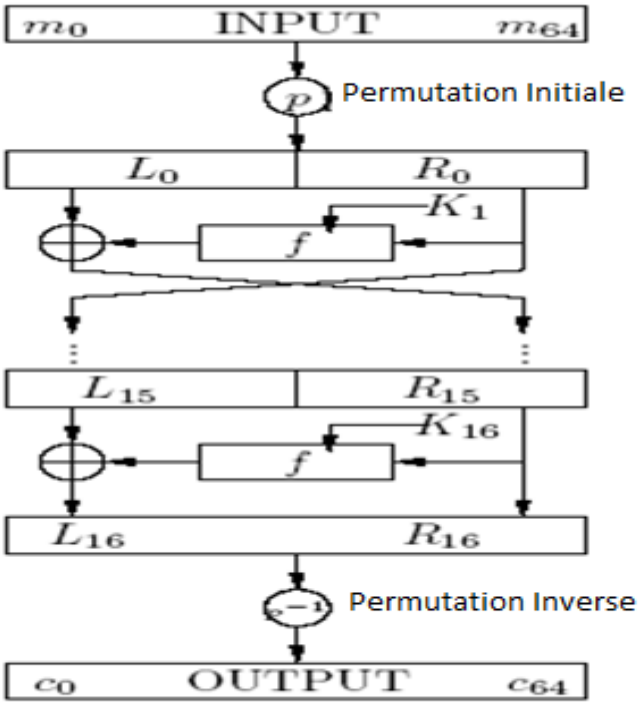


Figure 4.2 : Schéma de chiffrement DES

- Après la permutation initiale:
  - Le message est séparé en deux moitiés de 32 bits L_0 et R_0.
  - A chaque itération de la procédure on détermine L_i et R_i en fonction de L_{i-1} et R_{i-1} obtenus précédemment. en appliquant le chiffrement de Feistel.
- **La fonction f**

  La fonction f est un chiffrement de ***Feistel*** du DES :

$$f(R_{i-1}, K_i) = P(S(E(R_{i-1}) \textbf{ XOR } K_i))$$

  - E: Fonction d'expansion, Boite S, Permutation P.(Annexe 1)

La fonction f

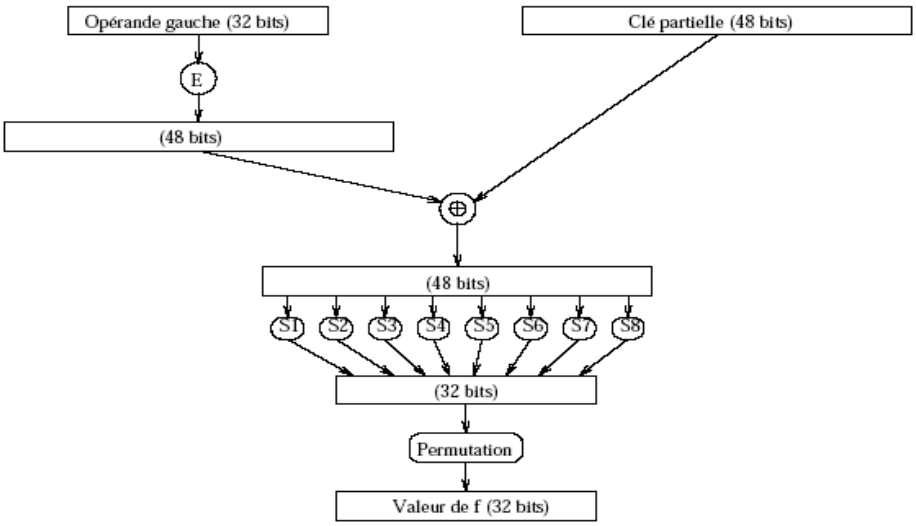


Figure 4.3 : La fonction f

a) Fonction d'expansion E : est une fonction qui duplique des bits de manière à faire passer d'un mot de 32 bits `a un mot de 48 bits. (Annexe)

b) Boite S (complexité du DES) : on a 8 boites S : 6 bits ---> 4 bits, fonctionnent de la manière suivante : Chaque mot de 6 bits b1b2b3b4b5b6 est décomposé en deux parties : b1b6 : partie jouant le rôle d'instruction b2b3b4b5 : partie jouant le rôle de donnée

La boite S_5 :

$$\begin{pmatrix} 2 & 12 & 4 & 1 & 7 & 10 & 11 & 6 & 8 & 5 & 3 & 15 & 13 & 0 & 14 & 9 \\ 14 & 11 & 2 & 12 & 4 & 7 & 13 & 1 & 5 & 0 & 15 & 10 & 3 & 9 & 8 & 6 \\ 4 & 2 & 1 & 11 & 10 & 13 & 7 & 8 & 15 & 9 & 12 & 5 & 6 & 3 & 0 & 14 \\ 11 & 8 & 12 & 7 & 1 & 14 & 2 & 13 & 6 & 15 & 0 & 9 & 10 & 4 & 5 & 3 \end{pmatrix}$$

Exemple 4.1

Ainsi S5(000111) :

• b1b6 = 01, on lit donc la deuxième ligne

• b2b3b4b5 = 0011 = 3, on regarde donc la 4ème colonne et le résultat est 12 = 1100

Les différentes boites S sont précisées dans l'annexe technique fournie sous forme de document papier.

c) Permutation P : est une permutation fixée permettant un couplage des sorties des S-boites.

- **La diversification des clés**

  - Le principe:

    - On applique d'abord une permutation PC1 à K
    - Puis à chacune des 16 étapes :
      - Chaque moitié de la chaine de 56 bits obtenue subit une rotation à gauche d'un cran aux étapes 1,2,9 et 16 et de deux crans aux autres étapes.
      - À chacune de ces étapes on obtient une clé partielle de 48 bits en appliquant la règle d'extraction PC2.
      - Les 56 bits de la clé sont numérotés de 1 à 64 en évitant les multiples de 8 puisque en pratique ces positions sont utilisées pour insérer des bits de parité (parité impaire en générale).

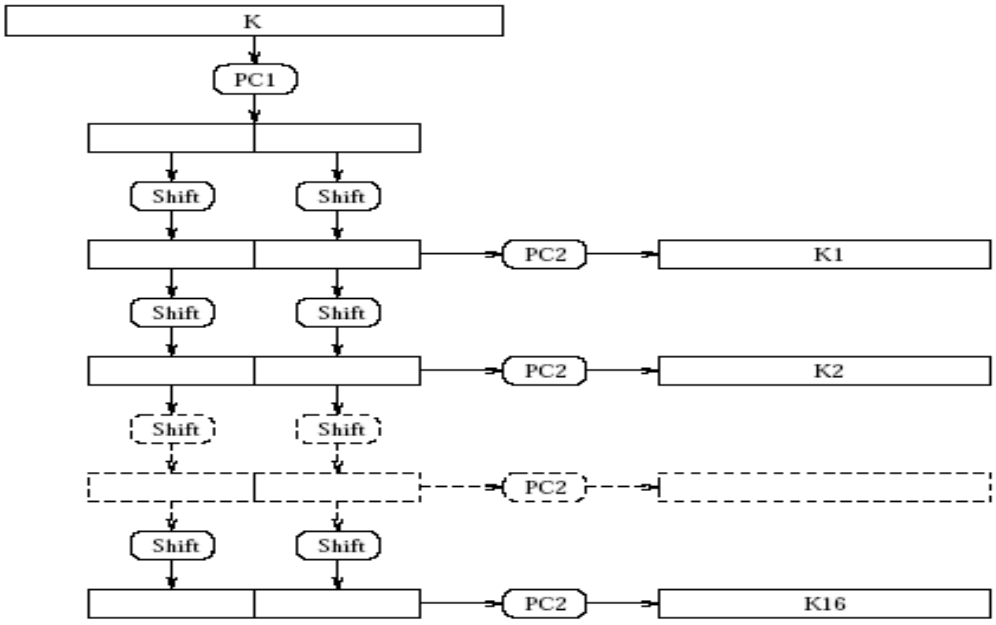


Figure 4.4 : La diversification des clés DES

Exemple 4.2

Soit

P=0123456789ABCDEF. Et K=133457799BBCDFF1.

Développement de P

0	0	0	0	0	0	0	1
0	0	1	0	0	0	1	1
0	1	0	0	0	1	0	1
0	1	1	0	0	1	1	1
1	0	0	0	1	0	0	1
1	0	1	0	1	0	1	1
1	1	0	0	1	1	0	1
1	1	1	0	1	1	1	1

Application de PI :

1	1	0	0	1	1	0	0
0	0	0	0	0	0	0	0
1	1	0	0	1	1	0	0
1	1	1	1	1	1	1	1
1	1	1	1	0	0	0	0
1	0	1	0	1	0	1	0
1	1	1	1	0	0	0	0
1	0	1	0	1	0	1	0

- On obtient L_0=11001100000000001100110011111111 et R_0=11110000101010101111000010101010

- **Développement binaire de K : 133457799BBCDFF1.**

0	0	0	1	0	0	1	1
0	0	1	1	0	1	0	0
0	1	0	1	0	1	1	1
0	1	1	1	1	0	0	1
1	0	0	1	1	0	1	1
1	0	1	1	1	1	0	0
1	1	0	1	1	1	1	1
1	1	1	1	0	0	0	1

Calcul de K1

Après application de PC1:

G_0=11110000110011001010101011111. D_0=01010101011001100111100011111.

Après le décalage circulaire à gauche:

G_1=11100001100110010101010111111. D_1=10101010110011001111000111110.

On obtient : K1=000110110000001011101111111111000111000001110010.

Puis on calcul $E(R_0)$ XOR K_1 =011000010001011110111010100001100110010100100111.

- Puis on calcule $f(R_0,K_1)$

- Et finalement on obtient R_1.

N.B : DES $^{-1}$

$$L^{i-1} = R^i \oplus f(L^i, K^i)$$
$$R^{i-1} = L^i.$$

4. Cryptanalyse de DES

4.1 La cryptanalyse différentielle

A été introduite par Biham et Shamir, c'est une attaque à messages clairs choisis. Permet de casser des versions restreintes de DES, dans lesquels on diminue le nombre de tours de la boucle principale. Permet de casser facilement un DES à 8 ou à 10 tours. Le DES complet à 16 tours est resté hors de portée de cette attaque.

4.2 La cryptanalyse linéaire

A été introduite par H.Gilbert et par M. Matsui. C'est une attaque à messages clairs connus. Elle n'est aussi utilisable que pour un DES restreint. Le DES réel n'est pas menacé par cette attaque.

4.3 La recherche exhaustive

L'accroissement de la puissance des ordinateurs, de leur nombre et des facilités de communication entre eux ont rendu envisageable une attaque de DES par recherche exhaustive de la clé (brut force attack).

Pour démontrer que la taille des clés (56 bits) devenait insuffisante, les laboratoires RSA ont lancé en janvier 97 un défi consistant à décrypter par recherche exhaustive un message chiffré par DES.

Une équipe coordonnant les milliers d'ordinateurs du monde entier a abouti à la découverte de la bonne clé le 17 juin 97 après avoir exploré un quart de l'espace des clés.

5. Triple-DES

Consiste à composer deux chiffrements DES de même clé séparés par un déchiffrement DES avec une autre clé :

- $\text{Triple-DES}_{k1,k2} = \text{DES}_{k1} \text{o } \text{DES}_{k2}^{-1} \text{ o} \text{DES}_{k1}$
- $\text{Triple-DES}_{k1,k2}^{-1} = \text{DES}_{k1}^{-1} \text{ o } \text{DES}_{k2} \text{ o} \text{DES}_{k1}^{-1}$
- Une clé est donc composée de deux clés et fait 112 bits ce qui met Triple-DES largement hors de portée d'une attaque exhaustive.

6. Rijndael – AES (*Advanced Encryption Standard*)

- Bien que la sécurité de Triple-DES semble élevée, il est un peu lent.
  Le besoin s'est fait sentir de définir un successeur à DES, entièrement nouveau, répondant à des critères modernes (par exemple en taille de clé), et susceptible de subvenir aux besoins

cryptographiques pendant de nombreuses années.

Un appel d'offres à été lancé par le NIST (National Institute of Standards and Technology) pour ce projet. Une quinzaine de candidats se sont mis sur les rangs. Cinq ont été retenus au cours de l'été 1999 (Mars, RC6, Rijndael, Serpent, TwoFish).

C'est finalement Rijndael qui a été retenu et qui devrait remplacer DES après une période de normalisation.

- C'est une création belge due à Joan Daemen et Vincent Rijmen. Comme l'imposait le cahier de charges de l'AES, il est spécifié dans plusieurs variantes utilisant des clés de tailles différentes : 128, 192 et 256 bits. De plus, il peut chiffrer des blocs de 128, 192 ou 256 bits.

- Les choix de la taille de la clé et de la taille des blocs sont indépendants, il y a donc en tout 9 variantes.

6.1 Le principe d'AES

- il faut disposer les octets des blocs selon une matrice 4 x N_b et les octets de la clé selon une matrice 4 x N_k où N_b et N_k valent 4, 6 ou 8 selon les longueurs respectives des blocs et de la clé.

- Le chiffrement par Rijndael est une succession de tours semblables.

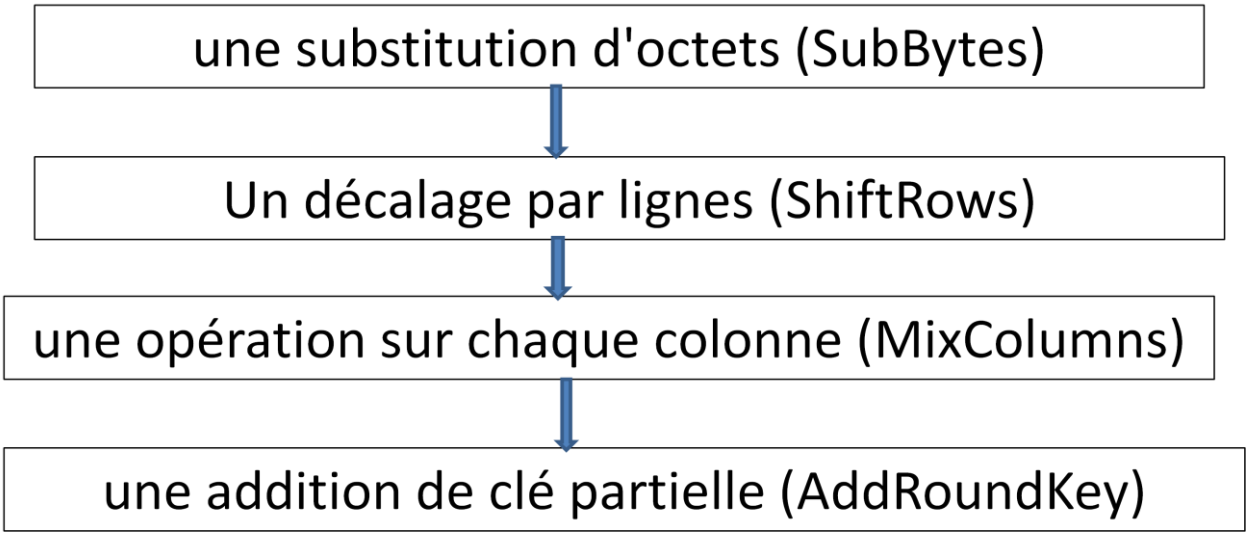
- Le nombre N_r de tours dépend de N_b et N_k :

  - $N_r = 10$ si $N_b = N_k = 4$.

  - $N_r = 12$ si $max(N_b, N_k) = 6$.

  - et $N_r = 14$ si $\max(N_b, N_k) = 8$.

Chaque tour: 4 phases



Remarque : pour le dernier tour l'opération sur chaque colonne est supprimée. Le premier tour est précédé d'une addition de clé partielle.

A partir de la clef secrète, K, on génère des sous clefs d'étage par l'algorithme de diversification de la clef, ExpandKey[i]. On obtient les clefs d'étage: K1,...,K_{Nr}.

7. Modes opératoires

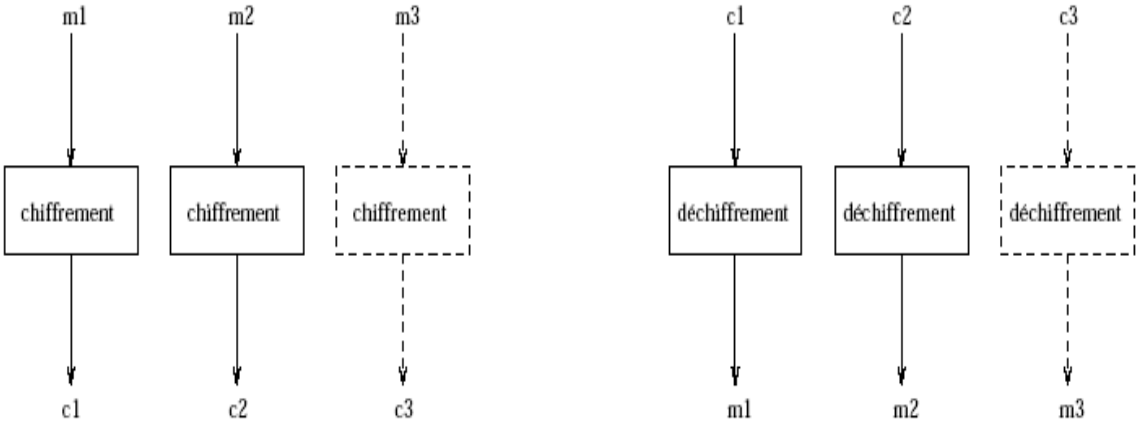
quatre modes de chiffrement par blocs sont possibles: ECB, CBC, CFB et OFB.

7.1 ***ECB (Electronic Code-Book)***

Consiste à chiffrer successivement chaque bloc d'un message avec la même clé.

Inconvénients: un attaquant actif peut permuter des blocs chiffrés et/ou en supprimer de telle façon que le clair modifié ait encore un sens, différent du sens initial.

Avantage : de laisser la liberté de chiffrer/déchiffrer les blocs dans n'importe quel ordre et, en cas de perte d'un bloc chiffré, de ne pas bloquer le déchiffrement des blocs restants.

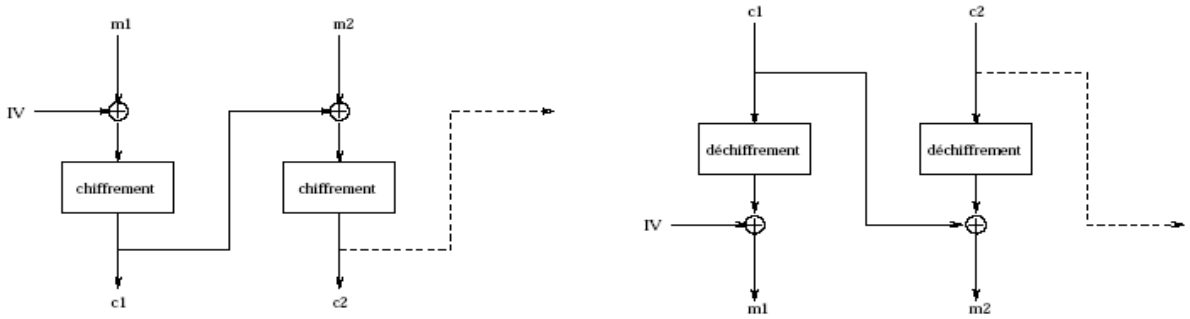


7.2 CBC *(Cipher Bloc Chaining)*

Consiste à, avant le chiffrement d'un bloc, le masquer par le résultat du chiffrement du bloc précédent au moyen de l'opération XOR. Le premier bloc clair est lui aussi masqué, par une valeur habituellement notée IV *(Initial* Value. La valeur initiale IV n'a pas besoin d'être secrète, et elle est en général transmise en clair avant le message chiffré. Noter que si le destinataire reçoit un bloc chiffré avec des bits erronés, cela affecte le déchiffrement de ce bloc et du suivant mais pas des autres.

Le chiffrement :$C1=E_K(m1+IV)$ i=1 et $Ci=E_K(ci-1+mi)$ i>=2

Le déchiffrement : $M1=D_K(c1)+IV$ i=1 et $Mi=D_K(ci)+ci-1$



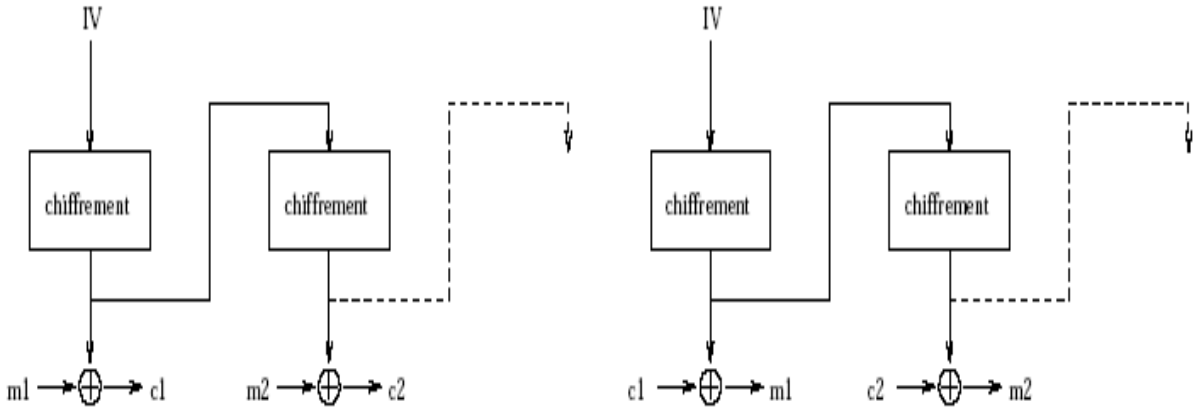
7.3 OFB *(Output FeedBack)*

Pour déchiffrer un message en mode ECB ou CBC, il faut avoir reçu chaque bloc chiffré complet avant de pouvoir obtenir n'importe quel bit du bloc clair correspondant. Cependant, certaines applications nécessitent de pouvoir déchiffrer le flux des données au fur et à mesure de leur

arrivée, même si la transmission se fait par petits morceaux. Cela est possible en utilisant le mode OFB (Output FeedBack).

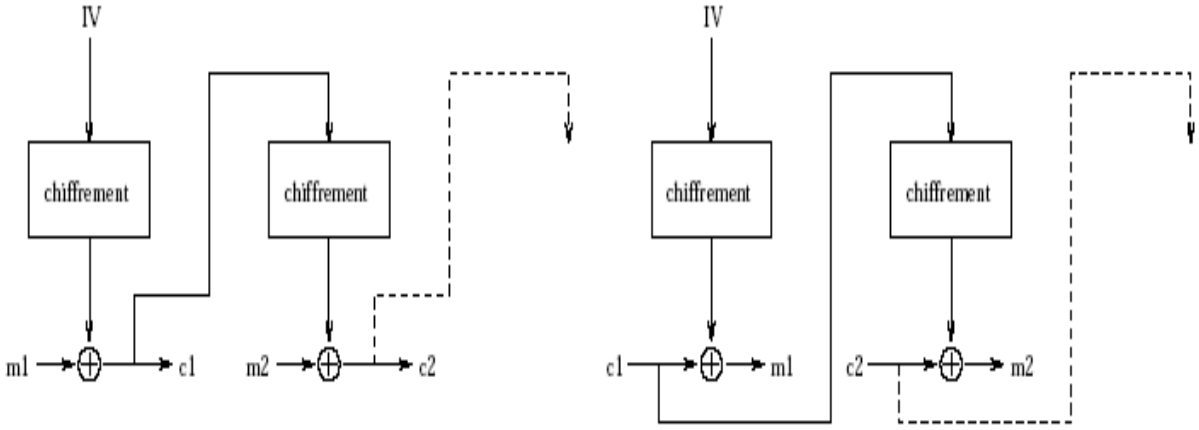
Principe : consiste à itérer la fonction de chiffrement sur une valeur initiale IV et à utiliser le flot de bits pseudo-aléatoires obtenus pour masquer les bits clairs à l'aide de l'opération XOR.

- A noter que l'opération de déchiffrement est identique à celle de chiffrement, et utilise la fonction de chiffrement de blocs et non celle de déchiffrement.
- inconvénient : un attaquant actif peut modifier des bits du message clair en modifiant les bits correspondants du chiffré (cela peut être aussi un avantage, si un bit du message chiffré est modifié par erreur au cours de la transmission, seul le bit correspondant du message clair sera affecté).



7.4 *CFB* (Cipher FeedBack)

est proche du mode OFB mais le flot de bits pseudo-aléatoires dépend cette fois des blocs chiffrés.
Un attaquant peut encore modifier un bit du clair en modifiant le bit correspondant du chiffré, mais alors le bloc suivant une fois déchiffré sera complètement différent du bloc original.



Le chiffrement : C1=M1+E(IV) pour i=1 et Ci=Mi+E(Ci-1) i>=2

Le déchiffrement : m1=c1+E(IV) i=1 et mi=ci+E(Ci-1) i>=2.

8. Exercices

Exercice 4.1

Le mode CBC (Cipher Block Chaining)

1- Donnez les algorithmes de chiffrement et déchiffrement de ce mode.

2- Soit VI = 1010 vecteur d'initialisation. Soient M = 010100011011010 message en clair et une clé K = (1->2, 2->3, 3-> 4, 4->1) par permutation. Chiffrez le message M.

Déchiffrez le message C= 0010 0110 0100 1101.

Exercice 4.2

Soit P=0123456789ABCDEF. Et K=133457799BBCDFF1. En appliquant DES Donner R1,L1 et K1.

Chapitre 5

Cryptographie à clé publique (Asymétrique)

1. Introduction

Dans le précédent chapitre on a étudié la cryptographie à clé secrète ou symétrique qui nécessite une clé privée connue seulement de l'envoyeur et de son correspondant. Dans ce type de crypto système on a besoin d'un canal parallèle pour la transmission de la clé en plus avec l'amélioration rapide de la capacité de calcul, ces contrainte rendent ce type de cryptosystème vulnérable. En 1976, Diffie, Hellman et Merkle propose d'élaborer un nouveau type de cryptosystème : la cryptographie à clé publique ou asymétrique.

Principe (Figure 5.1) :

Alice génère deux clés : la clé publique qu'elle envoie à Bob et la clé privée qu'elle conserve précieusement sans la divulguer à quiconque.

Bob chiffre le message avec la clé publique d'Alice et envoie le texte chiffré, Alice déchiffre le message grâce à sa clé privée.

- Fondée sur l'existence de fonctions a sens unique.
- Une **fonction à sens unique** est une fonction qui peut être aisément calculée, mais qui est difficile à inverser.
- Certaines **fonctions à sens unique** sont appelées fonctions à brèche secrète en raison d'une « brèche secrète » qui permet à quelqu'un la connaissant de revenir facilement en arrière.
- Il est simple d'appliquer cette fonction a un message, mais extrêmement difficile de retrouver ce message a partir du moment ou on l'a transforme.

Déroulement :

- Bob souhaite pouvoir recevoir des messages chiffrés de n'importe qui.
- Il génère une valeur (clef publique) a partir d'une fonction a sens unique.
- Il diffuse la clef publique, mais garde secrète l'information permettant d'inverser cette fonction (clef secrète).

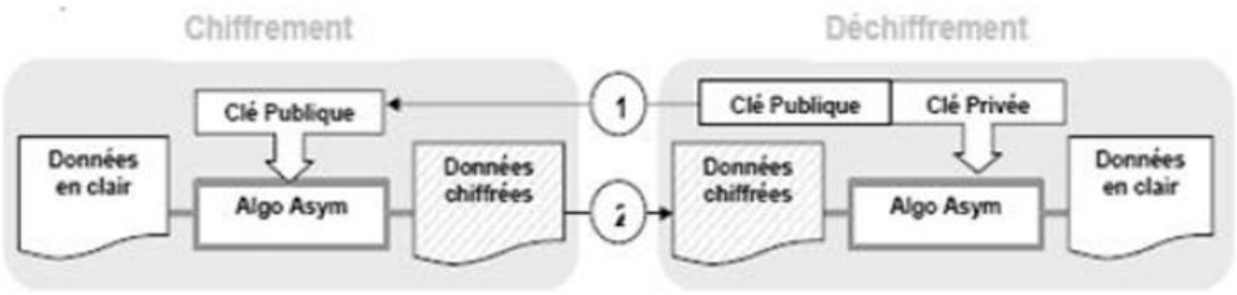


Figure 5.1 : Principe de la cryptographie à clé publique(Asymétrique)

Plusieurs systèmes ont été proposés. Les plus connus sont :

- Le cryptosystème RSA
- DLP & Diffie-Hellman
- Le cryptosystème El Gamal

2. Le cryptosystème RSA

Ce cryptosystème est proposé par Ronald Rivest(R), Adi Shamir(S) et Leonard Adleman (A) qui ont eu l'idée d'utiliser les anneaux Z/nZ et le théorème de Euler-Fermat publié en 1761 par le mathématicien suisse Leonhard Euler pour obtenir des fonctions trappes, ou fonctions a sens unique a brèche secrète.

Théorème 5.1 (Euler-Fermat)

Pour tout entier $n > 0$ et tout entier a premier avec n,

$$a^{\varphi(n)} = 1 \bmod n$$

ou φ est la fonction indicatrice d'Euler et *mod* désigne la congruence sur les entiers.

C'est a l'heure actuelle le système a clé publique le plus utilise (Netscape, les cartes bancaires, de nombreux sites web commerciaux).

RSA repose sur le calcul dans les groupes Z/nZ plus précisément, sur l'exponentiation modulaire.

Principe :

Alice veut envoyer un message à Bob :

- M un entier représentant le message.
- Bob choisit p et q deux nombres premiers et on note n leur produit.
- Bob choisit e un entier premier avec p-1 et q-1.
- On a $\phi(n)=(p-1)(q-1)$. Donc e est premier avec $\phi(n)$ et on obtient (via Bézout) qu'il est inversible modulo $\phi(n)$, i.e. il existe un entier d tel que $ed=1 \bmod(\phi(n))$.

- Le message chiffré sera alors présenté par $C=M^e \text{ Mod } n$.
- Pour déchiffrer C, on calcule $C^d \text{ mod } n$. (d inverse de e mod ϕ(n))

  - (n,e) est appelé clé publique
  - (n,d) est appelé clé privée.

Exemple 5.1

Alice		Bob
M		
		choisit p et q e premier avec $p-1$ et $q-1$
		calcule $n = p \times q$ d tel que $ed \equiv 1 \pmod{\varphi(n)}$
	⟵	envoie (n, e) à Alice
calcule $C = M^e \pmod{n}$ et l'envoie à Bob	⟶	
		calcule $C^d \pmod{n}$ et en déduit M

Soit : p= 47 et q= 59

- on calcule : n=p.q=2773 et ϕ(n)= (p-1)(q-1)=46.58=2668
- on choisit e premier avec ϕ(n) Ex e=17.
- on calcule avec l'algorithme d'Euclide étendu, d tel que d.e=1 mod ϕ(n), soit d=157 ?

donc :La clé publique (e,n)= (17,2773) et la clé privée d=157

- chiffrement du message M =01000010=66
- $C=M^e \text{ Mod } n = (66)^{17} \text{ Mod } 2773 = 872$
- Déchiffrement de C : $C^d \text{ Mod } n= 872^{157} \text{ Mod } 2773 = 66$.

Pour trouver d tel que d.e=1 mod ϕ(n)

- On cherche d tel que : d.e=1mod(p-1).(q-1)
- On a : uxa+vxb (u et v coefficients de Bézout):
- Soit : a=(p-1).(q-1)=(47-1).(59-1)=2668; b=e=17.

Algo. D'Euclide étendu

2668=1x2668+0x17

17=0x2668+1x17

16=2668-156x17

1=17-1x16=17-1x(2668-156x17)=-2668+157x17

Donc :u=-1 et v=157

D'où: 17^{-1} =157mod2668 Alors d=157.

Exponentiation modulaire

- $C=M^e \bmod n=66^{17} \bmod 2773$

1. on a 17=10001

2.i=0: $66^{20} \bmod 2773 = 66 \bmod 2773$

i=1 : $66^{21} \bmod 2773 = 66^2 \bmod 2773 = 1583 \bmod 2773$

i=2: $66^{22} \bmod 2773 = 1583^2 \bmod 2773 = 1870 \bmod 2773$

i=3: $66^{23} \bmod 2773 = 1870^2 \bmod 2773 = 147 \bmod 2773$

i=4: $66^{24} \bmod 2773 = 147^2 \bmod 2773 = 2198 \bmod 2773$

3. On en déduit : $66^{17} \bmod 2773 = 66^{20} . 66^{24} \bmod 2773 = 66.2198 \bmod 2773 = 872 \bmod 2773$. donc C=872.

2.1 La sécurité de RSA

La sécurité RSA dépend de la difficulté de factoriser de grands entiers. Par exemple, en utilisant une vaste banque d'ordinateurs et les meilleurs méthodes actuellement connues, on a réussit à factoriser en 5 mois le nombre :

3107418240490043721350750035888567930037346022842727545720161948823206440518081504556346829671723286782437916272838033415471073108501919548529007337724822783525742386454014691736602477652346609

Comme produit des deux nombres premiers

163473364580925384844313388386509085984178367003309231218111085238933310010450815121211816751157951907128166482211312685157393541397547189678996851549366663853908802710380210449895719126146557 1

Pour l'instant, si les nombres p et q, servant à construire la clé de RSA, sont choisit avec plus de 150 chiffres, la factorisation de n = p · q semble être complètement hors de portée des ordinateurs modernes.

3. DLP & Diffie-Hellman

Autre problème difficile : Discret Logarithme Problem

- **Definition** (Logarithme discret)
  Soit $G =< g >= \{g^i\}_{0 \leq i < n}$ un groupe monogène fini d'ordre n. Soit $h \in G$. Alors le logarithme discret de h en base g, noté $log_g h$, est l'unique entier x tel que $h = g^x (0 \leq x < n)$.
- DLP consiste alors à résoudre le problème suivant :
  Etant donné G, g, h, trouver $x = log_g h$.
- Exemple : $p = 97$ et $G = \mathbb{Z}/97\mathbb{Z} = 1, 2, ..., 96 = \{5^i\}_{0 \leq i < 96}$;
  $5^{32} = 35 \mod 97 \Rightarrow log_5 35 = 32$ dans $\mathbb{Z}/97\mathbb{Z}$.

Alice et Bob veulent partager une clef secrète K. On suppose que les données $G, n = |G|$ et g sont publiques.

- Alice choisit un entier $1 \leq a \leq n - 1$ au hasard.
- Alice calcule $A = g^a$ et l'envoie à Bob.
- Bob choisit un entier $1 \leq b \leq n - 1$ au hasard.
- Bob calcule $B = g^b$ et l'envoie à Alice.
- Alice est en mesure de calculer B^a et Bob de calculer A^b. La clef commune est donc

$$K = g^{ab} = A^b = B^a.$$

Exemple 5.2

Alice		Bob
génère a $A = g^a \mod p$		génère b $B = g^b \mod p$
	$A \longrightarrow$ $\longleftarrow B$	
(dispose de $[a, A, B, p]$) Clé secrète : $K = B^a \mod p$		(dispose de$[b, A, B, p]$ Clé secrète : $K = A^b \mod p$

Exemple 5.3

1. Alice et Bob choisissent un nombre premier p et une base g. Dans notre exemple, $p = 23$ et $g = 3$
2. Alice choisit un nombre secret $a = 6$
3. Elle envoie à Bob la valeur $g^a \mod p = 3^6 \mod 23 = 16$
4. Bob choisit à son tour un nombre secret $b = 15$
5. Bob envoie à Alice la valeur $g^b \mod p = 3^{15} \mod 23 = 12$
6. Alice peut maintenant calculer la clé secrète :

$$(g^b \mod p)^a \mod p = 12^6 \mod 23 = 9$$

7. Bob fait de même et obtient la même clé qu'Alice :

$$(g^a \mod p)^b \mod p = 16^{15} \mod 23 = 9$$

4. Le cryptosystème d' El Gamal

Données publiques pré-requise :

- $(G, .) =< g >$ un groupe cyclique d'ordre n

Génération des clefs

- Bob choisit $a \in [1, n-1]$ et calcule $A = g^a$ dans G.
- Clef publique : (G, g, n, A).
- Clef secrète : a.

Le chiffrement :

- Alice souhaite envoyer le message M à Bob
- Alice récupère la clé publique (G, g, n, A) de Bob.

- Alice choisit au hasard k [1, n-1]
- Le message chiffré qu'Alice envoie à Bob est C = (y_1, y_2) avec : $y_1 = g^k$ et $y_2 = M.A^k$

Le déchiffrement :

- Bob reçoit le message chiffré C = (y_1, y_2)
- Il lui suffit alors de calculer $M=Y_2/y_1^a$.
- En effet : $M.A^k / g^{k\,a} = M. g^{k\,a} / g^{k\,a} = M$

Exemple 5.4

Prenons $p = 2357$ et $g = 2$ qui est d'ordre maximal 2356.

- Bob choisit $a = 1751$, et calcule $g^a = 2^{1751} \mod 2357 = 1185$.
- La clé publique est $p = 2357, g = 2, g^a = 1185$.
- Soit le message $m = 2035$
- Alice choisit $k = 1520$,
- calcule $y_1 = 2^{1520} \mod 2357 = 1430$, et $w = 1185^{1520} \mod 2357 = 2084$, puis $y_2 = wm \mod p = 697$.
- Alice envoie $y_1 = 1430, y_2 = 697$.
- Bob calcule $y_1^a = 1430^{1751} = 2084 \mod p$, puis $y_2/y_1^a = 697/2084 = 2035 \mod p$.

4.1 La sécurité d'El Gamal

Casser l'algorithme El Gamal est dans la plupart des cas au moins aussi difficile que de calculer le logarithme discret. Cependant, il est possible qu'il existe des moyens de casser l'algorithme sans résoudre le problème du logarithme discret.

5. Exercices

Exercice 5.1

Appliquer l'algorithme d'Euclide étendu pour le couple (a,b)=(17,50).

Exercice 5.2

On considère les nombres premiers p = 11, q = 23.

1. Calculer $\Phi(n)$.
2. Calculer la clé publique et privée :

Indication : revient à calculer n, e, d.

3. Chiffrer le message x = 165.

4. Déchiffrer le message 110.

Exercice 5.3

1. Supposons qu'Alice et Bob partagent p = 233 et g = 45.
2. si Alice choisit a = 11 et Bob b = 20, alors :
3. Quelle est leur clef secrète commune ?

Chapitre 6

Fonction de hachage et signatures électroniques

1. Fonction de hachage

- **Définition (Fonction de Hachage) :**
  Une fonction de hachage H est une application facilement calculable qui transforme une chaîne binaire de taille quelconque t en une chaîne binaire de taille fixe n, appelée empreinte de hachage.
- On parle de *collision* entre x et x' lorsque

$$x \neq x' \text{ et } H(x) = H(x')$$

- Si y est tel que $y = H(x)$, alors x est appelé préimage de y

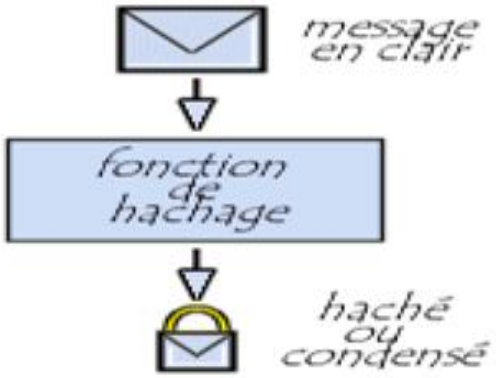


Figure 6.1 : Les fonctions de hachage

Exemple 6.1

Entrée	Fonction de Hachage	Empreinte
Renard	Fonction de Hachage	DFCD3454
Le renard court sur la glace	Fonction de Hachage	52ED879E
Le renard marche sur la glace	Fonction de Hachage	46042841

Définition 6.1 Une fonction de hachage h est à collisions faibles difficiles si étant donnée un message x, il est calculatoirement difficile d'obtenir un message $x' \neq x$ tel que $h(x) = h(x')$.

Définition 6.2 Une fonction de hachage h est à collisions fortes difficiles s'il est calculatoirement difficile d'obtenir deux messages différents x et x' tels que $h(x) = h(x')$.

Propriétés :

- Compression (réduire la longueur du texte d'un message) et facilité de calcul.
- Résistance à la préimage : étant donné y, il est difficile de trouver x tel que y=H(x).
- Résistance à la seconde préimage : étant donné x, il est difficile de trouver x' != x tel que H(x)=H(x').

1.1 Attaques des anniversaires

Une fonction de hachage doit aussi résister aux attaques des anniversaires. Pour comprendre cette attaque on commence par :
Le paradoxe des anniversaires : Dans une classe, quelle est la probabilité pour que 2 élèves fêtent leurs anniversaires le même jour? Avec 365 jours par an, une trentaine d'élèves dans la classe, on se dit qu'elle doit être faible...
On va calculer la probabilité pour que, dans un groupe de k élèves, ces élèves aient tous un jour d'anniversaire différent.

Pour deux élèves : p_2= 364/365=1-1/365 car le premier peut avoir son anniversaire n'importe quand.
Pour trois élèves : p_3= (364/365)(363/365)=(1- 1/365)(1- 2/365) car le premier peut avoir son anniversaire n'importe quand,
Pour k élèves on a p_k=(1- 1/365)(1- 2/365)…. (1- (k-1)/365), donc la probabilité pour que 2 élèves fêtent leurs anniversaires le même jour = 1- p_k.

Si k=23, on an une probabilité de 1/2 pour que 2 personnes aient leur anniversaire le même jour, contrairement à ce que l'intuition laisse présumer. A partir de 50 personnes, il n'y a que 3% de chances que tous les anniversaires diffèrent.

L'attaque des anniversaires utilise ce paradoxe, dans ce cas le nombre des éléves (k) devient le nombre de haches(résumés) résultat du hachage. Le nombre de jours dans l'année(365) devient le nombre des haches(n). La probabilité de non-collision est donc :

$$\left(1-\frac{1}{n}\right)\left(1-\frac{2}{n}\right)\ldots\left(1-\frac{k-1}{n}\right)=\prod_{i=1}^{k-1}(1-\frac{i}{n})$$

on a $e^{-x} \approx 1 - x$ si x petit donc

$$\prod_{i=1}^{k-1}(1-\frac{i}{n}) \approx \prod_{i=1}^{k-1} e^{-1/n} = e^{-\frac{k(k-1)}{2n}}$$

La probabilité d'une collision est donc : 1- $e^{-\frac{k(k-1)}{2n}}$

Si l'on veut que cette probabilité soit inférieure à ε alors il suffit de prendre :

$$k \geq \sqrt{2nlog\frac{1}{1-\varepsilon}}$$

Si le haché (= le résumé) est codé sur *b* bits, il y a $n=2^b$ résumés possibles.

Question : Combien l'attaquant doit-il essayer de textes avant de trouver le même résumé avec une probabilité d'au moins $\varepsilon = 1/2$?

Une empreinte de 40 bits est vulnérable à une attaque des anniversaires avec probabilité supérieure à 1/2 en utilisant seulement 2^{20} messages aléatoires. Pour une empreinte de 128 bits il faut 2^{64} messages aléatoires. Le choix d'une empreinte de 160 bits donne une bonne résistance aux attaques anniversaires. Mais l'augmentation de la puissance de calcul disponible pour un attaquant pousse à augmenter la taille des empreintes, c'est pourquoi AES prévoit des empreintes de 256 `a 512 bits.

1.2. Exemple de fonction de hachage

Plusieurs variétés de condensats de message ont été proposées les plus utilisés MD5 et SHA-1.

1.2.1 La fonction de hachage MD5

- Proposé par Rivest en 1991
- Blocs de $b = 512$ bits, sortie de $n = 128$ bits
  - 16 sous-blocs M_i de 32 bits
  - 64 constantes fixées K_i
- 64 rondes sur 4 sous-blocs de 32 bits A, B, C, D
  - 4×16 sous-rondes où F prend les valeurs :
    1. $F = (B \text{ AND } C) \text{ OR } (\overline{B} \text{ AND } D)$
    2. $F = (D \text{ AND } B) \text{ OR } (\overline{D} \text{ AND } C)$
    3. $F = B \oplus C \oplus D$
    4. $F = C \oplus (B \text{ OR } \overline{D})$
- Sécurité de MD5 face aux collisions
  - Force brute : $\mathcal{O}(2^{64})$ (attaque anniversaire)

MD5 n'est plus considérée comme sûr aujourd'hui.

SHA-1: Secure Hash Algorithm 1

- Publié dans FIPS PUB 180-1 par NIST en 1995
- Blocs de $b = 512$ bits, sortie de $n = 160$ bits
  - 16 sous-blocs M_i de 32 bits
  - Etendus en 80 nouveaux blocs W_t
  - 4 constantes fixées K_t
  - 80 rondes sur 5 sous-blocs de 32 bits A,B,C,D,E
  - 4×20 sous-rondes où F prend les valeurs :

1. $F = (B \text{ AND } C) \text{ OR } (\overline{B} \text{ AND } D)$
2. $F = B \oplus C \oplus D$
3. $F = (B \text{ AND } C) \oplus (B \text{ AND } D) \oplus (C \text{ AND } D)$
4. $F = B \oplus C \oplus D$

- Sécurité de SHA-1 face aux collisions
  - Force brute : $\mathcal{O}(2^{80})$ (attaque anniversaire)

Bilan

Fonction	Empreinte	Compl. requise	Rés. aux coll.	Comp. attaque
MD5	128 bits	$\mathcal{O}(2^{64})$	Cassé [1]	$\mathcal{O}(2^{30})$
SHA-1	160 bits	$\mathcal{O}(2^{80})$	Cassé [2]	$\mathcal{O}(2^{63})$
HAVAL	256 bits	$\mathcal{O}(2^{128})$	Cassé [3]	$\mathcal{O}(2^{10})$
SHA-256	256 bits	$\mathcal{O}(2^{128})$	Sûr	
Whirlpool	512 bits	$\mathcal{O}(2^{256})$	Sûr	

2. Signature numérique

But des signatures manuscrites :

- prouver l'identité de leur auteur et/ou
- l'accord du signataire avec le contenu du document

La signature éléctronique dépend du signataire et du document/

Objectifs d'une signature éléctronique

- Une signature est authentique.
- Une signature ne peut être falsifiée (imitée).
- Une signature n'est pas réutilisable sur un autre document.
- Un document signé est inaltérable.
- Une signature ne peut pas être reniée.

Signature utilisant un cryptosystème à clef publique :

- Alice signe M en utilisant :
  - $h_M = H(M)$ le hachage de M
  - sa clé secrète K_d .
  - la fonction de déchiffrement D.
  - Résultat : $s(M) = D_{K_d}(h_M)$
- Document signé : $[M, s(M)]$
- Vérification de $[M, s(M)]$:
  - utilise la clé publique K_e d'Alice et la fonction de chiffrement E
  - $E_{K_e}(s(M)) = h_M = ?H(M)$
  - Seule Alice a pu générer $s(M)$

2.1 Signature RSA

Alice dispose de :

- Une clef publique : (n, e).
- Une clef secrète : d.
- Une fonction de hachage à sens unique H.

Génération d'une signature RSA

Alice souhaite signer un document M

- Alice calcule $h_M = H(M)$ (on suppose $0 \leq h_M < n$)
- Signature de M : $s(M) = (h_M)^d \mod n$
- Le document signé est alors $[M, s(M)]$.

Vérification d'une signature RSA

- Bob reçoit un document signé $[M', s(M)]$ d'Alice.
  Ce document est potentiellement altéré/illégitime
- Il récupère la clé publique d'Alice (n, e)
- Il calcule $h_{M'} = H(M')$
- Il vérifie l'identité : $s(M)^e = h_{M'} \mod n$
  En effet : $s(M)^e = (h_M)^{e.d} \mod n = h_M \mod n = h_M$ et si le document est authentique : $h_M = h_{M'}$.
- La sécurité est donc celle du cryptosystème RSA.

2.2 Signature El Gamal

Alice dispose de :

- Une clef publique : (p, g, A) avec $A=g^a$.
- Une clef secrète : a.
- Une fonction de hachage à sens unique H.

Génération d'une signature El Gamal

- Alice souhaite signer un document M
- Alice calcule $h_M = H(M)$ (on suppose $0 \leq h_M < p$)
- Elle choisit au hasard un entier $k \in [1, p-2]$ tel que $pgcd(k, p-1) = 1 (\Rightarrow k - 1 \in \mathbb{Z}_{p-1}$ existe).
- Signature de M : $s(M) = (r, s)$ avec $r = g^k \mod p$ et $s = k^{-1}(h_M - a.r) \mod (p-1)$
- Le document signé est alors $[M, s(M)]$.

Vérification d'une signature El Gamal

- Bob reçoit un document signé $[M', s(M)]$ d'Alice.
  - Rappel : $s(M) = (r, s)$
  - Ce document est potentiellement altéré/illégitime
- Il récupère la clé publique d'Alice (p, g, A)
- Il calcule $h_{M'} = H(M')$
- Il vérifie l'identité : $A^r r^s = g^{h_{M'}} \mod p$

  En effet, $A^r r^s = g^{a.r} . g^{kk^{-1}(h_M - a.r)} \mod p$
  $= g^{h_M} \mod p$

  Si le document est authentique : $h_M = h_{M'} \Rightarrow g^{h_M} = g^{h_{M'}} \mod p$

2.3 Le standard DSS :Digital Signature Standard

- En 1991 le NIST a proposé d'utiliser une variante **d'EL Gamal** pour son nouveau standard DSS.
- DSS se base sur l'algorithme DSA.
- **DSA (Digital Signature Algorithm)**

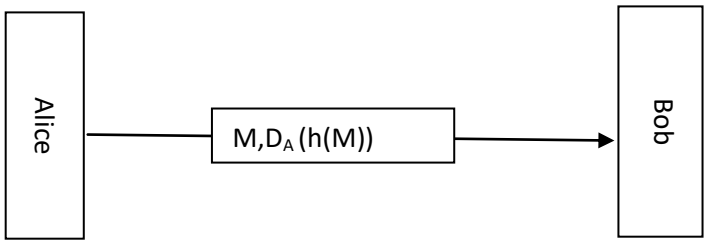
La sécurité de cet algorithme repose sur la difficulté à calculer les logarithmes discrets dans un groupe fini.

3. Exercices

Exercice 6.1

Soit le schéma de signature El Gamal avec p = 467, g = 2 et a = 127. Signez est vérifiez les messages w = 100 et r = 213.

Exercice 6.2



Signature numérique à l'aide d'une fonction de hachage

A la figure au-dessus nous voyons comment Alice peut envoyer à Bob un message signé. Si un intrus remplace M, Bob peut le détecter, mais qu'arrive-t-il s'il remplace à la fois M et la signature ?

Chapitre 7

Autorité de Certification et Protocoles d'authentification

1. Introduction

- si Alice et Bob ne se connaissent pas, comment vont-ils faire pour se transmettre leurs clés publiques afin d'entamer leur échange de courrier ?

- La solution simple : qu'ils mettent tous deux leur clé publique sur leur site Web.

- Mais si Alice voudrait utiliser la clé publique de Bob qui l'affiche sur son site Web.

- Eve un intrus intercepte la requête et y répond en renvoyant une fausse page d'accueil avec une clé publique qui est la sienne.

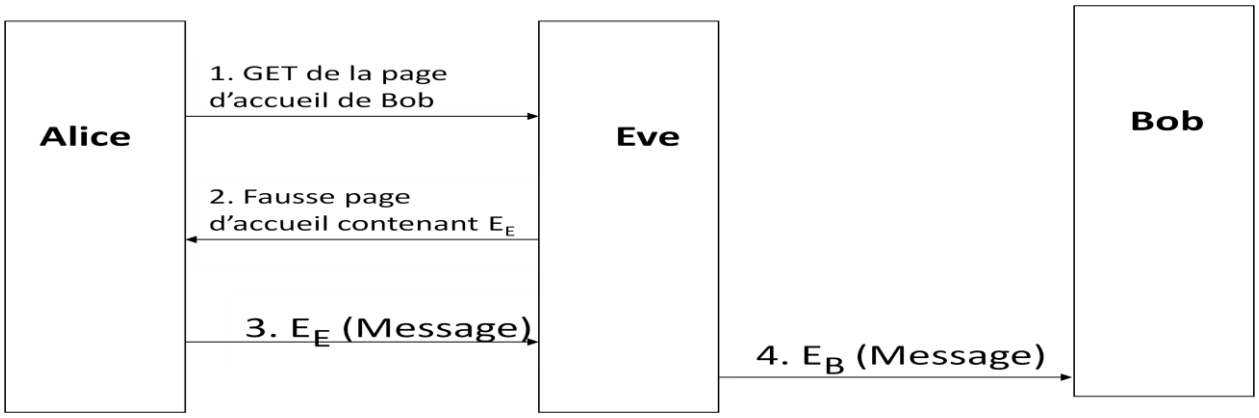


Figure 7.1 : clé publique de Bob

Pour résoudre ce problème :

- une première tentative consiste à faire appel à un centre de distribution de clés ouvert 24h/24 et chargé d'envoyer les clés publiques à qui les demande.

Le problème : ce centre deviendrait rapidement un goulet d'étranglement, et si jamais il tombait en panne, la sécurité serait fortement compromise.

Pour cette raison, une solution différente a été imaginée, qui ne fait pas appel à une distribution des clés publiques centralisée.

- On va certifier que les clés publiques appartiennent bien à leurs légitimes propriétaires (individus, entreprises, organisations), cet organisme certificateur est appelé CA (Certification Athority).

Un certificat permet d'attester l'identité du destinataire de la même façon qu'un papier d'identité permet d'identifier une personne.

2. Infrastructure à clef publique (ICP-PKI)

L'infrastructure à clef publique définit un ensemble de services de sécurité afin de rendre les échanges électroniques fiables.

- A l'intérieur de l'ICP, les éléments sont organisés hiérarchiquement dans le but de garantir un niveau de sécurité élevé. Elle est composée de :
- une autorité d'enregistrement (Registration Authority)
- une autorité de certification
- un système de distribution des clefs

Les ICP sont évolutives et interopérables c'est-à-dire qu'elles sont capables de suivre la croissance du nombre d'utilisateurs et doivent supporter l'ajout de nouvelles autorités de certification et l'établissement de certification croisée entre plusieurs autorités.

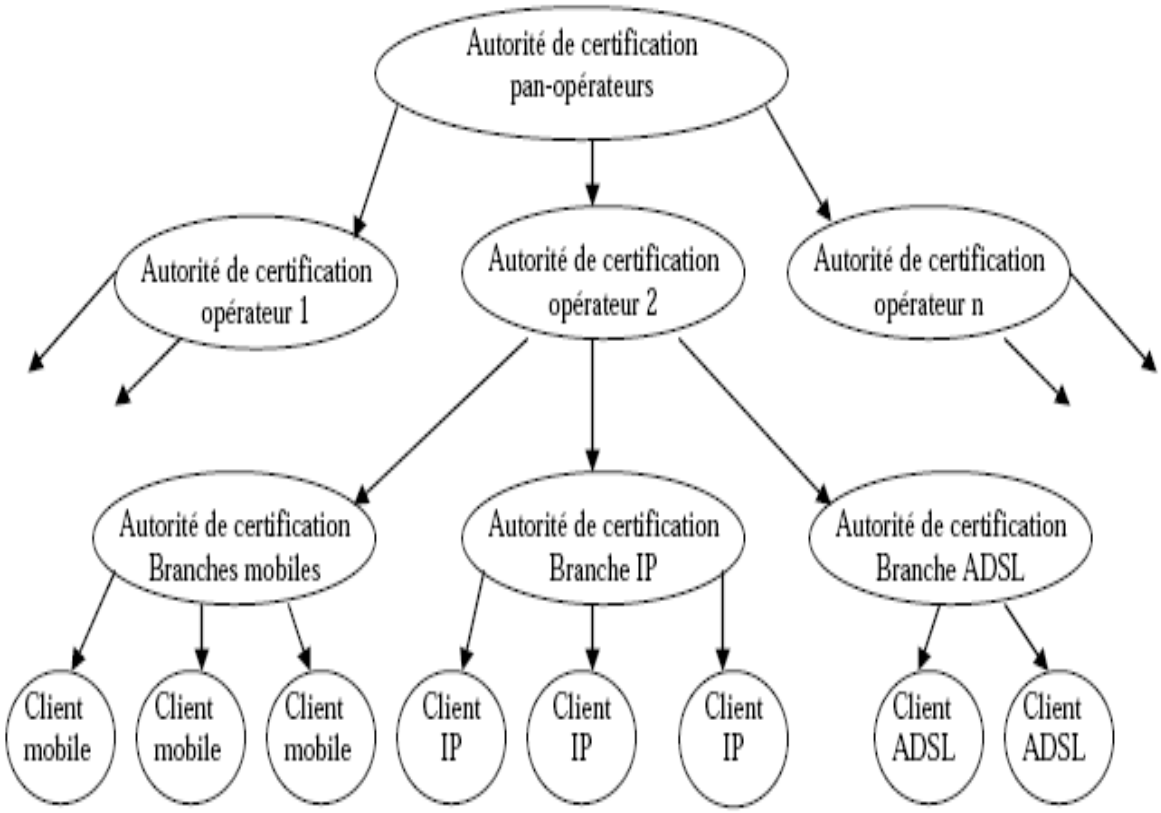


Figure 7.2 : organisation des autorités de certification

2.1 Obtenir un certificat numérique

Il existe trois classes de certificats correspondant à différents niveaux de sécurité :

- Certificats de classe 1 : Le demandeur ne fournit qu'une adresse e-mail.
- Certificats de classe 2 : Ils requièrent une preuve d'identité du demandeur.
- Certificats de classe 3 : Ces certificats ne peuvent être délivrés que si le demandeur est présent physiquement.

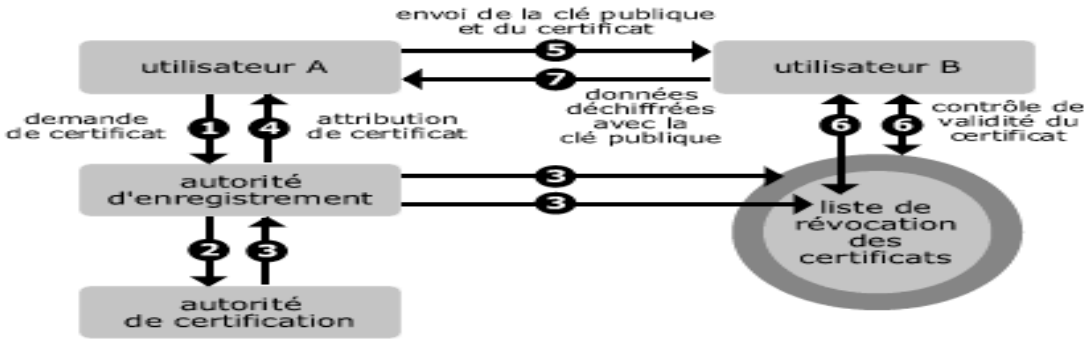


Figure 7.3 : Obtenir un certificat numérique

Revenons au problème précédent : l'intrus peut placer son propre certificat sur sa fausse page d'accueil. Alice lira le certificat elle verra que ce n'est pas le nom de Bob qui y figure.

L'intrus peut modifier la page d'accueil de Bob en remplaçant la clé publique de celui-ci par la sienne, cependant, lorsqu'Alice exécutera l'algorithme SHA-1 sur le certificat elle obtiendra un hachage qui ne sera pas identique à celui qu'elle obtient lorsqu' elle applique la clé publique bien connue de la CA au bloc de signature.

2.2 Cycle de vie d'un certificat

Pour des raisons de sécurité, un certificat est accordé pour une durée limitée.

Il peut être également remis en question dans la période de validité pour différentes raisons :

- volonté du détenteur du certificat
- changement de situation du détenteur
- volonté de l'autorité de certification
- sécurité

Le certificat est alors suspendu ou révoqué, la suspension ou la révocation étant notifié dans un annuaire spécifique facilement accessible (en ligne).

2.3 Certificats X.509

Le format de certificats numériques X.509 de l'ISO (Organisation Internationale de Normalisation) est le plus répandu.

Un certificat X.509 est composé de la signature de l'autorité de certification et d'informations tel que :

- la version de la norme X.509 désigné par un nombre entier : 0 pour la version 1, 1 pour la version 2 et 2 pour la version 3.
- le numéro de série : chaque certificat a un numéro unique.
- l'algorithme de signature utilisé par l'autorité : MD5, DSA, etc…
- la période de validité du certificat
- le nom de l'autorité de certification
- le nom du sujet
- renseignements sur la certification de la clef publique :
    - algorithme utilisé
    - chaîne de bits représentant la clef publique
- informations optionnelles spécifiques à la version 3

```
Certificate:
   Data:
      Version: v3 (0x2)
      Serial Number: 3 (0x3)
      Signature Algorithm: PKCS #1 MD5 With RSA Encryption
      Issuer: OU=Ace Certificate Authority, O=Ace Industry, C=US
      Validity:
         Not Before: Fri Oct 17 18:36:25 1997
         Not After: Sun Oct 17 18:36:25 1999
      Subject: CN=Jane Doe, OU=Finance, O=Ace Industry, C=US
      Subject Public Key Info:
         Algorithm: PKCS #1 RSA Encryption
         Public Key:
            Modulus:
               00:ca:fa:79:98:8f:19:f8:d7:de:e4:49:80:48:e6:2a:2a:86: [...]
      Extensions:
            Identifier: Certificate Type
               Critical: no
               Certified Usage:
                  SSL Client
            Identifier: Authority Key Identifier
               Critical: no
               Key Identifier:
                  f2:f2:06:59:90:18:47:51:f5:89:33:5a:31:7a:e6:5c:fb:36:
                  26:c9
      Signature:
         Algorithm: PKCS #1 MD5 With RSA Encryption
         Signature:
            6d:23:af:f3:d3:b6:7a:df:90:df:cd:7e:18:6c:01:69:8e:54:65:fc:06: [...]
```

Exemple de certificat X.509 version 3

- Certificate signature Value : Valeur de signature du certificat;
- Certificate signature algorithm : algorithme et niveau de chiffrement utilisé;
- Version : Version du certificat;
- Serial Number : Numéro de série dans l'AC;
- Issuer : AC validante;
- Validity : durée de validité;
- Subject : Distinguished Name (DN), champ servant généralement à nommer le certificat. Il est donc important de bien choisir son DN:
    - Il doit être lisible par l'homme avec quelques restrictions des bonnes pratiques (pas d'email, pas de rôle).
    - L'identifiant unique interne d'un certificat et (Issuer + Serial Number + version), mais est peu lisible. Dans l'organisation c'est le DN qui remplit ce rôle.

- Subject Public Key : clé public;
- Subject Public Key algorithm : algorithme et niveau de chiffrement utilisé;
- Extensions : informations critique et non critique sur le nombre d'AC, sur les points de distribution et :
    - Certificates Key Usage : Authentification, chiffrement, signature, CRL, etc;
    - Certificate Extended Key Usage : Horodatage, OCSP, EFS, etc.

2.4 Les composants

L'infrastructure de gestion de clé se découpe généralement en quatre parties :

1) Autorité de certification (AC) : en charge de la signature des demandes de certificats (CSR) et des listes de révocations (CRL).
2) Autorité d'enregistrement (AE) : en charge de la vérification et de la délivrance des identities.
3) Autorité de dépôt : stockage des certificats et des listes de revocations.
4) Sujet : utilisateur ou système, sujet de l'identité appelé aussi entité terminale (EE).

Auquel peuvent s'ajouter d'autres éléments :

- Autorité de séquestre : stockage centralisé des clés de chiffrement à des fins de recouvrement.
- Certaines solutions offrent des éléments supplémentaires : Une entité d'enrôlement (EE) qui est un comptoir pour faire les demandes (spécificité Open-Trust).

L'infrastructure de gestion de clé est une structure hiérarchique pyramidale :

- Au sommet une AC "Racine" : qui est la base de la confiance
- Des AC intermédiaires:
    - Elles sont certifiées par l'AC racine;
    - Elles peuvent délivrer des certifications a d'autres AC intermédiaires ou opérationnelle.
- Des AC opérationnelles:
    - Elles sont certifiées par l'AC racine ou par une AC intermédiaire.
    - Elles peuvent délivrer des certification aux entités terminales.

2.5 Les protocoles

Les PKCS (Public Key Cryptographic Standards) sont les protocoles standards de sécurité permettant l'échange des informations au sein d'une PKI:

#1: Cryptographie Standard;

#3: Hellman clef Agreement Standard;

#6 : Certificate Syntax Standard#7Message Syntax Standard;

#8: Clef Information Syntax Standard;

#10: Request Syntax Standard;

#11: Token Interface Standard;

#12: Information Exchange Syntax Standard;

#15: Token Information Format Standard.

Les plus couramment utilisés sont PKCS10, PKCS11, PKCS12 aussi appelés enveloppes P10 ou P12.

3. Protocoles d'authentification

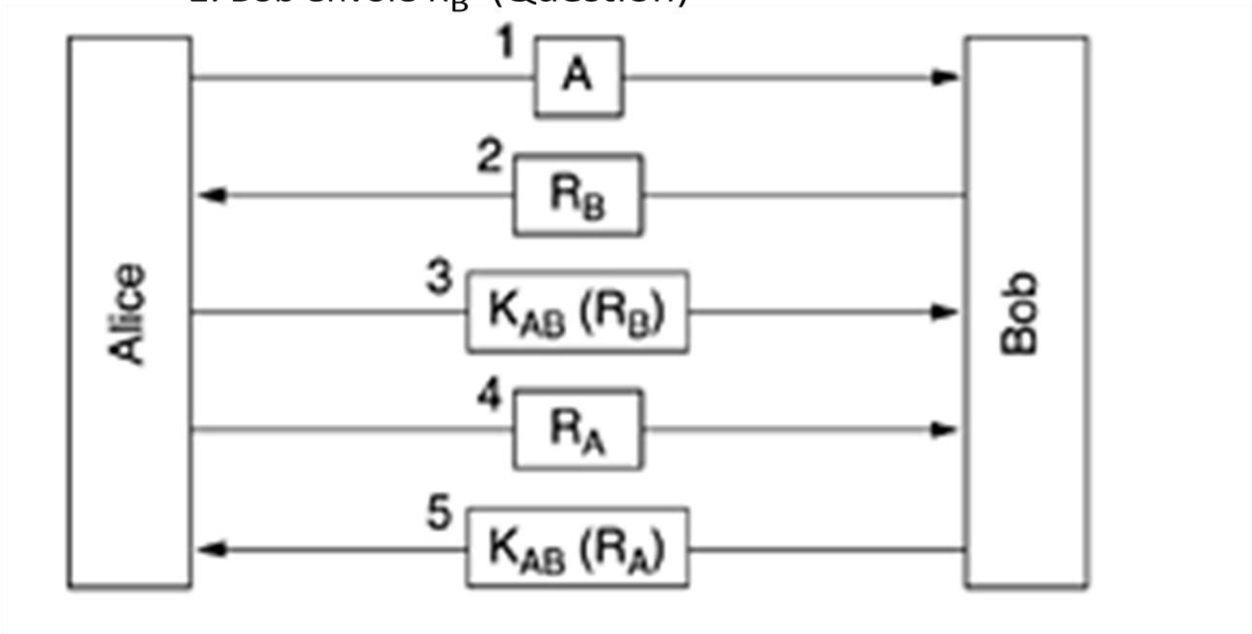
L'authentification consiste à vérifier que lors de l'établissement d'une communication le partenaire est bien celui qu'il suppose être et non un imposteur. Cela nécessite la mise en place de protocoles complexes reposant sur l'usage de la cryptographie.

3.1 Authentification fondée sur une clé secrète partagée

Pour ce protocole (Figure 7.3) nous supposons qu'Alice et Bob partagent déjà une clé secrète K_{AB}. le protocole repose sur un principe que l'on rencontre dans de nombreux protocoles d'authentification ou l'une des deux parties envoie un nombre aléatoire à l'autre qui le transforme d'une façon particulière et le renvoie, ce type de protocole est appelé protocole Question-Réponse.

1. Alice envoie son identité A à Bob .

2. Bob envoie R_B (Question)



3. Alice chiffre R_B et envoie le message $K_{AB}(R_B)$.

Figure 7.4 : Authentification fondée sur la clé secrète

Ce protocol est vulnerable à l'attaque réflexion (Figure 7.5) qui aura lieu si Alice peut ouvrir plusieurs sessions avec Bob en même temps.

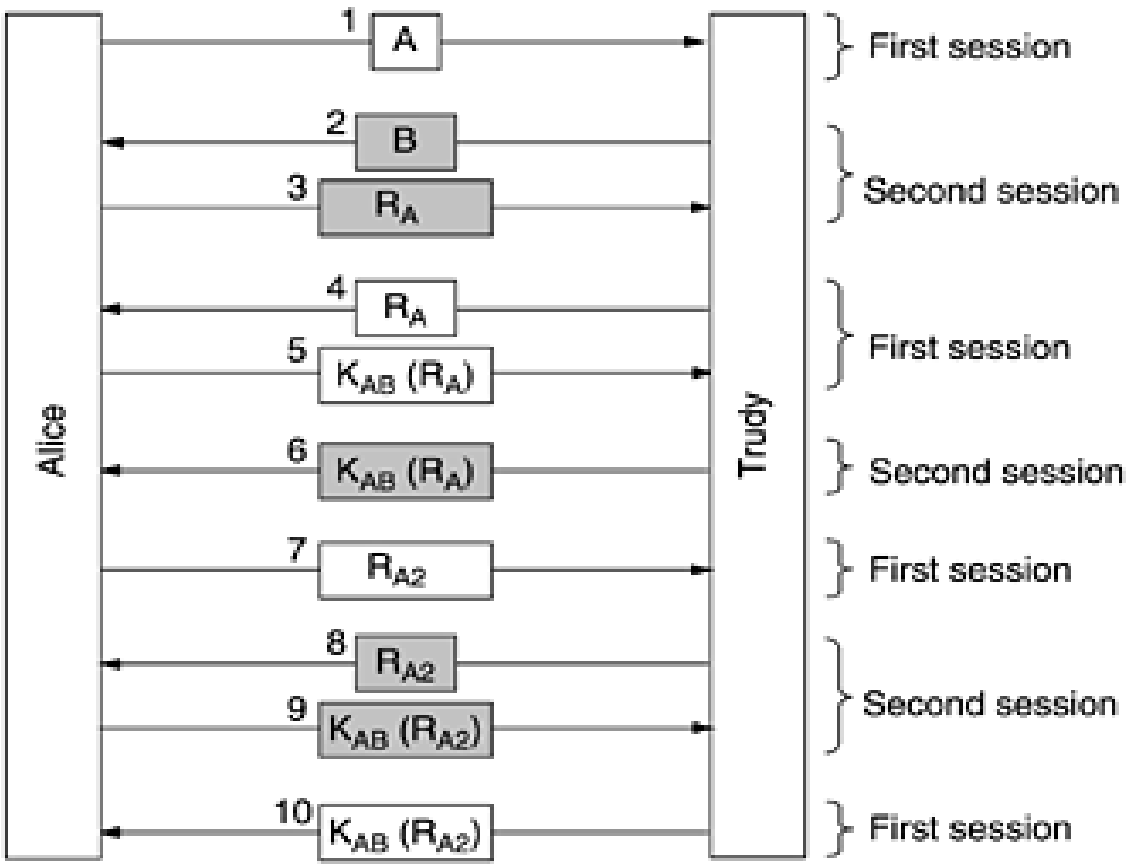


Figure 7.5 : Attaque par réflexion

3.2 Authentification à l'aide d'un centre de distribution de clés

Cette approche nécessite l'intervention d'un centre de distribution de clés (KDC) en qui tout le monde ait confiance. Dans ce modèle chaque utilisateur à sa propre clé qu'il partage avec le KDC. La gestion de l'authentification et des clés de session passe par le KDC.

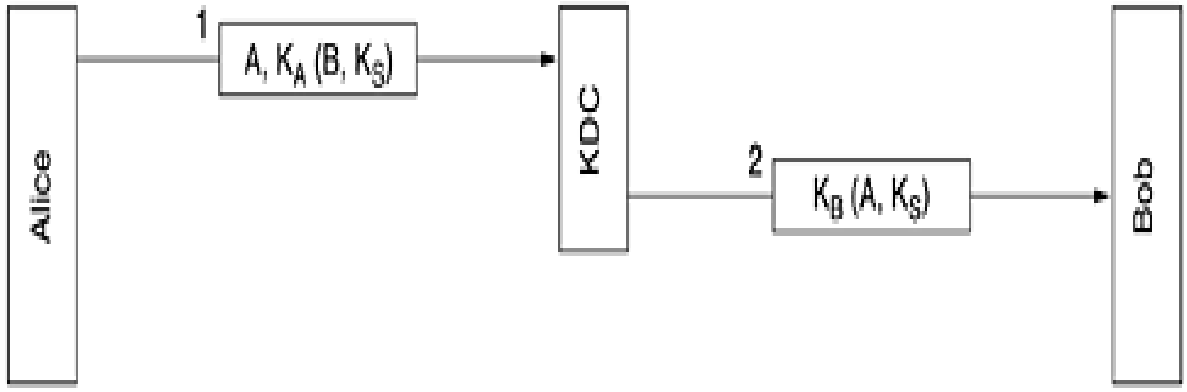


Figure 7.6 : Authentification à l'aide d'un KDC

Alice choisit une clé de session K_s et annonce au KDC qu'elle veut dialoguer avec Bob en utilisant cette clé. Le message est chiffré avec la clé secrète d'Alice K_A qu'elle ne partage qu'avec le KDC.

Le KDC déchiffre le message et en extrait l'identité de Bob ainsi que la clé de session, il construit alors un nouveau message contenant l'identité d'Alice et la clé de session et l'envoie à Bob avec la clé K_B.

Dans ce cas on a une faille appelée attaque par rejeu (replay attack) ou un intrus capte le message2 et le suivant par exemple une transaction bancaire et rejout les messages une autre fois. Pour remédier on utilise le protocole d'authentification de NeedHam-Schroeder(Figure 7.7)

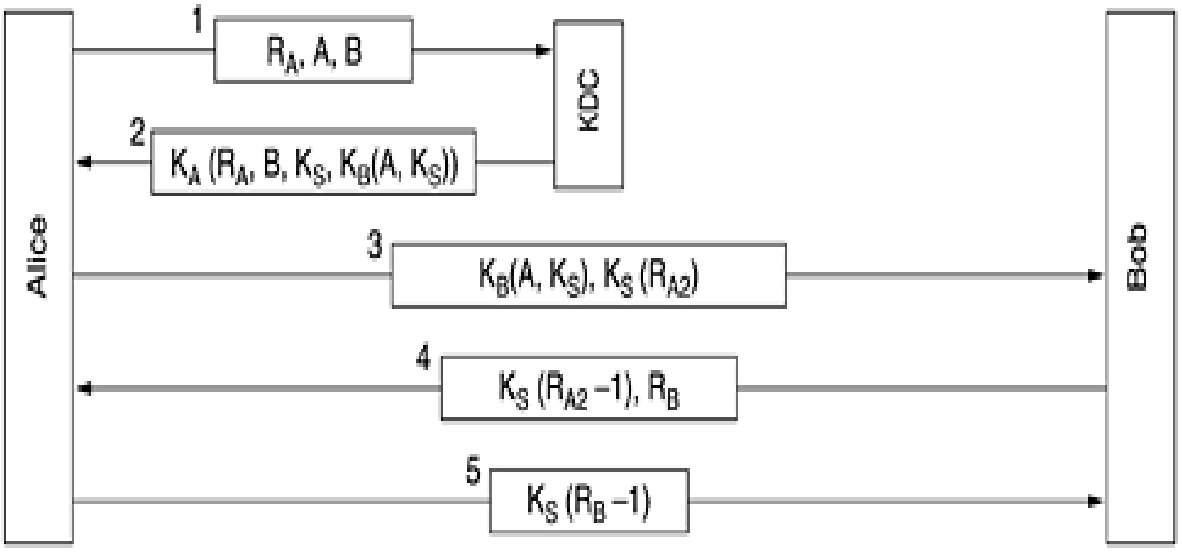


Figure 7.7 : Le protocole d'authentification de NeedHam-Schroeder

Pour ce protocole on a un petit problème, si un intrus réussit à obtenir une ancienne clé de session en clair, il peut initier une nouvelle session avec Bob en rejouant le message3. Solution : le protocole d'authentification Otway-Rees :

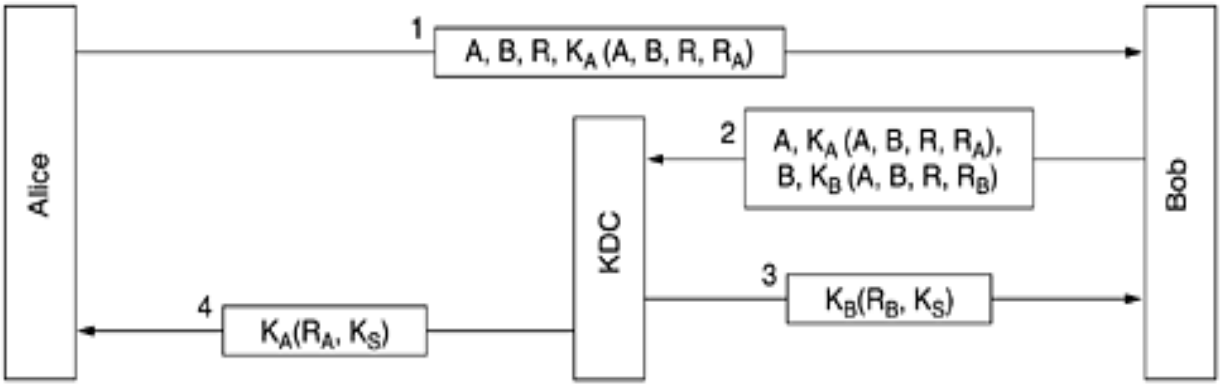


Figure 7.8 : Le protocole d'authentification de Otway-Rees

3.3 Authentification au moyen de Kerberos

Kerberos: nom du chien à plusieurs têtes qui dans la mythologie grecque gardait l'entrée des enfers. C'est un protocole d'authentification fondé sur une variante du protocole Needham-Shroeder. Figure 7.8

- Alice : une station de travail cliente.
- Kerberos utilise trois serveurs :
    - Un serveur d'authentification AS qui contrôle les utilisateurs au cours de la connexion (Login).
    - Un serveur de vérification de Ticket (Tucket Granting Server) TGS qui délivre des preuves de tickets d'identité.
    - Bob : le serveur qui va effectuer le travail demandé par Alice.
- AS joue le rôle identique à celui du KDC.

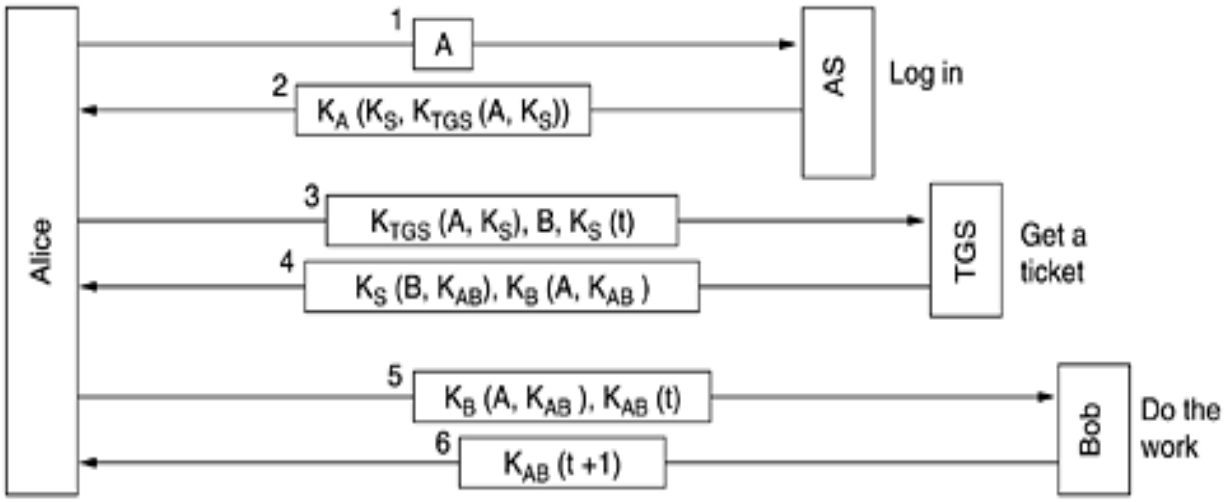


Figure 7.9 : Authentification au moyen de Kerberos

3.4 Authentification par cryptographie publique

Pour commencer Alice doit connaître la clé publique de Bob. A partir d'une PKI Alice demande la clé publique de Bob (incluse dans son certificat). Figure 7.9

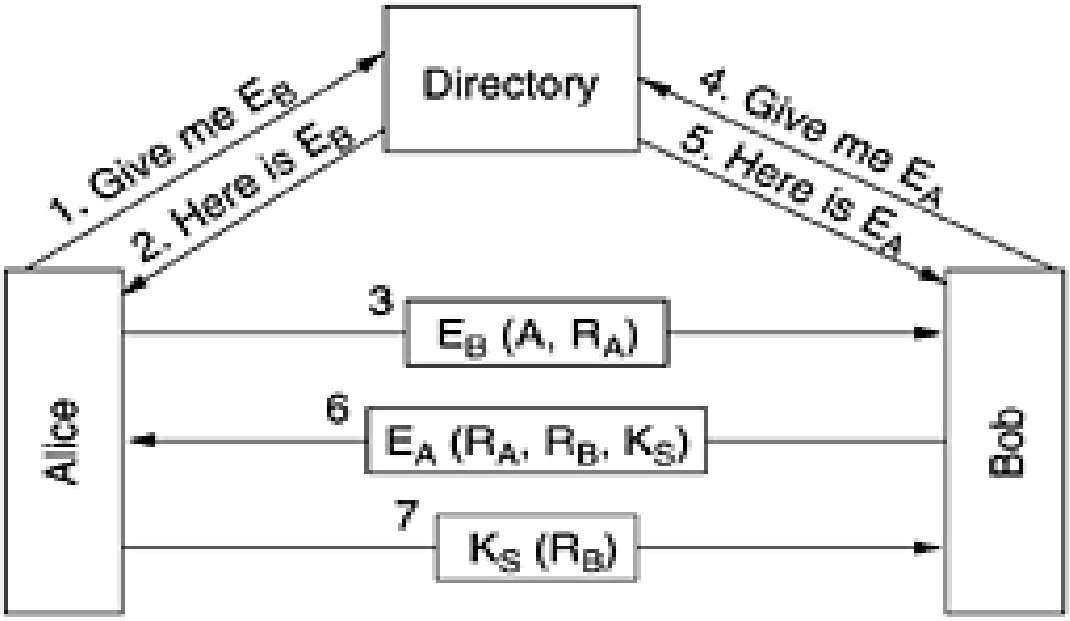


Figure 7.10 : Authentification par cryptographie publique

4. Preuves sans connaissance (Zero-knowledge proofs)

Les preuves sans connaissance (ZKP) sont des protocoles interactifs dans lesquels une partie, appelée le prouveur, peut convaincre l'autre partie, appelée le vérificateur, qu'une affirmation est vraie sans rien révéler d'autre que le fait que l'assertion prouvée est vraie. Pour cela il faut disposer d'un secret personnel *s* et d'un protocole qui permet de persuader le vérifieur(un ordinateur ou une personne) auquel on se connecte (s'adresse) qu'on connait *s* sans avoir à le révéler et ce à chaque fois à l'aide de messages différents. Cette propriété fait des ZKP des outils très puissants pour la conception de protocoles cryptographiques sécurisés.

Les ZKP peuvent être soit interactifs où un prouveur convainc un vérificateur spécifique mais doit répéter ce processus pour chaque vérificateur individuel ou non interactif où un prouveur génère une preuve qui peut être vérifiée par toute personne utilisant la même preuve.

Les trois caractéristiques fondamentales qui définissent un ZKP comprennent :

Un exemple (Figure 7.10) conceptuel pour comprendre intuitivement la preuve sans connaissance ZRP consiste à imaginer une grotte avec une seule entrée mais deux chemins (chemins A et B) qui se connectent à une porte commune verrouillée par une phrase secrète. Alice veut prouver à Bob qu'elle connaît le code d'accès à la porte mais sans révéler le code à Bob. Pour ce faire, Bob reste à l'extérieur de la grotte et Alice marche à l'intérieur en empruntant l'un des deux chemins (sans que Bob sache quel chemin a été emprunté). Bob demande alors à Alice de reprendre l'un des deux chemins jusqu'à l'entrée de la grotte (choisi au hasard). Si Alice a choisi le chemin A elle reprend ce chemin jusqu'à la porte, mais si Bob lui demande ensuite de reprendre le chemin B, la seule façon de faire est qu'Alice connaisse le code d'accès de la porte verrouillée. Ce processus peut être répété plusieurs fois pour prouver qu'Alice a connaissance du code d'accès de la porte et n'a pas choisi le bon chemin à prendre initialement avec un degré élevé de probabilité. Une fois ce processus terminé, Bob a un degré élevé de confiance qu'Alice connaît le code d'accès de la porte sans révéler le code d'accès à Bob.

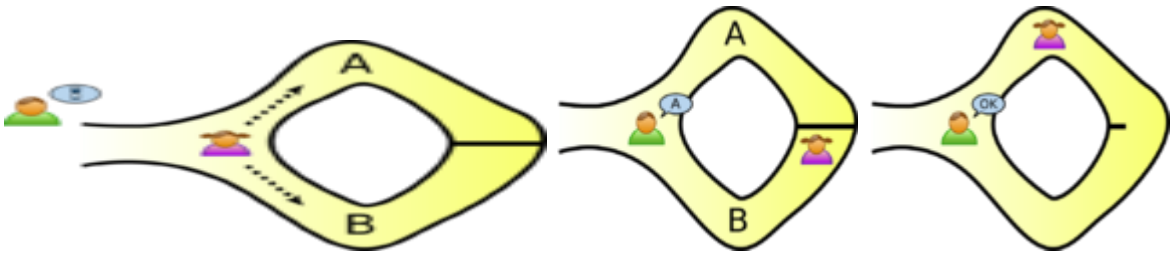


Figure 7.11 : Exemple preuve sans connaissance

Exemple 7.1 Le protocole d'identification de Fiat-Shamir

La sécurité repose sur la difficulté d'extraire des racines carrées modulo de grands entiers composés *n* de factorisation inconnue ainsi que sa factorisation.

But : A prouve la connaissance de s à B pendant t exécution du protocole

1. Initialisation

a) Un centre de confiance T sélectionne et publie un module de type RSA n = pq mais garde secrets les nombres premiers p et q.

b) Chaque demandeur A sélectionne un secret *s* premier avec *n*, $1 \leq s \leq n-1$, calcule $v = s^2 \bmod n$, et enregistre *v* avec T comme clé publique.

2. les différents messages du protocole : Chacune des t tours a trois messages avec la forme suivante :

A envoie à B : $x = r^2 \bmod n$ (1) …….. Témoin

B envoie à A : $e \in \{0, 1\}$ (2)……. Défi

A envoie à B : $y = r.s^e \bmod n$ (3) ……. La réponse

3. Les étapes suivantes sont itérées t fois (séquentiellement et indépendamment). B accepte la preuve si tous les t tours réussissent :

(a) A choisit *r*, $1 \leq r \leq n-1$ et envoie $x=r^2 \bmod n$ à B

(b) B choisit aléatoirement $e \in \{0, 1\}$ et l'envoie à A

(c) A calcule et envoie à B $y=r.s^e \bmod n$ (soit $y=r \bmod n$ si $e=0$ ou $y=r.s \bmod n$ si $e=1$)

(d) B accepte si $y^2 = x.v^e \pmod n$ (Selon e=0, $y^2 = x$ ou $y^2 = x.v \bmod n$, tel que $v = s^2 \bmod n$) et rejette si non.

Supposons que A veuille tromper B, choisisse un *r* aléatoire et envoie $x=r^2/v$, si e = 0, alors A renvoie avec succès à B $y = r$, dans le cas ou $e = 1$, A ne pourra pas répondre correctement puisqu'elle ne connaît pas *s*, et avoir la racine carrée de *v* modulo *n* est assez difficile. Si A connaît vraiment le secret, elle pourra toujours répondre correctement, quel que soit le *e* sélectionné.

Un prouveur A sachant s peut répondre aux deux questions, mais sinon peut au mieux répondre à l'une des deux questions, et n'a donc qu'une probabilité de 1/2 d'échapper à la détection. Pour

diminuer la probabilité de tricher on choisit une valeur acceptablement petite de 2^{-t} (par exemple, $t = 20$ ou $t = 40$, 2^{-t} est $1/2^t$ qui est la probabilité de tricher pendant t exécution de l'algorithme), le protocole est itéré t fois, B accepte l'identité de A uniquement si toutes les t questions (sur t tours) sont répondu avec succès.

Ce protocole a été amélioré pour garantir plus de sécurité, on trouve le protocole d'identification Feige-Fiat-Shamir et autres.

Chapitre 8

Autres cryptogrammes à clé public

1. Les courbes elliptiques

Définition 8.1 : soit a,b deux nombres réel tels que $4a^3+27b^2\neq 0$. Une courbe elliptique non singulière E définie sur R est l'ensemble des points de coordonnées (x,y) de R^2 qui sont des solutions de l'′equation :

$$y^2 = x^3 + ax + b \qquad (8.1)$$

Plus un point spécial, noté O ou ∞, appelé le point à l'infini.

La condition $4a^3 + 27b^2 \neq 0$ assure que la courbe est sans point double. Si $4a^3 + 27b^2 \neq 0$ on a affaire à une courbe singulière.

La figure 8.1 est la courbe elliptique qui correspond à $y^2=x^3-4x$

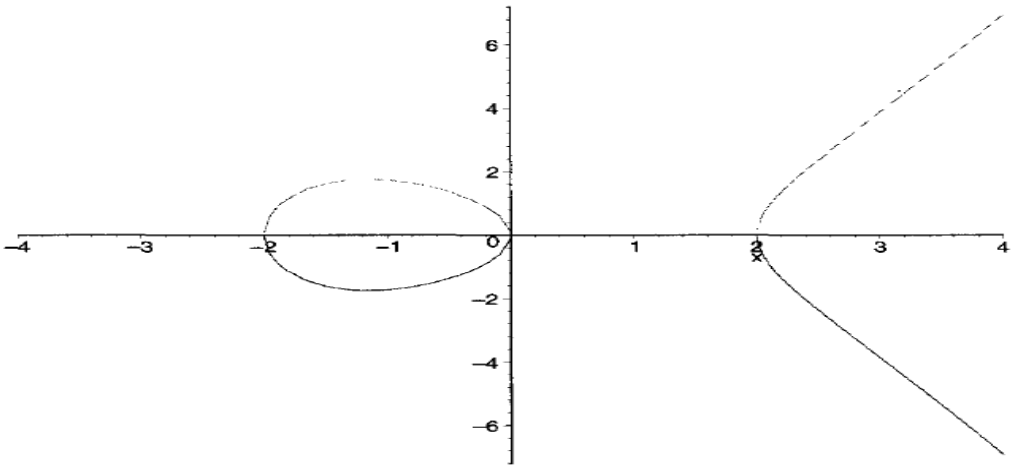


Figure 8.1 : une courbe elliptique sur R

1.1 Groupe des points d'une courbe elliptique

Si E est une courbe elliptique non-singulière on définit sur les points de E une addition notée + qui fera de l'ensemble des points de E ($E(K) = \{ (x, y) \in K^2 \mid y^2 = x^3 + ax + b\} \cup \{O\}$avec $4a^3 + 27b^2 \neq 0$ ou K est un corps commutatif) un groupe abélien :

- $\forall$ P ϵ E, par définition P + O = O + P = P. ou O est l'élément identité

Soit P ϵ E et Q ϵ E ou $P=(x_1,y_1)$ et $Q=(x_2,y_2)$, on considère les 3 cas :

1) $x_1 \neq x_2$
2) $x_1 = x_2$ et $y_1 = -y_2$

3) $x_1 = x_2$ et $y_1 = y_2$

Cas 1 :

On note L la droite passant par P et Q. Elle coupe E en 3 points dont deux, P et Q, sont déjà connus, on note R′ = (x′, y′) le troisième point d'intersection. On prend le symétrique de R′ par rapport à l'axe des abscisses. On obtient un point R = (x′,−y′) = (x, y) qui est encore sur E. On pose par définition

P + Q = R

Par un calcul sans mystère on trouve l'équation de L

y = λx + ν tel que la pointe de L $\lambda=\frac{y_2-y_1}{x_2-x_1}$ et $\nu = y_1 - \lambda x_1 = y_2 - \lambda x_2$

D'où $y = \left(\left(\frac{y_2-y_1}{x_2-x_1}\right)x + \left(y_1 - \frac{y_2-y_1}{x_2-x_1}x_1\right)\right)$

Pour trouver l'intersection de E∩L, on remplace y = λx + ν dans l'équation de E on obtient :

$(\lambda x + \nu)^2 = x^3+ax+b$ qui donne $x^3 - \lambda^2x^2+(a-2\lambda\nu)x+b-\nu^2=0$ (8.2)

L'équation (8.2) est une équation cubique sur *R(nombres réels)* ayant deux racines réelles, la troisième racine , x', doit aussi être réelle. La somme des trois racines doit être le négatif du coefficient du terme quadratique (λ^2) de l'équation (8.2).

Donc $x' = \lambda^2 - x_1 - x_2$ est la x-cordonnée du point R′ alors la x-coordonnée de R est x= x'

Pour l'y-coordonnée de R′ (y') alors la y-coordonnée de R est ($y = -y'$)

On a $\lambda=\frac{y'-y_1}{x'-x_1}$ alors $y' = \lambda(x' - x_1) + y_1$

On obtient alors les coordonnées de R

$x= \lambda^2 - x_1 - x_2$ et $y= \lambda(x_1 - x') - y_1$ avec $\lambda= \frac{y_2-y_1}{x_2-x_1}$

Cas 2 :

On définie (x,y)+(x,-y) = *O* pour tous les $(x,y) \in E$ ou (x,y) et (x,-y) sont des inverses par rapport à l'addition dans les courbes elliptiques.

Cas 3 :

Dans ce cas on fait P+P on suppose que y≠0 (si non on est dans le cas 2), le cas 3 ressemble au cas 1 sauf qu'on définit L comme la tangente de E au point P. la pointe de L est calculée en utilisant la différenciation de l'équation de E :

$$2y\,\frac{dy}{dx} = 3x^2 + a$$

On substituant *x=x1 et y=y1* donc la pointe $\lambda= \frac{3x_1^3+a}{2y_1}$

On obtient le même résultat que le cas1 en changeant λ.

On montre que la loi ainsi définie est :

1. Stable sur E: si P et Q appartiennent à E, P + Q ∈ E
2. Associative: P, Q et R appartiennent `a E alors (P + Q) + R = P +(Q + R)
3. Commutative: P + Q = Q + P
4. possède un élément neutre, O: P + O = O + P = P
5. Chaque point de E admet un opposé pour cette addition: ∀ P ∈E ∃Q ∈ E tel que P + Q = O

Cette loi fait donc des points de E définis sur K un groupe abélien.

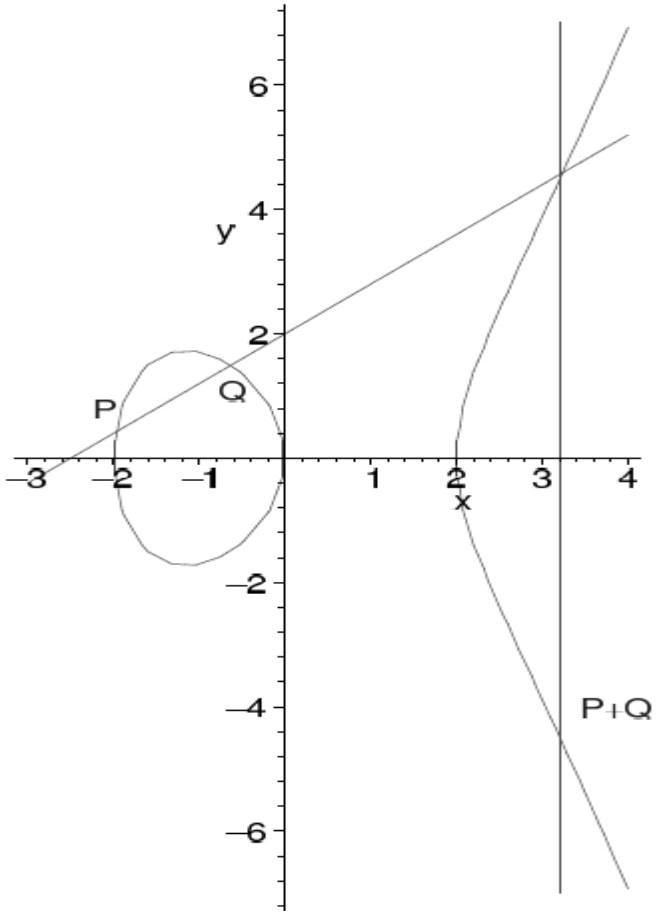


Figure 8.2 : P+Q dans la courbe elliptique sur R, $y^2=x^3- 4x$

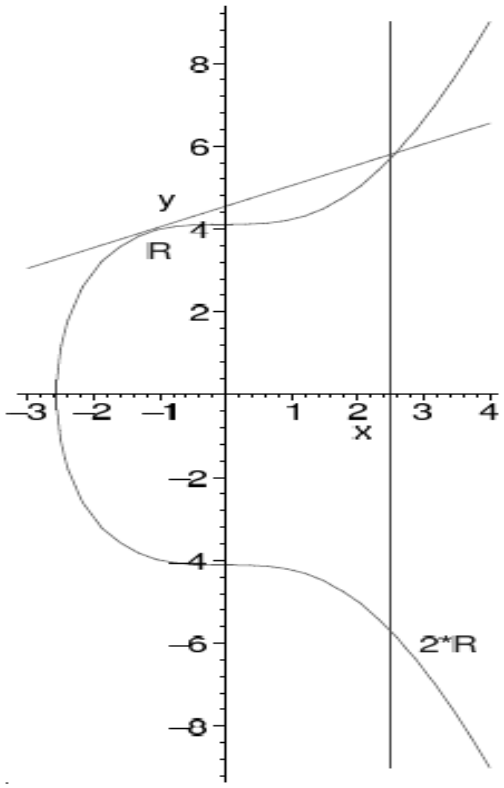


Figure 8.3 : R+R=2R dans la courbe elliptique sur *R*, $y^2 = x^3 + 17$

Remarque : les formules donnant les coordonnées de la somme de deux points P et Q sur la courbe elliptique E sont des fonctions rationnelles à coefficients dans ***Z*** de leurs coordonnées. Ce qui montre que l'appartenance des coordonnées de P et Q à ***R*** ne joue aucun rôle.

1.2 Courbe elliptique modulo un nombre premier

Soit p>3 un nombre premier, les courbes elliptiques sur $\boldsymbol{Z_p}$ peuvent être définis exactement comme ils ont été définis sur les réels

Définition 8.2 : Soit p >3 un nombre premier, la courbe elliptique $y^2 = x^3 + ax + b$ sur $\boldsymbol{Z_p}$ est l'ensemble des points de coordonnées $(x,y) \in \boldsymbol{Z_p} X\ \boldsymbol{Z_p}$ de la congruence

$$y^2 \equiv x^3 + ax + b \pmod{p}$$

Avec a, b deux nombres réels tels que $4a^3+27b^2 \neq 0 \pmod{p}$.

Plus un point spécial, noté O ou ∞, appelé le point à l'infini.

L'opération d'addition est définie comme suit : soit $P(x_1,y_1)$et $Q(x_2,y_2)$ deux points de **E**

Si $x_2=x_1$ et $y_2= - y_1$ Alors $P+Q=O$

Si Non $P+Q=(x_3,y_3)$ avec

$x_3= \lambda^2 - x_1-x_2$ et $y_3= \lambda(x_1 - x_3) - y_1$ avec $\lambda= \frac{y_2-y_1}{x_2-x_1}$ si P ≠ Q

ou

$\lambda= (3x_1^3 + a)(2y_1)^{-1}$ si P=Q.

Exemple 8.1

Calculons par exemple les points de la courbe elliptique

y2 = x3 + x + 6 dans $F_{11}(q=p^e)$

x	$x^3 + x + 6 \mod 11$	résidu quadratique?	y
0	6	non	
1	8	non	
2	5	oui	4,7
3	3	oui	5,6
4	8	non	
5	4	oui	2,9
6	8	non	
7	4	oui	2,9
8	9	oui	3,8
9	7	non	
10	4	oui	2,9

Cette courbe sur F11 admet 13 points y compris celui à l'infini.

Le nombre de points dans une courbe elliptique

Un résultat de HASSE montre que $p^e+1-2\sqrt{p^e} \leq |E(F_p^e)| \leq p^e+1+2\sqrt{p^e}$

Le calcul de $|E(F_p^e)|$ est difficile mais il existe un algorithme performant pour le faire l'algorithme de Schoof.

Multiplication d'un point d'une EC par un entier

Deux points sur une courbe elliptique peuvent être ajoutés et le résultat est un autre point. Cette opération est connue sous le nom addition de points d'une EC. Si nous ajoutons un point G à lui-même, le résultat est G + G = 2 * G. Si nous ajoutons à nouveau G au résultat, nous obtiendrons 3 * G et ainsi de suite. C'est ainsi que la multiplication des points d'une EC est définie. Un point G sur une courbe elliptique sur un corps fini peut être multiplié par un entier *k* et le résultat est un autre point EC P sur la même courbe et cette opération est rapide :

$$P = k * G$$

La multiplication d'un point d'une EC par 0 renvoie le point EC l'infini O.

Exemple 8.2

Soit le point G = (15, 13) sur la courbe elliptique sur le corps fini $\mathbf{F_{17}}$: $y^2 = x^3 + 7 \pmod{17}$ et k = 6. On a P = k * G = 6 * (15, 13) = (5, 8)

Le point générateur dans une EC

Pour les courbes elliptiques sur des corps finis, les cryptosystèmes ECC définissent un point EC prédéfini (constant) spécial appelé point générateur G (point de base), qui peut générer n'importe quel autre point de son sous-groupe sur la courbe elliptique en multipliant G par un nombre entier dans la plage [1...r]. Le nombre r est appelé "ordre" du sous-groupe cyclique (le nombre total de tous les points du sous-groupe).

Lorsque G et *n* sont soigneusement sélectionnés, tous les points EC possibles sur la courbe (y compris le point spécial infini) peuvent être générés à partir du générateur G en le multipliant par un entier dans la plage [1...*n*]. où *n* est "l'ordre de la courbe".

Les sous-groupes de courbes elliptiques ont généralement de nombreux points générateurs, mais les cryptographes en sélectionnent soigneusement un, qui génère le groupe entier (ou sous-groupe) et convient aux optimisations de performances dans les calculs. C'est le générateur connu sous le nom de "G".

On sait que pour certaines courbes, différents points générateurs génèrent des sous-groupes d'ordre différent. Cela signifie que certains points utilisés comme générateurs pour la même courbe généreront des sous-groupes plus petits que d'autres. Si le groupe est petit, la sécurité est faible.

Exemple 8.3

Dans l'exemple ci-dessus (l'EC sur le corps fini $y^2 \equiv x^3 + 7 \bmod 17$), si nous prenons le point G = (15, 13) comme générateur, tout autre point de la courbe peut être obtenu en multipliant G par un nombre entier dans la plage [0...17]. Ainsi l'ordre de cette EC est n = 18.

Notez également que si nous prenons le point (5, 9) comme générateur, il ne générera que 3 points EC : (5, 8), (5, 9) et l'infini.

Clé privée, clé publique et point générateur dans ECC

Dans l'ECC, lorsqu'on multiplie un point EC fixe G (le point générateur) par un certain entier *k* (*k* peut être considéré comme une clé privée), on obtient un point EC P (sa clé publique correspondante).

Par conséquent, dans ECC nous avons :

- E Courbe elliptique (EC) sur corps fini $\mathbb{F}_p$
- G = point générateur (constante fixe, un point de base sur l'EC)
- k = clé privée (entier)
- P = clé publique (point)

Il est très rapide de calculer P = k * G, en utilisant les algorithmes de multiplication ECC bien connus en temps au plus $klog_2 k$, par ex. l'algorithme « double et addition ». Pour les courbes 256 bits, il ne faudra que quelques centaines d'opérations EC simples.

Il est extrêmement lent (considéré comme infaisable pour les grands k) de calculer k = P/G.

1.3 Echage de clé Diffie–Hellman basé elliptique courbe ECDH (Diffie–Hellman Elliptic Curve)

ECDH est basé sur la propriété suivante des points d'une courbe elliptique :

$$(a * G) * b = (b * G) * a$$

Si nous avons deux nombres secrets *a* et *b* (deux clés privées, appartenant à Alice et Bob) et une courbe elliptique ECC avec le point générateur G, nous pouvons échanger sur un canal non sécurisé les valeurs (a * G) et (b * G) (les clés publiques d'Alice et de Bob) et ensuite nous pouvons dériver un secret partagé : secret = (a * G) * b = (b * G) * a. Assez simple. L'équation ci-dessus prend la forme suivante :

alicePubKey * bobPrivKey = bobPubKey * alicePrivKey = secret

L'algorithme ECDH

- Alice génère une paire de clés ECC aléatoire : {alicePrivKey, alicePubKey = alicePrivKey * G}
- Bob génère une paire de clés ECC aléatoire : {bobPrivKey, bobPubKey = bobPrivKey * G}
- Alice et Bob échangent leurs clés publiques via le canal non sécurisé (par exemple via Internet)
- Alice calcule sharedKey = bobPubKey * alicePrivKey
- Bob calcule sharedKey = alicePubKey * bobPrivKey
- Alice et Bob ont la même
  sharedKey = bobPubKey * alicePrivKey == alicePubKey * bobPrivKey

1.4 Sécurité

Pour une sécurité suffisante dans les cryptosystèmes basés sur le problème du logarithme discret on a besoin d'utiliser un grand nombre premier avec une taille d'au moins 2048 bits. L'arithmétique modulo avec de tels grands nombres premiers est lente (un coût de calcul) et augmente considérablement le coût de déploiement des cryptosystèmes qui utilisent ce type de groupe. Le groupe de points d'une courbe elliptique sur un champ fini offre le même niveau de sécurité en utilisant des nombres premiers avec une taille moins que les nombres choisis pour les autres cryptosystème ce qui réduit le coût de calcul. Les deux courbes les plus utilisées sont appelées P256 et Curve25519.

2. Chiffrement basé identité (IBE- Identite Based Encryption)

Le schéma fondé sur l'identité (ou identity-based) est une notion de cryptologie introduite par Adi Shamir en 1984. IBE permet la distribution efficace des clés dans un système multi-utilisateurs c'est un schéma dans lequel la clé publique de l'utilisateur est une information significative, telle qu'une adresse e-mail ou un identifiant de périphérique. Une autorité de confiance (Générateur de clé privée (PKG)) utilise une clé secrète principale pour calculer la clé secrète de l'utilisateur. Le chiffrement basé sur l'identité (IBE) a été introduit pour surmonter les problèmes de complexité associés aux approches traditionnelles basées sur PKI (Public Key Infrastructure).

IBE est spécifié par quatre algorithmes aléatoires : Installation, Extraction, Cryptage, Décryptage :

Installation : prend un paramètre de sécurité k et retourne params (paramètres système) et la clé principale (master-key). Les paramètres du système (params) comprennent une description d'un

espace de message fini *M*, et une description d'un espace de texte chiffré fini *C*. Les paramètres du système seront connus du public, tandis que la clé principale (master-key) ne sera connue que par le centre Générateur de Clé Privée (Private Key Generator- (PKG)).

Extraction : prend comme paramètres d'entrée, la clé principale (master-key) et un ID arbitraire ID $\epsilon \{0,1\}^*$, et renvoie une clé privée *d*. Ici, ID est une chaîne arbitraire qui sera utilisée comme clé publique et *d* est la clé de déchiffrement privée correspondante. L'algorithme Extraction extrait une clé privée de la clé publique donnée.

Cryptage : prend comme paramètres d'entrée, ID et M ϵ *M*. Il renvoie un texte chiffré C ϵ *C*.

Décryptage : prend comme paramètres d'entrée, C ϵ *C*, et la clé privée *d*. Il renvoie M ϵ *M*.

Ces algorithmes doivent satisfaire la contrainte de cohérence standard, à savoir lorsque *d* est la clé privée généré par l'algorithme Extraction lorsqu'on lui donne l'ID comme clé publique.

On a : $\forall$ M ϵ *M*: Decrypt(params;C; d) = M where C = Encrypt(params; ID;M)

Exemple 8.4

Pour expliquer les idées de base d' IBE , nous décrivons le schéma proposé par Dan Boneh et Matthew Franklin dans [C:\Users\nsc\Documents\divers\cryptanalyse\ Identity-BasedEncryptionfromtheWeilPairing.pdf] :

Installation : choisir <p, G, G_T, g, ê > ou *p* : un nombre premier, G et G_T deux groupe d'ordre p et une application bilinéaire (bilinear map) ê : GXG $\longrightarrow$ G_T et g un générateur de G.

Choisir $\alpha \epsilon Z^*_p$ et calculer $Q= g^{\alpha}$ (α est la clé principale- master-key de PKG)

Choisir deux fonctions de hachage : $H_1 :\{0,1\}^* \longrightarrow G$ et $H_2 : G_T \longrightarrow :\{0,1\}^n$

L'espace des messages $M = \{0,1\}^n$ et l'espace de messages chiffrés $C= GX:\{0,1\}^n$

Les paramètres du système Params=<p, n,G, G_T, g, ê,$Q= g^{\alpha}$, H_1, H_2 >

Extraction : pour un ID $\epsilon \{0,1\}^*$: (1) Calculer $Q_{id}= H_1(ID) \epsilon G$

(2) la clé secrète est *ks* = Q_{id}^{α} (α est la clé principale)

Cryptage : pour chiffrer un message M ϵ *M* avec la clé publique ID : Choisir $r \epsilon Z^*_p$ et calculer CT=(C_1,C_2) Tel que :

$C_1=g^r$ et $C_2=M+H_2(\hat{e}(Q_{id},Q)^r)$ c.a.d $(M+H_2(\hat{e}(H_1(ID), g^{\alpha})^r)$

Décryptage : pour déchiffrer CT on calcule :

$M= C_2+ H_2(\hat{e}(ks, C_1)$

Validation :

$M= M+H_2(\hat{e}(Q_{id},Q)^r) + H_2(\hat{e}(ks, C_1) = M+H_2(\hat{e}(H_1(ID), g^{\alpha})^r+ H_2(\hat{e}(H_1(ID)^{\alpha}, g^r))$

$M= M+H_2(\hat{e}(H_1(ID), g)^{\alpha r}+ H_2(\hat{e}(H_1(ID), g)^{\alpha r})=M$

3. Chiffrement basé attributs (ABE- Attributs Based Encryption)

3.1 Introduction

Sahai et Waters [6] ont introduit le concept de chiffrement basé sur les attributs (ABE) comme une étape pour développer des schémas de chiffrement à haute expressivité. Le chiffrement basé attributs (ABE) est un chiffrement à clé publique qui permet aux utilisateurs de chiffrer et de déchiffrer les messages en fonction des attributs de l'utilisateur (ex. : la profession qu'il exerce, le lieu où il habite, etc.). ABE prend les attributs comme clé publique et les associe au texte chiffré et à la clé secrète de l'utilisateur. C'est un moyen efficace de résoudre les problèmes dans les scénarios de contrôle d'accès. Dans ABE, l'attribut joue un rôle très important. L'attribut est utilisé comme politique d'accès pour contrôler l'accès des utilisateurs. Il est destiné au cryptage un-à-plusieurs c.à.d. Ceux qui sont capables de répondre à certaines exigences spécifiées peuvent décrypter.

Le chiffrement ABE est divisé en deux catégories : Ciphertext-Policy ABE (CP-ABE) et Key-Policy ABE (KP-ABE) en fonction de la politique d'accès intégrée dans le texte chiffré ou dans la clé secrète de l'utilisateur.

3.2 Les différentes entités dans ABE

les entités (Figure 8.4) qui peuvent intervenir dans un schéma ABE sont :

L'utilisateur

L'utilisateur est celui qui veut accéder aux données. Si un utilisateur possède un ensemble d'attributs satisfaisant la politique d'accès des données chiffrées, et n'est révoqué d'aucun des groupes d'attributs valides, alors il pourra déchiffrer le texte chiffré et obtenir les données.

Le propriétaire des données

Le propriétaire des données est la personne qui possède les données et qui souhaite de les charger dans le centre de stockage de données externe pour le partage. Le propriétaire des données est responsable de définir la politique d'accès et également c'est celui qui souhaite crypter les données afin d'assurer la sécurité.

Le centre de génération de clés (KGC- Key Generation Center)

C'est une autorité clé qui génère les paramètres publics et secrets. Il est chargé d'émettre, de révoquer et de mettre à jour les clés d'attribut pour les utilisateurs. Il exécutera honnêtement les tâches assignées dans le système.

Le centre de stockage de données

Le centre de stockage de données stocke les données des utilisateurs. Le service de partage de données est fourni par cette entité. Il assigne et révoque les attributs aux utilisateurs valides.

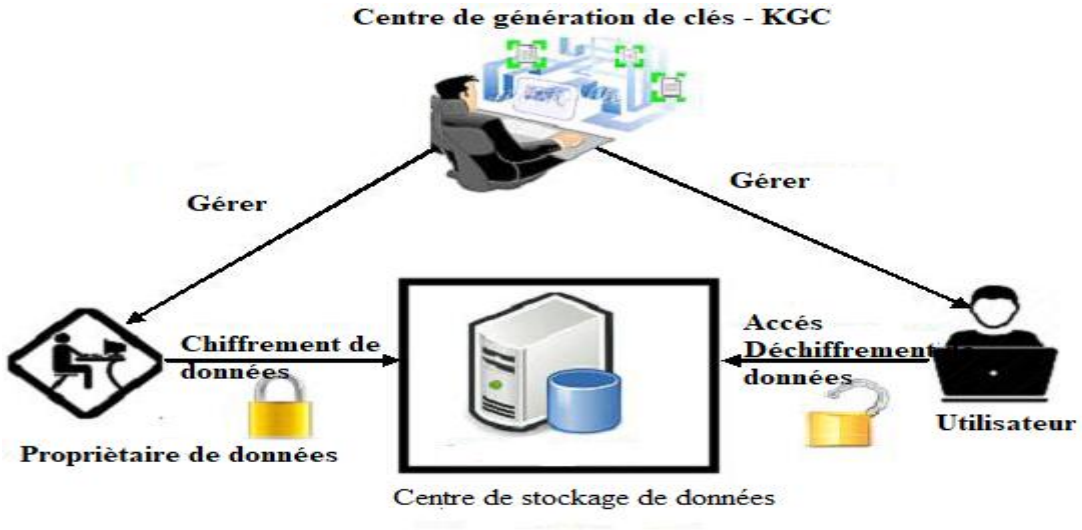


Figure 8.4 : Les entités de chiffrement ABE

3.3 Stratégies de schéma ABE

Le chiffrement ABE est divisé en deux catégories :

3.3.1 Ciphertext-Policy ABE (CP-ABE)

CP-ABE peut être considéré comme une généralisation du chiffrement basé sur l'identité (IBE). Ainsi, comme dans le chiffrement basé sur l'identité, il existe une seule clé publique et une clé privée principale qui peut être utilisée pour créer des clés privées. Chaque utilisateur est associé à un ensemble d'attributs et sa clé privée est générée sur la base de ces attributs. Lors du cryptage d'un message M, le crypteur spécifie une structure d'accès (politique) qui est exprimée en termes d'ensemble d'attributs sélectionnés pour M. Le message est crypté en fonction de la structure d'accès (politique) de telle sorte que seuls ceux dont les attributs satisfont à cette structure d'accès peuvent décrypter le message.

3.3.2 Key-Policy ABE (KP-ABE)

Dans KP-ABE, chaque texte chiffré est associé à un ensemble d'attributs descriptifs et la clé secrète de l'utilisateur est émise par une autorité de confiance capture la structure d'accès ou politique d'accès. Un utilisateur ne peut déchiffrer un texte chiffré que si l'ensemble d'attributs associé au texte chiffré satisfait la politique d'accès associée à sa clé privée. KP-ABE convient aux organisations structurées avec la règle : « qui peut lire des documents particuliers ».

3.3.3 Algorithme pour ABE

Les schémas ABE se composent généralement de quatre algorithmes fondamentaux :

Installation

Cet algorithme prend en entrée un paramètre de sécurité K et renvoie une clé publique PP et une clé secrète principale MSK. PP est utilisé par l'expéditeur de messages pour le cryptage. MSK est utilisé pour générer la clé secrète de l'utilisateur et n'est connu que de l'autorité.

Génération de clés

L'autorité exécute cet algorithme dans le but de générer une clé secrète SK. L'algorithme prend en entrée le paramètre public PP, la clé secrète principale MSK, l'ensemble d'attributs S et en sortie on obtient une clé de déchiffrement SK qui permet à l'utilisateur de déchiffrer un message chiffré à base d'attributs et politique d'accès.

Chiffrement

Cet algorithme est exécuté par un expéditeur qui souhaite chiffrer un message M, avec le paramètre public PP, un ensemble d'attributs S, une structure d'accès T. Cet algorithme délivre le texte chiffré CT.

Déchiffrement

Cet algorithme prend en entrée le texte chiffré CT et la clé secrète SK pour un ensemble d'attributs. il renvoie le message M si et seulement si CT satisfait la structure d'accès associée au texte chiffré CT.

4. Exercices

Exercice 8.1

Que signifié le schéma de chiffrement hybride ECIES en le présentant ?

Exercice 8.2

Présenter un schéma de chiffrement de données basé sur les courbes elliptiques ?

Bibliographie

[1] William Stallings, Cryptography and Network Security Principles and Practices, 4ième Edition, Edition Prentice Hall,2005.

[2] Jonathan Katz **&** Yehuda Lindell, Introduction to Modern Cryptography, Chapman & Hall/CRC Press, 2007.

[3] Douglas R. Stinson, Cryptography Theory And Practice, 3ième Edition, Chapman & Hall/CRC Press, 2006.

[4] Alfred J. Menezes, Paul C. van Oorscho t **&** Scott A. Vanstone, Handbook Of Applied Cryptography, CRC Press,2001.

[5] Boneh D., Franklin M. (2001) Identity-Based Encryption from the Weil Pairing. In: Kilian J. (eds) Advances in Cryptology — CRYPTO 2001. CRYPTO 2001. Lecture Notes in Computer Science, vol 2139. Springer, Berlin, Heidelberg. https://doi.org/10.1007/3-540-44647-8_13.

[6] Rajeev, A., Kenu, T.S., & Babu, D.S. (2016). A Survey on Attribute Based Encryption Schemes for Data Sharing, IOSR Journal of Computer Engineering (IOSR-JCE) e-ISSN: 2278-0661,p-ISSN: 2278-8727, PP 19-23.