

# Cryptographie et sécurité

Yves GERARD  
Université Lyon 1

February 9, 2009

# Contents

<b>1</b>	<b>Introduction</b>	<b>5</b>
1.1	Références de base . . . . .	5
<b>2</b>	<b>Cryptosystème</b>	<b>6</b>
2.1	Définition d'un cryptosystème . . . . .	6
2.2	Notations . . . . .	6
2.2.1	$e_K, d_K$ . . . . .	6
2.2.2	$e, d$ . . . . .	7
2.3	Cryptosystème symétrique . . . . .	7
2.4	Cryptosystème asymétrique . . . . .	7
<b>3</b>	<b>Exemple de cryptosystème symétrique: DES</b>	<b>8</b>
3.1	Data Encryption Standard . . . . .	8
3.2	Description détaillée . . . . .	8
3.3	Sécurité . . . . .	8
3.4	Triple DES (TDES) . . . . .	9
3.5	Advanced Encryption Standard (AES) . . . . .	10
<b>4</b>	<b>Exemple de cryptosystème asymétrique: RSA</b>	<b>11</b>
4.1	Outils mathématiques: calcul modulaire . . . . .	11
4.2	$(\mathbb{Z}/n\mathbb{Z})^\times$ . . . . .	11
4.2.1	$(\mathbb{Z}/n\mathbb{Z}, +, \cdot)$ . . . . .	11
4.2.2	$(\mathbb{Z}/n\mathbb{Z})^\times$ . . . . .	11
4.2.3	Corps $F_p$ . . . . .	12
4.2.4	Fonction d'Euler . . . . .	12
4.3	Ordre dans un groupe commutatif . . . . .	13
4.3.1	. . . . .	13
4.3.2	. . . . .	13

4.3.3	Théorème de Lagrange . . . . .	14
4.4	(Petit théorème de Fermat). . . . .	14
4.5	. . . . .	15
4.5.1	Calcul de la fonction d'Euler . . . . .	15
4.6	Euler et application à la cryptographie . . . . .	16
4.6.1	. . . . .	16
4.7	Méthode RSA . . . . .	16
4.7.1	Exemple . . . . .	17
4.8	Factorisation . . . . .	18
4.8.1	Méthode " $p - 1$ " de Pollard (1974) . . . . .	18
4.8.2	Algorithme de Pollard . . . . .	18
4.8.3	Grandes méthodes . . . . .	19
<b>5</b>	<b>Primalité</b> . . . . .	<b>20</b>
5.1	Densité . . . . .	20
5.2	Algorithme des puissances . . . . .	20
5.3	Test de Fermat . . . . .	21
5.4	$((\mathbb{Z}/p\mathbb{Z})^\times)^2$ . . . . .	22
5.5	Symbole de Legendre . . . . .	23
5.5.1	Définition . . . . .	23
5.5.2	. . . . .	23
5.5.3	Exemples . . . . .	23
5.6	Symbole de Jacobi . . . . .	23
5.6.1	Définition . . . . .	23
5.6.2	Résultats . . . . .	24
5.7	Test (critère) d'Euler . . . . .	24
5.7.1	Définition . . . . .	24
5.7.2	Exemples . . . . .	25
5.8	Test de Solovay-Strassen . . . . .	25
5.8.1	. . . . .	25
5.8.2	Test de base . . . . .	25
5.8.3	Test itéré . . . . .	25
5.8.4	Engendrement de premier . . . . .	26
<b>6</b>	<b>Signature</b> . . . . .	<b>27</b>
6.1	Propriétés de signature . . . . .	27
6.2	Exemple élémentaire avec clef publique . . . . .	27
6.3	Schema de signature . . . . .	28

6.4	Remarques . . . . .	28
6.5	ElGamal . . . . .	29
6.5.1	Logarithme discret. Fonction à sens unique . . . . .	29
6.5.2	Echange de clef. Diffie-Hellman . . . . .	30
6.5.3	Cryptosystème El Gamal . . . . .	30
6.5.4	Exemple de schema de signature. El Gamal asymétrique non déterministe . . . . .	31
6.5.5	Autres schema de signature . . . . .	32
<b>7</b>	<b>Hachage</b>	<b>33</b>
7.1	Signature, intégrité et hachage . . . . .	33
7.2	Propriétés . . . . .	34
7.3	Avec cryptosystème . . . . .	36
7.4	MD4 . . . . .	36
7.5	MD5 . . . . .	37
7.6	SHA-x . . . . .	37
7.7	Autres . . . . .	37
<b>8</b>	<b>Protocoles de partage de secret et d'authentification</b>	<b>38</b>
8.1	Objectif des protocoles . . . . .	38
8.2	Exemple de protocole avec tierce partie de confiance. . . . .	39
8.2.1	schema 1 . . . . .	39
8.2.2	schema 2 . . . . .	39
8.2.3	schema 3 . . . . .	40
8.2.4	schema 4: Needham-Schroeder . . . . .	40
8.2.5	schema 5: Needham-Schroeder-Denning-Sacco . . . . .	40
8.2.6	Variante de Needham-Schroeder sous Kerberos . . . . .	41
8.3	Types et degrés d'authentification . . . . .	41
8.3.1	Introduction . . . . .	41
8.3.2	Authentification de clef . . . . .	41
8.3.3	Authentification d'identité . . . . .	43
8.4	Typologie d'attaque . . . . .	44
8.4.1	Ecoute furtive . . . . .	44
8.4.2	Altération . . . . .	44
8.4.3	Rejeu . . . . .	44
8.4.4	Préjeu . . . . .	44
8.4.5	Réflexion . . . . .	44
8.4.6	Déni de service (Deny Of Service, DoS attack) . . . . .	45

8.4.7	Polysémie . . . . .	45
8.4.8	Cryptanalyse . . . . .	46
8.4.9	Manipulation de certificat . . . . .	46
8.4.10	Interaction de protocoles . . . . .	47
8.5	Exemples de protocoles . . . . .	47
8.5.1	Sécurité antérieure . . . . .	47
8.5.2	Protocole de transport de clef (“Forward Secrecy”) . . .	47
8.5.3	Diffie-Hellman . . . . .	48
8.5.4	Protocole d’établissement de clef avec tierce partie . . .	48
8.5.5	Protocole STS (Station To Station) . . . . .	49

# Chapter 1

## Introduction

### 1.1 Références de base

1. Revue française bimensuelle "MISC",  
Diamond
2. Handbook of Applied Cryptography, A. Menezes, P. van Oorschot, S.  
Vanstone, CRC Press 1997.  
[www.cacr.math.uwaterloo.ca/hac](http://www.cacr.math.uwaterloo.ca/hac)

# Chapter 2

## Cryptosystème

### 2.1 Définition d'un cryptosystème

Un *cryptosystème* est la donnée

- d'un ensemble fini  $\mathcal{M}$  de messages (textes) clairs
- d'un ensemble fini  $\mathcal{C}$  de cryptogrammes (messages cryptés)
- d'un ensemble fini  $\mathcal{K}$  de clefs

- d'une application d'encryptage

$$e : \mathcal{M} \times \mathcal{K} \rightarrow \mathcal{C}$$

et d'une application de décryptage

$$d : \mathcal{C} \times \mathcal{K} \rightarrow \mathcal{M}$$

telles que, pour tout  $M \in \mathcal{M}$  et tout  $K \in \mathcal{K}$

$$d(e(M, K), K) = M$$

### 2.2 Notations

#### 2.2.1 $e_K, d_K$

Pour toute clef  $K$ , on pose

$$e_K(M) = e(M, K)$$

$$d_K(C) = d(C, K)$$

et on a donc

$$d_K(e_K(M)) = M$$

pout tout message clair  $M$

### 2.2.2 $e, d$

Pour toute clef  $K$ , on peut aussi l'écrire sous la forme

$$K = (e, d)$$

$e$  étant la sous-clef servant à encrypter

$$M \mapsto e_K(M) = \{M\}_e$$

et  $d$  étant la sous-clef servant à décrypter

$$C \mapsto d_K(C) = \{C\}_d$$

et on a donc encore sous cette nouvelle forme

$$\{\{M\}_e\}_d = M$$

## 2.3 Cryptosystème symétrique

Si  $e = d$ , la clef  $K = e = d$  est dite *symétrique* de même que le cryptosystème et on dit aussi, dans ce cas, que la clef est *secrète* (ou bien *privée*).

## 2.4 Cryptosystème asymétrique

Si  $e$  est publique et  $d$  est privée, on dit que le système (ou la clef) est *asymétrique* ou encore que c'est un système à *clef publique*. Dans un tel système asymétrique, le calcul de  $e$  (resp.  $d$ ) en fonction de  $d$  (resp.  $e$ ) doit être infaisable pratiquement.



# Chapter 3

## Exemple de cryptosystème symétrique: DES

### 3.1 Data Encryption Standard

Dérivé de Lucifer (IBM, Horst Feistel), mis au point par la NSA (National Security Agency) en 1976 , approuvé en 1978 par le NBS (National Bureau of Standards, aujourd'hui appelé NIST - National Institute of Standards and Technology): norme FIPS-46-3.

Le DES a été normalisé par l'American National Standard Institute(ANSI) sous le nom de ANSI X3.92, plus connu sous la dénomination DEA (Data Encryption Algorithm).

### 3.2 Description détaillée

Chiffrement itératif par blocs de 64 bits selon un schéma de Feistel à 16 passes (rondes). Le déchiffrement se fait donc avec le même algorithme mais en prenant les clefs dans l'ordre inverse (de 16 à 1).

### 3.3 Sécurité

- Il résiste à la cryptanalyse différentielle (Eli Biham et Adi Shamir, 1993): analyse de quelques centaines de textes clairs bien choisis en observant les différences dans les cryptogrammes.

Les "S-boxes" ne sont pas linéaires et la méthode différentielle devient équivalente à la recherche exhaustive à partir des 16 passes (bien) choisies par la NSA (et moins bonne pour plus de passes).

- Il est vulnérable

à la cryptanalyse linéaire (Mitsuru Matsui, 1993): attaque à texte clair connu avec approximation linéaire du DES; complexité en  $2^{39}$  DES

aux attaques qui combinent la cryptanalyse linéaire et différentielle (Langford-Hellman, 1994) avec compromis temps-mémoire (réduction à une table de l'ordre du téraoctet).

aussi par calcul distribué (distributed.net): clef cassée en moins d'un jour.

Machines ad hoc imaginées:

- Diffie-Hellman (77): un million de "Very Large Scale Integration" calculant chacun un million de clefs DES par seconde; un jour de calcul; 20 millions de dollars 1977.

- M. Wiener (1993): 16 fois, en parallèle, 5767 VLSI calculant chacun  $5.10^7$  clefs DES par seconde; un jour et demi de calcul; 100.000 dollars 1993.

Machine construite:

Deep Crack (1998): 200 000 dollars, clef cassée en moins d'une semaine.

### 3.4 Triple DES (TDES)

Chaque clef  $K$  de DES définit une permutation de  $(F_2)^{64}$  et il y a  $2^{56}$  clefs (ou DES) possibles (complexité a priori égale au nombre de clefs).

En 1992, Campbell et Wiener ont démontré que le cardinal du groupe engendré par ces  $2^{56}$  permutations était supérieur ou égal à  $10^{2499}$ . Le triple DES ne se réduit donc pas au DES.

### 3.5 Advanced Encryption Standard (AES)

Le triple DES est encore utilisé mais il est remplacé aujourd'hui par l'AES plus rapide et considéré comme beaucoup plus fiable.

# Chapter 4

## Exemple de cryptosystème asymétrique: RSA

### 4.1 Outils mathématiques: calcul modulaire

#### 4.2 $(\mathbb{Z}/n\mathbb{Z})^\times$

Sauf mention contraire explicite,  $n$  désignera un entier naturel non nul.

##### 4.2.1 $(\mathbb{Z}/n\mathbb{Z}, +, \cdot)$

Si  $\bar{a} = a + n\mathbb{Z}$  et  $\bar{b} = b + n\mathbb{Z}$  sont deux éléments de  $\mathbb{Z}/n\mathbb{Z}$ , on peut poser

$$\bar{a} + \bar{b} = \overline{a + b}$$

$$\bar{a} \cdot \bar{b} = \overline{a \cdot b}$$

car cela ne dépend pas des représentants choisis dans les classes.  
Ainsi  $(\mathbb{Z}/n\mathbb{Z}, +, \cdot)$  est un anneau commutatif.

##### 4.2.2 $(\mathbb{Z}/n\mathbb{Z})^\times$

Pour deux entiers relatifs  $a$  et  $n$  on a (par l'algorithme d'Euclide)

$$\text{pgcd}(a, n) \in Za + Zn$$

$a$  et  $n$  sont donc étrangers (premiers entre eux,  $\text{pgcd}(a, n) = 1$ ) si (et seulement si) il existe deux entiers relatifs  $u$  et  $v$  tels que

$$1 = ua + vn$$

relation (de *Bezout*) qui peut s'écrire

$$ua \equiv 1 \pmod{n}$$

On voit donc la classe  $\bar{a} = a + nZ \in Z/nZ$  est inversible (pour la multiplication) si et seulement si  $n$  et  $a$  (ou un représentant quelconque de la classe  $a + nZ$ ) sont étrangers.

On note  $(Z/nZ)^\times$  le sous-ensemble de  $Z/nZ$  formé des éléments inversibles;  
 $(Z/nZ)^\times$  est un groupe pour la multiplication.

### 4.2.3 Corps $F_p$

Si  $p$  est un entier premier, tous les éléments non nuls de  $Z/pZ$  sont donc inversibles et on dit que  $Z/pZ$  est un *corps commutatif* noté  $F_p$ .

### 4.2.4 Fonction d'Euler

Pour  $n \in N^*$ , on pose

$$\phi(n) = \text{card}((Z/nZ)^\times)$$

La fonction  $\phi$  s'appelle la fonction (ou indicateur) d'Euler (**Leonhard Euler** (1707-1783)).

On a donc

$$\phi(1) = 1$$

$$\phi(p) = p - 1$$

si  $p$  est un entier premier

et, pour deux entiers étrangers (premiers entre eux)  $a$  et  $b$ ,

$$\phi(ab) = \phi(a)\phi(b)$$

## 4.3 Ordre dans un groupe commutatif

### 4.3.1

Soit  $(G, +)$  un groupe commutatif noté additivement, d'élément neutre noté 0 (resp.  $(G, \cdot)$  un groupe commutatif noté multiplicativement, d'élément neutre noté 1).

Si  $a \in G$  on pose

$$0a = 0$$

(resp.

$$a^0 = 1)$$

et

$$na = (n - 1)a + a$$

(resp.

$$a^n = a^{(n-1)} \cdot a)$$

pour  $n \in \mathbb{N}^*$ .

### 4.3.2

Si  $a$  engendre un groupe fini, l'implication, pour deux entiers  $i$  et  $j$

$$ja = ia \implies (j - i)a = 0$$

(resp.  $a^j = a^i \implies a^{(j-i)} = 1$ )

montre que le groupe engendré par  $a$  est *cyclique*

$$\langle a \rangle = \{0a = 0, a, \dots, (n - 1)a\}$$

(resp.  $\langle a \rangle = \{a^0 = 1, a, \dots, a^{(n-1)}\}$ )

où  $n \in \mathbb{N}$  est le plus petit entier  $k > 0$  (au sens de l'ordre naturel) tel que  $ka = 0$  (resp.  $a^k = 1$ )

Cet entier est appelé l'*ordre de a*, noté  $\text{ord}(a)$ ; c'est aussi le cardinal du sous-groupe  $\langle a \rangle$  engendré par  $a$  et on a

$$\{k \in \mathbb{N}^* \mid ka = 0\} = \text{ord}(a) \cdot \mathbb{N}^*$$

(resp.  $\{k \in \mathbb{N}^* \mid a^k = 1\} = \text{ord}(a) \cdot \mathbb{N}^*$ )

### 4.3.3 Théorème de Lagrange

Plus généralement pour un groupe fini (non nécessairement monogène), le cardinal du groupe s'appelle aussi l'*ordre* du groupe, noté  $\text{ord}(G)$ .

**Théorème 1 (Joseph Louis Lagrange (1736-1813)).**

*Si  $G$  est un groupe commutatif et si  $a \in G$ , alors l'ordre de  $a$  divise l'ordre de  $G$ .*

La relation  $R_a$  dans  $G$  en notation additive

$$x R_a y \iff x - y \in \langle a \rangle$$

est une relation d'équivalence et les classes d'équivalence  $x + \langle a \rangle$  ont toutes le même cardinal:  $\text{card}(\langle a \rangle)$  puisque

$$y \mapsto x + y$$

$$\langle a \rangle \longrightarrow x + \langle a \rangle$$

est une bijection.

## 4.4 (Petit théorème de Fermat).

**Théorème 2 (Pierre de Fermat (1601-1665)).**

*Si  $p$  est un entier premier, pour tout entier  $n$  on a*

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

Ce test de "primalité" pour un nombre  $p$  s'avère en général très efficace, mais il existe des nombres (de "Carmichael") qui rendent vraie la congruence de Fermat pour tout entier  $n$  mais qui ne sont pas premiers (le plus petit d'entre eux est 561).

## 4.5

- Théorème 3**
1. L'ordre de la classe  $a+nZ$  dans  $(Z/nZ, +)$  est  $n/\text{pgcd}(a, n)$ .
  2.  $\phi(n)$  est aussi égal au nombre de générateurs du groupe additif  $(Z/nZ, +)$ .
  3. Si  $p$  est premier et  $s \in N^*$  on a

$$p^s = \sum_{k=0}^s \phi(p^k)$$
$$\phi(p^s) = p^s - p^{s-1}$$

En effet, d'après le lemme d'Euclide, si  $a, b \in Z^*$  on a

$$ba \equiv 0 \pmod{n} \iff b \equiv 0 \pmod{n/\text{pgcd}(a, n)}$$

Ainsi, l'ordre de la classe  $a + nZ$  dans  $(Z/nZ, +)$  est  $n/\text{pgcd}(a, n)$ .

Enfin, en répartissant les éléments en classes d'éléments de même ordre, on obtient

$$p^s = \sum_{k=0}^s \phi(p^k) = p^{s-1} + \phi(p^s)$$

### 4.5.1 Calcul de la fonction d'Euler

En résumant ce qui précède on a

- 1.

$$\phi(1) = 1$$

2. Si  $a$  et  $b$  sont étrangers

$$\phi(ab) = \phi(a)\phi(b)$$

3. Si  $p$  est premier et  $s \in N^*$

$$\phi(p^s) = p^s - p^{s-1}$$



## 4.6 Euler et application à la cryptographie

### 4.6.1

La congruence de Fermat se généralise ainsi

**Théorème 4 (Euler).**

*Soient  $p$  et  $q$  deux entiers premiers distincts. Pour tout entier  $m$  et tout entier naturel  $k$ , on a*

$$m^{k\phi(pq)+1} \equiv m \pmod{pq}$$

En effet, d'après Fermat si  $m \not\equiv 0 \pmod{p}$ ,

$$m^{\phi(p)} \equiv 1 \pmod{p}$$

implique

$$m^{k\phi(p)\phi(q)} \equiv 1 \pmod{p}$$

et symétriquement si  $m \not\equiv 0 \pmod{q}$ ,

$$m^{k\phi(p)\phi(q)} \equiv 1 \pmod{q}$$

Puisque  $p$  et  $q$  sont étrangers, on a donc

$$m^{k\phi(p)\phi(q)} \equiv 1 \pmod{pq}$$

Enfin, si  $m \equiv 0 \pmod{p}$ , on a

$$m^{k\phi(p)\phi(q)+1} \equiv m \pmod{p}$$

puisque  $k\phi(p)\phi(q) + 1 \neq 0$ .

## 4.7 Méthode RSA

En 1978, **R.L. Rivest**, **A. Shamir** et **L. Adleman** ont proposé une méthode de cryptographie à clef publique dont le principe est le suivant

1. Soient  $p$  et  $q$  deux "grands" entiers premiers distincts et soit  $n = pq$

2. Soient  $d$  un "grand" entier étranger à  $\phi(n)$  et soit  $e \in [1, \phi(n)[$  l'entier défini par

$$de \equiv 1 \pmod{\phi(n)}$$

3. Appelons *clef privée* l'entier  $d$  et *clef publique* le couple  $(e, n)$ .
4. Un message, considéré après codage comme un entier, est au besoin décomposé en blocs formés d'entiers  $M \in [0, n]$ , puis chaque bloc  $M$  est alors crypté en un message  $c(M)$  par

$$c(M) \equiv M^e \pmod{n}$$

5. Pour décoder  $c(M)$ , on peut utiliser la clef privée puisque

$$c(M)^d \equiv M^{de} \equiv M \pmod{n}$$

Cette méthode a été basée sur l'idée que la complexité du décodage est équivalente à celle de la décomposition d'un grand nombre en nombre premiers au sens où trouver un algorithme "efficient" pour décoder RSA dans le cas général donnerait un algorithme efficient pour la factorisation.

### 4.7.1 Exemple

Voici l'exemple qu'ont proposé Rivest, Shamir et Adleman.

$$p = 47, q = 59, n = 2773, \phi(n) = 2668, d = 157$$

Avec un algorithme de type Euclide, on obtient

$$e = 17$$

En utilisant le codage

$$\text{espace} = 00, A = 01, B = 02, \dots, Z = 26$$

le texte

$$M = ITS\ ALL\ GREEK\ TO\ ME$$

devient, en blocs de longueur 4,

$$M = 0920\ 1900\ 0112\ 1200\ 0718\ 0505\ 1100\ 2015\ 0013\ 0500$$

Alors

$$c(0920) \equiv 920^{17} \equiv 948 \pmod{2773}$$

... le message codé global est alors

$M = 0948\ 2342\ 1084\ 1444\ 2663\ 2390\ 0778\ 0774\ 0219\ 1655$

et on vérifie que

$$948^{157} \equiv 920 \pmod{2773} \dots$$

## 4.8 Factorisation

### 4.8.1 Méthode " $p - 1$ " de Pollard (1974)

Soit  $n$  un entier non premier possédant un facteur premier  $p$  tel que  $p - 1$  n'ait que des facteurs premiers inférieurs ou égaux à une borne  $B$  assez petite (pour les calculs qui suivent).

En posant

$$k = \prod q^e$$

le produit étant pris sur les entiers naturels  $e$  et les entiers premiers  $q$  tels que  $q^e \leq B < q^{e+1}$ .

Par hypothèse on a

$$k \equiv 0 \pmod{p - 1}$$

Si  $a \not\equiv 0 \pmod{p}$  et si

$$\text{pgcd}(a^k - 1, n) < n$$

Alors on a trouvé un facteur strict de  $n$  (situation à éviter dans RSA).

### 4.8.2 Algorithme de Pollard

$n$  impair,  $B$

1.  $a=2$
2. pour  $j = 2$  à  $B$  faire  
 $a \leftarrow (a^j \bmod n)$
3.  $d = \text{pgcd}(a - 1, n)$

4. si  $1 < d < n$ :  $d$  facteur de  $n$   
 sinon échec  
 (Test avec  $a = 2$  et  $k$  remplacé par  $B!$ )  
 Complexité polynomiale si on choisit  $B = O(\ln(n)^i)$ ,  $i \in \mathbb{N}$  mais réussite faible, sinon méthode impraticable calculatoirement.

### 4.8.3 Grandes méthodes

1. Crible quadratique (Quadratic Sieve (QS), Pomerance, 1983)  
 complexité

$$O(\exp((1 + o(1))\sqrt{\ln(n) \cdot \ln \ln(n)}))$$

2. Crible du corps de nombres (Number Field Sieve (NFS), Pollard, 1988)  
 complexité

$$O(\exp((1,9229\dots + o(1)) \cdot (\ln(n))^{\frac{1}{3}} \cdot (\ln \ln(n))^{\frac{2}{3}}))$$

3. Courbe elliptique (Elliptic Curve Method (ECM) Lenstra, 1985)  
 complexité (en notant  $p$  le plus petit facteur premier de  $n$ )

$$O(\exp(\sqrt{(2 + o_p(1)) \ln(p) \cdot \ln \ln(p) (\ln(n))^2}))$$

# Chapter 5

## Primalité

### 5.1 Densité

Si on pose, pour  $x$  réel,

$$\pi(x) = \text{card}(\{p \mid p \text{ premier}, p \leq x\})$$

alors, en  $+\infty$ , on a

$$\frac{\pi(x)}{x} \sim \frac{1}{\ln(x)}$$

### 5.2 Algorithme des puissances

Pour calculer,

$$a^e \pmod{n}$$

( $a$ ,  $e$  et  $n$  entiers naturels)

on définit les applications

$$Q : x \mapsto x^2$$

et

$$M : x \mapsto ax$$

En écrivant  $b$  en base 2:

$$b = (b_k b_{k-1} b_{k-2} \dots b_0)_2$$

on définit une suite

$$S = (h_{k-1} h_{k-2} \dots h_0)$$

où  $h_i$  est égal à

$Q$  si  $b_i = 0$

$QM$  si  $b_i = 1$

En partant de la valeur  $a$ , et en calculant modulo  $n$ , on applique alors les opérateurs  $Q$  ou  $M$  en lisant de gauche à droite la suite  $S$ .

Exemple:

$$3^{340} \pmod{341} = 56$$

$$340 = (101010100)_2$$

$$S = (QQMQQMQQMQQ)$$

et

$$(3)^2 \pmod{341} = 9$$

$$(9)^2 \pmod{341} = 81$$

$$(81) * 3 \pmod{341} = 243$$

$$(243)^2 \pmod{341} = 56$$

$$(56)^2 \pmod{341} = 67$$

$$(67) * 3 \pmod{341} = 201$$

$$(201)^2 \pmod{341} = 163$$

$$(163)^2 \pmod{341} = 312$$

$$(312) * 3 \pmod{341} = 254$$

$$(254)^2 \pmod{341} = 67$$

$$(67)^2 \pmod{341} = 56$$

### 5.3 Test de Fermat

Soit  $n \in 1 + 2Z$  et soit  $a \in N^*$  premier avec  $n$ .

On dit que  $n$  est *pseudo(-Fermat)-premier en base  $a$* , si

$$a^{n-1} \equiv 1 \pmod{n}$$

Exemple:

341 est pseudo-premier en base 2 car

$$2^{340} \equiv (2^{10})^{34} \equiv 1 \pmod{341}$$

mais il ne l'est pas en base 3 car

$$3^{340} \equiv 56 \pmod{341}$$

Un nombre premier est évidemment pseudo-premier en toute base  $a$  possible ( $a$  et  $n$  étrangers), mais il existe une infinité de nombres non premiers, dits de Charmichaël, qui sont pourtant pseudo-premiers en toute base  $a$ .

**Théorème 5** *Si  $n$  est de la forme*

$$n = \prod_{j=1}^k p_j$$

*avec  $k \geq 2$  et où les  $p_j$  sont des entiers premiers distincts.*

*Alors  $n$  est de Charmichaël si et seulement si,*

*pour tout  $j \in [1, k]$ ,  $(p_j - 1)$  divise  $(n - 1)$*

On peut ainsi vérifier que  $561=3.11.17$  est de Charmichaël et c'est le plus petit.

Les suivants sont:  $1729=7.13.19$ ,  $8911=7.19.67$ ,  $10585=5.29.73$ ,  $294.409=37.73.109...$

## 5.4 $((\mathbb{Z}/p\mathbb{Z})^\times)^2$

Soit  $p$  un entier premier,  $p > 2$ .

On note  $((\mathbb{Z}/p\mathbb{Z})^\times)^2$  le groupe (sous-groupe multiplicatif de  $(\mathbb{Z}/p\mathbb{Z})^\times$ ) des éléments qui sont des carrés dans  $(\mathbb{Z}/p\mathbb{Z})^\times$

**Théorème 6**

$$(\mathbb{Z}/p\mathbb{Z})^\times)^2 = \{x \in F_p \mid x^{\frac{p-1}{2}} = 1\}$$

$$\text{card}(((\mathbb{Z}/p\mathbb{Z})^\times)^2) = \frac{p-1}{2}$$

## 5.5 Symbole de Legendre

### 5.5.1 Définition

Soit  $p$  un entier premier,  $p > 2$ .

Pour  $a \in Z$ , on définit le symbole de Legendre de la manière suivante

$$\left(\frac{a}{p}\right) = \begin{cases} 1 & \text{si } a + Zp \in ((Z/pZ)^\times)^2 \\ -1 & \text{si } a + Zp \in F_p^\times - ((Z/pZ)^\times)^2 \\ 0 & \text{si } a \in Zp \end{cases}$$

### 5.5.2

**Théorème 7** *Si  $p > 2$  est premier alors*

$$\left(\frac{a}{p}\right) = (a^{\frac{p-1}{2}} \bmod p)$$

### 5.5.3 Exemples

1.

$$\left(\frac{2}{11}\right) = (2^5 \bmod 11) = -1$$

Pour ce résultat, on peut aussi faire la table des carrés modulo 11 ou bien appliquer la formule ci-infra en 4.6.2.2

2.

$$\left(\frac{2}{31}\right) = (2^{15} \bmod 31) = 1$$

## 5.6 Symbole de Jacobi

### 5.6.1 Définition

Pour  $a \in Z$  et  $n \in 1 + 2Z$ , si  $n$  est de la forme

$$n = \prod_{j=1}^k p_j$$



où les  $p_j$  sont des entiers premiers,  
alors on définit le symbole de Jacobi de la manière suivante

$$\left(\frac{a}{n}\right) = \prod_{j=1}^k \left(\frac{a}{p_j}\right)$$

Le symbole de Jacobi est par définition (un morphisme) multiplicatif

$$\left(\frac{ab}{n}\right) = \left(\frac{a}{n}\right) \cdot \left(\frac{b}{n}\right)$$

et pour,  $m$  et  $n$  impairs, on a la loi de *réciprocité quadratique de Gauss*

$$\left(\frac{m}{n}\right) = (-1)^{\frac{m-1}{2} \cdot \frac{n-1}{2}} \left(\frac{n}{m}\right)$$

### 5.6.2 Résultats

1.

$$\left(\frac{-1}{n}\right) = (-1)^{\frac{n-1}{2}}$$

2.

$$\left(\frac{2}{n}\right) = (-1)^{\frac{n^2-1}{8}}$$

3. Si  $p > 2$  est premier

$$\left(\frac{2}{p}\right) = \begin{cases} 1 & \text{si } (p \bmod 8) = \pm 1 \\ -1 & \text{si } (p \bmod 8) = \pm 3 \end{cases}$$

4. Si  $p > 3$  est premier

$$\left(\frac{3}{p}\right) = \begin{cases} 1 & \text{si } (p \bmod 12) = \pm 1 \\ -1 & \text{si } (p \bmod 12) = \pm 5 \end{cases}$$

## 5.7 Test (critère) d'Euler

### 5.7.1 Définition

Soit  $n$  un entier impair.

On dit que  $n$  est (*pseudo-premier*) *eulérien* en base  $a$  si

$$\left(\frac{a}{n}\right) = (a^{\frac{n-1}{2}} \bmod n)$$

## 5.7.2 Exemples

$$(2^{10})^{17} \bmod 341 = 1^{17} \bmod 341 = 1$$

On en déduit a fortiori (et on l'a déjà vu) que 341 est pseudo-(Fermat)-premier en base 2.

Mais

$$\left(\frac{2}{341}\right) = \left(\frac{2}{11}\right) \cdot \left(\frac{2}{31}\right) = (-1) \cdot 1 = -1$$

et on voit ainsi que 341 n'est pas pseudo-eulérien en base 2.

## 5.8 Test de Solovay-Strassen

### 5.8.1

**Théorème 8** *Soit  $n$  un entier impair, non premier.*

*Alors, on a*

$$\text{card}(\{a \in [1, n-1] \mid n \text{ eulérien en base } a\}) \leq \frac{n-1}{2}$$

*En particulier, pour  $a$  choisi au hasard, la probabilité que  $n$  soit eulérien en base  $a$  est inférieure ou égale à  $\frac{1}{2}$ .*

### 5.8.2 Test de base

Soit  $n$  un entier impair fixé.

$a$  étant choisi "aléatoirement" dans l'intervalle  $]1, n-1]$ ,

si  $\left(\frac{a}{n}\right) = \left(a^{\frac{n-1}{2}} \bmod n\right)$ , alors  $n$  est probablement premier  
sinon  $n$  n'est pas premier.

(Test avec biais sur réponse positive ("Monte-Carlo")).

### 5.8.3 Test itéré

Répétant le test en faisant varier aléatoirement la base, si  $n$  est non premier alors la probabilité pour qu'il soit testé probablement premier  $k$  fois de suite est inférieure ou égale à  $\frac{1}{2^k}$ .

### 5.8.4 Engendrement de premier

**Théorème 9** Soit  $B \in \mathbb{N}$  une borne fixée.

Soit  $n \in \mathbb{N}$  choisi aléatoirement dans  $]1, B]$ . Si  $n$  est testé probablement premier  $k$  fois de suite, alors la probabilité pour que  $n$  ne soit pas premier est inférieure ou égale à

$$\frac{B}{\pi(n).2^k} \simeq \frac{\ln(B)}{2^k}$$

# Chapter 6

## Signature

### 6.1 Propriétés de signature

Une signature a pour but principal d'authentifier l'identité de l'auteur d'un message et donne en général une preuve de l'intégrité du message.

Elle peut associer le cryptage du message lui-même et, en introduisant un estampillage, procurer la propriété de non-révocation.

### 6.2 Exemple élémentaire avec clef publique

Alice, notée  $A$ , possède la clef

$$K_A = (e, d)$$

( $e$  publique,  $d$  privée).

Elle envoie à Bob, noté  $B$ , un message  $M$  qu'elle signe:

$$A \rightarrow B : (M, \text{sgn}_A(M))$$

avec la signature

$$\text{sgn}_A(M) = \{M\}_d$$

Bob vérifie la signature en testant la véracité de l'égalité

$$\{\{\{M\}_d\}_e\} = M$$

Si on pose

$$\text{ver}_e(M, \text{sgn}_A(M)) = \begin{cases} 1 & \text{si égalité vraie} \\ 0 & \text{sinon} \end{cases}$$

on peut généraliser ce schéma.

## 6.3 Schema de signature

Un *schema de signature* est la donnée

- d'un ensemble fini  $\mathcal{M}$  de messages
- d'un ensemble fini  $\mathcal{S}$  de signature de messages
- d'un ensemble fini  $\mathcal{K}$  de clefs
  - d'un ensemble fini  $\mathcal{AS}$  d'algorithmes de signature  $sgn : \mathcal{M} \rightarrow \mathcal{S}$
  - d'un ensemble fini  $\mathcal{AV}$  d'algorithmes de vérification

$$ver : \mathcal{M} \times \mathcal{S} \rightarrow \{0, 1\}$$

- et d'une application

$$\mathcal{K} \rightarrow \mathcal{AS} \times \mathcal{AV}$$

$$K \mapsto (sgn_K, ver_K)$$

telle que, pour tout  $M \in \mathcal{M}$  et tout  $S \in \mathcal{S}$

- 1.

$$ver_K(M, S) = \begin{cases} 1 & \text{si } S = sgn_K(M) \\ 0 & \text{sinon} \end{cases}$$

2.  $sgn_K$  soit privée et  $ver_K$  publique
3.  $ver_K(M, \cdot)^{-1}$  ne soit pas calculatoirement faisable (sans connaître  $sgn_K$ )

## 6.4 Remarques

1. Dans le cas symétrique,  $A$  et  $B$  doivent partager en secret la clef  $K$ .  
Dans le cas asymétrique  $K = (e, d)$ , une partie de la clef est connue :

$$ver_K = ver_e$$

(faiblesse par rapport au cas symétrique)

2. Dans le cas où Oscar, noté  $O$ , est "l'homme du milieu", il peut se substituer à  $A$ :

1.  $A \rightarrow O || B : (M, sgn_A(M))$

- 1?.  $O_D \rightarrow B : (M, sgn_D(M))$

3. Dans le cas (plus difficile) où Oscar peut engendrer aléatoirement un couple  $(M, S)$  tel que  $M$  ait du sens et tel que

$$ver_{K_O}(M, S) = 1$$

alors il peut se faire passer pour  $A$  auprès de  $B$  ("impersonnification") en lui envoyant  $(M, S)$  ("contrefaçon").

Dans le cas RSA, cela revient à calculer  $S$  comme racine  $d$ -ième de  $M$  modulo  $n$

4. Pour contrer le cas 2 ( $M$  non accessible par cryptage) et le cas 3 (contrer l'impersonnification), on peut modifier l'envoi

$$A \rightarrow B : \{(M, sgn_A(M))\}_{K_B}$$

si  $K_B$  est privée entre  $A$  et  $B$  ou bien

$$A \rightarrow B : \{(M, sgn_A(M))\}_{e_B}$$

si  $e_B$  est publique.

Mais pas la signature après

$$A \rightarrow B : (\{M\}_{K_B}, \{\{M\}_{K_B}\}_{K_A})$$

car, évidemment, on aurait l'attaque suivante

$$1. A \rightarrow O || B : (\{M\}_{K_B}, \{\{M\}_{K_B}\}_{K_A})$$

$$1'. O_D \rightarrow B : (\{M\}_{K_B}, \{\{M\}_{K_B}\}_{K_D})$$

## 6.5 ElGamal

### 6.5.1 Logarithme discret. Fonction à sens unique

Soit  $G$  un groupe cyclique (noté multiplicativement).

On appelle "exponentiation discrète", de base un générateur  $g$  du groupe  $G$ , l'application de  $\mathbb{Z}/\text{card}(G)\mathbb{Z}$  sur  $G$

$$\exp_g : i \mapsto g^i$$

qui est une bijection facile à calculer.

L'application réciproque, notée  $\log_g$ , est appelée "logarithme discret" de base

$g$ . En général,  $\log_g$  est difficile à calculer et on dit que  $\exp_g$  est à "sens unique".

Par exemple, avec  $p$  un entier premier et  $g$  un générateur du groupe cyclique  $G = (\mathbb{Z}/p\mathbb{Z})^\times$  la bijection de  $[0, p - 1[$  sur  $G$

$$\exp_g : i \mapsto g^i \pmod{p}$$

est à "sens unique" si  $p$  est "assez grand" et bien choisi .

### 6.5.2 Echange de clef. Diffie-Hellman

Base du protocole.

Soit  $G$  un groupe cyclique (noté multiplicativement) dans lequel le logarithme discret  $\log_g$ , de base un générateur  $g$  du groupe  $G$ , est difficile à calculer.

Alice choisit aléatoirement sa clef privée  $n_a \in ]1, \text{card}(G) - 1]$  et envoie à Bob sa clef publique  $K_A = (G, g, \alpha = g^{n_a})$

De même, Bob choisit aléatoirement sa clef privée  $n_b \in ]1, \text{card}(G) - 1]$  et envoie à Alice sa clef publique  $K_B = (G, g, \beta = g^{n_b})$ . La clef secrète commune est alors:

$$K_{AB} = g^{n_a n_b} = \alpha^{n_b} = \beta^{n_a}$$

Ce protocole est attaquable par l'homme du milieu (voir chapitre 8). Des attaques sur le logarithme discret existent dont la complexité est atteignable dans certains cas (Shank, Pohlig-Hellman...)

### 6.5.3 Cryptosystème El Gamal

Soit  $p$  un entier premier et  $g$  un générateur du groupe multiplicatif  $(\mathbb{Z}/p\mathbb{Z})^\times$  tels que l'application  $\exp_g$  soit à sens unique. Après un échange de clef  $K_{AB} = g^{n_a n_b}$  basé, comme précédemment, sur le protocole de Diffie-Hellman, un message de Bob peut-être considéré par codage comme un entier  $M \in ]1, p - 1[$ . Pour Alice, il est crypté par Bob en un message

$$MC = M.K_{AB} \pmod{p}$$

et décrypté par Alice

$$M \equiv MC.(K_{AB})^{-1} \equiv MC.\beta^{-n_a}$$

Avantage du système: multiplication modulaire et usage de tables précalculées; variation du cryptage  $MC$  pour un même message  $M$  à condition de changer de clef pour chaque message.

Faiblesse: si usage deux fois de la même clef  $K_{AB}$ . Si Oscar connaît le message en clair  $M_1$  et son crypté (attaque sur "message en clair", voir chapitre 8):

$$MC_1 = M_1 \cdot K_{AB}$$

ainsi qu'un autre message  $M_2$  crypté par la même clef:

$$MC_2 \equiv M_2 \cdot K_{AB}$$

il peut obtenir la clef

$$K_{AB} \equiv MC_1 \cdot (M_1)^{-1}.$$

et décrypter le message  $M_2$ .

#### 6.5.4 Exemple de schema de signature. El Gamal asymétrique non déterministe

Soit  $p$  un entier premier et  $g$  un générateur du groupe multiplicatif  $(\mathbb{Z}/p\mathbb{Z})^\times$  tels que l'application  $\exp_g$  soit à sens unique.

- Ensemble des messages  $\mathcal{M} = (\mathbb{Z}/p\mathbb{Z})^\times$
- Ensemble des signatures  $\mathcal{S} = (\mathbb{Z}/p\mathbb{Z})^\times \times \mathbb{Z}/(p-1)\mathbb{Z}$

Si Alice a pour clef privée  $n_a \in ]1, p-1[$ , a publié sa clef publique  $K_A = (G, g, \alpha = g^{n_a})$ , et si Alice veut signer de manière non déterministe un message  $M$ , elle choisit aléatoirement un entier  $m$  et calcule

$$\text{sign}_A(M) = (\gamma, \delta)$$

avec

$$\gamma = g^m \pmod{p}$$

et

$$\delta = (M - n_a \gamma) m^{-1} \pmod{p-1}$$

Le schema de signature est alors le suivant

$$\text{ver}_{K_A}(M, \text{sign}_A(M)) = \begin{cases} 1 & \text{si } \alpha^\gamma \gamma^\delta = g^M \pmod{p} \\ 0 & \text{sinon} \end{cases}$$



### 6.5.5 Autres schéma de signature

Amélioration d'El Gamal: "Digital Signature Scheme (DSS)"

"Lamport Signature Scheme (LSS)"

"One Time Signature Scheme (OTS)"...

Le schéma de signature peut être amélioré par rajout de protocoles:

- de "non désavouement" ("indéniable signature")
- d'estampillage
- d'élément de preuve en cas de contrefaçon ("Fail-stop signature")...

# Chapter 7

## Hachage

### 7.1 Signature, intégrité et hachage

Un schéma de signature authentifie un message et son émetteur et est donc a fortiori un schéma d'intégrité du message.

On définit alors un schéma d'intégrité comme un schéma de signature dans lequel:

- la signature, appelée *hachage* ou *empreinte*, est à valeurs dans un espace de longueur fixe (indépendamment de la longueur des messages, par exemple longueur 128 bits pour MD4 ou MD5)
- la clef n'est pas liée à une identité et est confondue avec l'algorithme du hachage qui doit être rapide en exécution.
- Le hachage possède la propriété de "résistance forte aux collisions" (voir ci-infra)
- La vérification pour un hachage  $h$  est donnée par

$$ver_h(M, H) = \begin{cases} 1 & \text{si } H = h(M) \\ 0 & \text{sinon} \end{cases}$$

$h(M)$  est appelé le *haché* ou *l'empreinte* du message  $M$

Un algorithme de signature pure (par exemple avec RSA) donne en général des calculs longs (lenteur) et produit une signature de même taille que le message; on applique donc, dans la pratique, un schéma de signature à de petits

messages (par exemple dans le Digital Signature Standard -DSS, 1994-: message de 160 bits, signature de 320 bits); pour ne pas introduire, dans le cas de longs messages, des problèmes de réassemblage et de contrefaçon, la signature est calculée sur un haché (résumé)  $h(M)$  du message  $M$ :

$$A \rightarrow B : (M, \text{sgn}_A(h(M)))$$

## 7.2 Propriétés

On appelle *collision* pour un hachage  $h$  la donnée de deux messages (distincts)  $M$  et  $M'$  ayant même empreinte:

$$h(M) = h(M')$$

Un hachage  $h$  est dit *cryptographique sur un ensemble de messages*  $\mathcal{M}$  s'il vérifie les propriétés suivantes

**P1.** *Résistance à la première préimage sur*  $\mathcal{M}$  *ou fonction à sens unique ("One-Way") sur*  $\mathcal{M}$

Pour toute valeur

$$H \in \text{Im}(h) = h(\mathcal{M})$$

dont on ne connaît pas d'antécédent, la "résistance à la première préimage" consiste en la difficulté calculatoire pour trouver un antécédent  $M \in \mathcal{M}$  de  $H$  par  $h$  (i.e.  $h(M) = H$ )

Si  $h$  n'est pas une fonction à sens unique, du point de vue théorique, il existe, par exemple, une attaque sur un schéma de signature combinant une signature pure non déterministe (exemple: schéma ElGamal avec clef publique à paramètre aléatoire) et une fonction de hachage  $h$ .

Oscar recherche aléatoirement une signature pure valide

$$(M, \text{sgn}_A(M))$$

telles que  $\text{sgn}_A(M) \in \text{Im}(h)$ ;

si  $h(M') = M$  il fabrique alors un message signé par Alice

$$(M', \text{sgn}_A(M)) = (M', \text{sgn}_A(h(M')))$$

(Du point de vue pratique, il faut que  $M'$  ait du sens...)

**P2.** *Résistance à la seconde préimage sur  $\mathcal{M}$  ou résistance forte aux collisions sur  $\mathcal{M}$*

Pour toute valeur

$$H \in \text{Im}(h) = h(\mathcal{M})$$

dont on connaît un antécédent  $M \in \mathcal{M}$  par  $h$  ( $h(M) = H$ ), la "résistance à la seconde préimage" consiste en la difficulté calculatoire pour trouver un autre antécédent  $M' \in \mathcal{M}$  distinct de  $M$ .

Dans ce cas, le hachage définit un schéma d'intégrité sur  $\mathcal{M}$ :

$$M \mapsto (M, h(M))$$

Sinon le test d'intégrité est en défaut sur  $\mathcal{M}$  et, par exemple, Oscar peut même modifier un message  $M \in \mathcal{M}$  signé à l'aide d'une fonction de hachage car, évidemment, on aurait l'attaque suivante

1.  $A \rightarrow O \parallel B : (M, \text{sgn}_A(h(M)))$

1'.  $O_A \rightarrow B : (M', \text{sgn}_A(h(M)))$

avec  $h(M') = h(M)$  et  $M' \neq M$ .

**P3.** *Résistance aux collisions sur  $\mathcal{M}$*

Difficulté calculatoire de trouver une collision avec deux messages distincts appartenant à  $\mathcal{M}$ .

Sinon, par exemple, Oscar peut fabriquer un message  $M$  et le faire signer par Alice suivant l'attaque:

1.  $O \rightarrow A : M'$

2.  $A \rightarrow O : (M', \text{sgn}_A(h(M')))$

3.  $O \rightarrow B : (M, \text{sgn}_A(h(M')))$

avec  $h(M') = h(M)$

$P2$  implique  $P3$

## 7.3 Avec cryptosystème

Un message  $M$  est scindé blocs consécutifs de longueur  $n$  ( $n \geq 128$ , "attaque anniversaire")

$$M = M_1 | M_2 | \dots | M_k$$

On calcule à partir d'une valeur initiale  $x_0$

$$x_i = f(x_{i-1}, M_i)$$

où  $f$  incorpore un cryptosystème et le hachage  $h$  est alors défini par

$$h(M) = x_k$$

Exemples d'applications  $f$  réputées sûres (certaines ne le sont pas indépendamment du cryptosystème)

1.

$$x_i = e_{x_{i-1}}(M_i) \oplus M_i$$

2.

$$x_i = e_{x_{i-1}}(M_i) \oplus M_i \oplus x_{i-1}$$

3.

$$x_i = e_{x_{i-1}}(M_i \oplus x_{i-1}) \oplus M_i$$

4.

$$x_i = e_{x_{i-1}}(M_i \oplus x_{i-1}) \oplus M_i \oplus x_{i-1}$$

## 7.4 MD4

Défini par Ronald Rivest (MIT, 1990, voir RFC 1320).

En trois rondes, à base d'opérations logiques bit à bit, d'addition entière modulo  $2^{32}$  et de permutation circulaires. Taille du haché: 128 bits.

Aujourd'hui inutilisé: failles dues à Den Boer et Bosselaers, Hans Dobbertin, Wang et al.

Collisions générées avec un nombre d'opérations de l'ordre de  $2^8$  opérations (trop faible par rapport aux  $2^{64}$  nécessaires pour un paradoxe des anniversaires: probabilité de collision très proche de 1)

## 7.5 MD5

Amélioration du MD4. RFC 1321. Utilisé sous Unix (stockage de mot de passe avec md5sum). Pas assez sûr, car actuellement, algorithme de Vlastimil Klima, collision en moins d'une minute sur un PC et nombreuses études en cours.

## 7.6 SHA-x

SHA-0 est cassé; SHA-1 n'est plus très sûr. On suggère maintenant d'utiliser plutôt des algorithmes tels que SHA-256, SHA-384 ou SHA-512.

## 7.7 Autres

RIPMD-160 ou Whirlpool ...encore recommandés.

# Chapter 8

## Protocoles de partage de secret et d'authentification

### 8.1 Objectif des protocoles

De manière générale, on considèrera le schéma suivant.

Alice (notée  $A$ , symbolisant une personne ou bien un groupe d'entités formé de personnes ou d'interfaces matérielles) veut contacter Bob (noté  $B$ ) pour engager une communication (session) avec lui sur un réseau non sécurisé.

Le protocole choisi doit tendre vers une authentification mutuelle des identités:

- Alice doit être certaine qu'elle communiquera vraiment avec Bob (et non avec un usurpateur d'identité)

- De même, Bob doit être certain qu'il communiquera vraiment avec Alice

De plus, le protocole peut aussi attribuer d'autres propriétés à la communication comme, particulièrement, la confidentialité (la communication n'est pas lisible par une personne autre que  $A$  ou  $B$ ).

Le protocole peut introduire une tierce partie de confiance ("trusted server") notée  $T$  pour l'établissement de la communication.

## 8.2 Exemple de protocole avec tierce partie de confiance.

Le but de ce protocole (Needham-Schroeder, 1978) est l'établissement d'une clef de session symétrique  $K_{AB}$  pour  $A$  et  $B$ , l'accent étant mis sur la fraîcheur de cette clef pour garantir la qualité de cette clef privée qui doit être possédée seulement par  $A$  et  $B$  pendant une session au plus. Contruisons le par étapes en l'améliorant en fonction des attaques possibles sur le protocole lui-même, en supposant que  $T$  est bien certifié et que l'intégrité, la confidentialité des messages sont bien assurées cryptographiquement.

L'attaquant est nommé Oscar (noté  $O$  ou  $O_X$  s'il usurpe l'identité de  $X$ ).

On note  $\{M\}_K$  le message  $M$  crypté par l'usage d'une clef symétrique  $K$ .

### 8.2.1 schema 1

1.  $A \rightarrow T : A, B$
2.  $T \rightarrow A : K_{AB}$
3.  $A \rightarrow B : K_{AB}, A$

Attaque par écoute passive.

### 8.2.2 schema 2

1.  $A \rightarrow T : A, B$
2.  $T \rightarrow A : \{K_{AB}\}_{K_{AT}}, \{K_{AB}\}_{K_{BT}}$
3.  $A \rightarrow B : \{K_{AB}\}_{K_{BT}}, A$

Attaque active (faible): "homme du milieu" entre  $A$  et  $B$  à la passe 3 par interception du message et rejeu sous une identité  $D$  choisie:

- 3'.  $A \rightarrow O||B : \{K_{AB}\}_{K_{BT}}, A$
4.  $O_D \rightarrow B : \{K_{AB}\}_{K_{BT}}, D$

Autre attaque active si Oscar fait partie du réseau de distribution de clefs auprès de  $T$ : "homme du milieu" entre  $A$  et  $T$  à la passe 1 par interception du message et isolement de  $B$



### 8.2.3 schema 3

1.  $A \rightarrow T : A, B$
2.  $T \rightarrow A : \{K_{AB}, B\}_{K_{AT}}, \{K_{AB}, A\}_{K_{BT}}$
3.  $A \rightarrow B : \{K_{AB}, A\}_{K_{BT}}$

Attaque possible par rejeu.

### 8.2.4 schema 4: Needham-Schroeder

Challenge anti-rejeu avec usage d'un NOMbre aléatoire utilisé Une Seule fois "nomus" (en anglais "nonce" pour "Number used ONLY onCE") servant d'horodateur:

1.  $A \rightarrow T : A, B, n_A$
2.  $T \rightarrow A : \{K_{AB}, B, n_A, \{K_{AB}, A\}_{K_{BT}}\}_{K_{AT}}$
3.  $A \rightarrow B : \{K_{AB}, A\}_{K_{BT}}$
4.  $B \rightarrow A : \{n_B\}_{K_{AB}}$
5.  $A \rightarrow B : \{n_B - 1\}_{K_{AB}}$

Challenge anti-rejeu imparfait: attaques encore possibles (Denning-Sacco, 1981) par rejeu de clef ou de message.

### 8.2.5 schema 5: Needham-Schroeder-Denning-Sacco

(Authentication d'une "bonne" clef de session)

1.  $B \rightarrow A : B, n_B$
2.  $A \rightarrow T : A, B, n_A, n_B$
3.  $T \rightarrow A : \{K_{AB}, B, n_A\}_{K_{AT}}, \{K_{AB}, A, n_B\}_{K_{BT}}$
4.  $A \rightarrow B : \{K_{AB}, A, n_B\}_{K_{BT}}$

## 8.2.6 Variante de Needham-Schroeder sous Kerberos

Basé sur le shema:

1.  $A \rightarrow T : A, B, n_A$
2.  $T \rightarrow A :$   
 $\{K_{AB}, B, n_A, t_{AB} = \{K_{AB}, A\}_{K_{BT}}\}_{K_{AT}}$
3.  $A \rightarrow B : \{n_A\}_{K_{AB}}, t_{AB}$
4.  $B \rightarrow A : \{n_A - 1, n_B\}_{K_{AB}}$
5.  $A \rightarrow B : \{n_B - 1\}_{K_{AB}}$

$t_{AB}$  étant le ticket de dialogue entre  $A$  et  $B$

## 8.3 Types et degrés d'authentification

### 8.3.1 Introduction

Même si le but est une authentification mutuelle, le protocole se déroule de manière disymétrique dans la mesure où la première passe est à l'initiative d'un des deux protagonistes. De plus, même si le but est d'authentifier une entité correspondante, la plupart des protocoles ne fournissent qu'une preuve indirecte de l'entité en communication en authentifiant un élément caractéristique de celle-ci. Par exemple, dans le protocole de Needham-Schroeder,  $A$  obtient une clef fraîche exclusivement utilisable avec  $B$  mais  $B$  n'a aucune certitude sur l'identité de  $A$  et il n'y a donc pas de complète authentification mutuelle.

De manière générale, on va définir des types et degrés d'authentification.

### 8.3.2 Authentification de clef

#### 1. Fraîcheur de clef

On dira de manière concise que

*la clef de  $A$  est fraîche pour  $B$*

si  $A$  possède une clef pour communiquer avec  $B$

et si  $A$  sait que cette clef est fraîche i.e. créée récemment pour servir

de clef symétrique de session avec  $B$  ou de clef publique éphémère de  $A$  pour  $B$  dans la session avec  $B$

## 2. **Exclusivité de clef**

On dira de manière concise que  
*la clef de  $A$  est exclusive pour  $B$*   
si  $A$  possède une clef pour communiquer avec  $B$   
et si  $A$  sait que

**cas 1:** à part lui même, sa clef de session n'est connue, au plus, que de  $B$  (et peut-être d'une tierce partie de confiance)

**cas 2:** à part lui même, sa clef publique n'est connue, au plus, que de  $B$  (et peut-être d'une tierce partie de confiance)

## 3. **Bonne clef**

On dira de manière concise que  
*la clef de  $A$  est bonne pour  $B$*   
si  $A$  possède une clef qui est fraîche et exclusive pour communiquer avec  $B$

## 4. **Confirmation de clef**

On dira de manière concise que  
*la clef de  $A$  est confirmée pour  $B$*   
si  $A$  possède une clef qui est bonne pour (communiquer avec)  $B$  (sans la lier à une session particulière avec  $B$ )  
et si  $A$  sait, par preuve récente, que  $B$  la possède.

## 5. **Croyance mutuelle en la clef de $A$ pour $B$**

On dira de manière concise qu'il y a  
*croyance mutuelle en la clef de  $A$  pour  $B$*   
si  $A$  possède une clef qui est confirmée pour communiquer avec  $B$   
et si  $A$  sait, par preuve récente, que  $B$  sait que la clef est bonne pour (communiquer avec)  $A$

### 8.3.3 Authentification d'identité

#### 1. Authentification de $B$ pour $A$

On dira de manière concise qu'il y a *authentification de  $B$  pour (ou par)  $A$*  si  $A$  possède une preuve récente de l'identité de  $B$  (au sens du protocole, basée sur une clef pouvant servir de clef de session ou d'initiation du protocole) et de son engagement actif dans le protocole.

Exemple de challenge à base de signature "sign":

#### Exemple E1

1.  $A \rightarrow B : n_A$
2.  $B \rightarrow A : \text{sign}_B(n_A)$

#### 2. Authentification semi-forte de $B$ pour $A$

On dira de manière concise qu'il y a *authentification semi-forte de  $B$  pour  $A$*  s'il y a authentification de  $B$  pour  $A$  et si  $A$  possède une preuve récente que  $B$  sait que  $A$  est bien son interlocuteur engagé dans le protocole.

L'exemple E1 est une authentification non semi-forte car vulnérable dans une attaque par réflexion faite par l'homme du milieu:

#### Exemple E1'

1.  $A \rightarrow O || B : n_A$
- 1'.  $O \rightarrow B : n_A$
- 2'.  $B \rightarrow O : \text{sign}_B(n_A)$
2.  $O_B \rightarrow A : \text{sign}_B(n_A)$

car  $B$  ne connaît pas  $A$  (et  $A$  croit avoir affaire à  $B$ ).

#### 3. Authentification forte de $B$ pour $A$

On dira de manière concise qu'il y a *authentification forte de  $B$  pour  $A$*  s'il y a authentification de  $B$  pour  $A$  et si  $A$  possède une preuve récente que  $B$  sait qu'il a été authentifié par  $A$

Exemple d'authentification forte:

### Exemple E2

1.  $A \rightarrow B : n_A$
2.  $B \rightarrow A : \text{sign}_B(A, n_A)$

#### 4. Authentification mutuelle

Un type d'authentification mutuelle est un type qui s'applique autant à  $A$  qu'à  $B$

## 8.4 Typologie d'attaque

### 8.4.1 Ecoute furtive

(Attaque passive)

### 8.4.2 Altération

### 8.4.3 Rejeu

Augmentation d'information pour l'analyse de l'attaquant ou usage d'une ancienne clef cassée (Voir exemple Denning-Sacco)

### 8.4.4 Préjeu

But et technique analogue au rejeu mais avec initiative à l'attaquant.

### 8.4.5 Réflexion

Exemple d'authentification par connaissance commune d'une clef secrète:

1.  $A \rightarrow B : \{N_A\}_K$
2.  $B \rightarrow A : \{N_B\}_K, N_A$
3.  $A \rightarrow B : N_B$

attaquable par réflexion:

1.  $A \rightarrow O || B : \{N_A\}_K$

- 1'.  $O_B \rightarrow A : \{N_A\}_K$
- 2'.  $A \rightarrow O||B : \{N'_A\}_K, N_A$
- 2.  $O_B \rightarrow A : \{N'_A\}_K, N_A$
- 3.  $A \rightarrow O||B : N'_A$
- 3'.  $O_B \rightarrow A : N'_A$

### 8.4.6 Déni de service (Deny Of Service, DoS attack)

Mise hors jeu de l'attaqué par détournement ou dépassement de sa capacité.

### 8.4.7 Polysémie

Exemple du protocole Otway-Rees garantissant aux deux parties la fraîcheur de la clef:

- 1.  $A \rightarrow B : M, A, B, \{n_A, M, A, B\}_{K_{AT}}$
- 2.  $B \rightarrow T : M, A, B, \{n_A, M, A, B\}_{K_{AT}}, \{n_B, M, A, B\}_{K_{BT}}$
- 3.  $T \rightarrow B : M, \{n_A, K_{AB}\}_{K_{AT}}, \{n_B, K_{AB}\}_{K_{BT}}$
- 4.  $B \rightarrow A : M, \{n_A, K_{AB}\}_{K_{AT}}$

et, si

$$\text{longueur}(K_{AB}) = \text{longueur}(M, A, B)$$

attaque :

- 1.  $A \rightarrow O||B : M, A, B, \{n_A, M, A, B\}_{K_{AT}}$
- 4'.  $O_B \rightarrow A : M, \{n_A, M, A, B\}_{K_{AT}}$

amenant Alice à prendre pour clef la valeur  $(M, A, B)$  connue d'Oscar (qui a usurpé l'identité de Bob).

### 8.4.8 Cryptanalyse

Décryptage de clef ou de données.

On distingue les types suivants d'attaque

1. à textes cryptés seuls  
sans autre information sur ces textes
2. à textes cryptés choisis  
pour lesquels on peut retrouver une partie des textes en clair ou avoir des informations sur eux
3. à textes choisis adaptés  
avec dépendance des messages cryptés successifs pour aider au décryptage
4. à textes clairs choisis  
avec connaissance de textes clairs et de leur cryptage, par exemple pour en obtenir la clef de cryptage et décrypter d'autres messages utilisant cette clef
5. à textes clairs choisis adaptés  
en rajoutant au cas précédent une dépendance de messages successifs
6. à textes clairs connus  
en faisant crypter par le crypto système un message connu pour trouver la clef par comparaison entre le message et son crypté.

### 8.4.9 Manipulation de certificat

Modification ou appropriation de certificat.

Exemple: protocole MTI (Matsumoto-Takashima-Imai) de certification entre  $A$  et  $B$ .

Soit  $G$  un groupe cyclique dans lequel l'exponentiation discrète de base un générateur  $g$  du groupe  $G$  est une fonction "à sens unique". Le protocole

1.  $A \rightarrow B : g^{n_A}, Cert(A) = (g^a, A)$

2.  $B \rightarrow A : g^{n_B}, Cert(B) = (g^b, B)$

a pour but d'établir une clef privée commune entre  $A$  et  $B$ :

$$K_{AB} = g^{a.n_B + b.n_A}$$

Après l'attaque:

1.  $A \rightarrow O || B : g^{n_A}, Cert(A)$

1'.  $O \rightarrow B : g^{n_A}, Cert(O)$

2'.  $B \rightarrow O : g^{n_B}, Cert(B)$

2.  $O_B \rightarrow A : g^{c.n_B}, Cert(B)$

$A$  et  $B$  possède une même clef (qui n'est pas bonne)

$$K_{AB} = g^{c.a.n_B + b.n_A}$$

car  $A$  croit que sa clef n'est connue que de  $B$  et de lui-même, alors que  $B$  croit posséder une clef privée avec seulement  $O$ .

Nécessité de validation de clef par challenge avant certification (par exemple: "zero-knowledge"...).

## 8.4.10 Interaction de protocoles

Usage d'une même clef dans deux protocoles enchâssés.

## 8.5 Exemples de protocoles

### 8.5.1 Sécurité antérieure

Certains protocoles peuvent être renforcés pour assurer la sécurité de précédents messages avant une compromission de clef. Par exemple:

### 8.5.2 Protocole de transport de clef ("Forward Secrecy")

Si  $K_t$  est une clef publique éphémère de  $A$  et  $h$  une fonction de hachage:

1.  $A \rightarrow B : K_t, n_A, sign_A(K_t, n_A)$

2.  $B \rightarrow A : [K_{AB}]_{K_t}, sign_B(h(K_{AB}), A, n_A)$



### 8.5.3 Diffie-Hellman

On rappelle qu'on choisit un groupe cyclique  $G$  (noté multiplicativement) dans lequel le logarithme discret  $\log_g$ , de base un générateur  $g$  du groupe  $G$ , est difficile à calculer.

Alice a pour clef privée  $n_a \in ]1, \text{card}(G) - 1]$  et pour clef publique  $K_A = (G, g, \alpha = g^{n_a})$

De même, Bob a pour clef privée  $n_b \in ]1, \text{card}(G) - 1]$  et pour clef publique  $K_B = (G, g, \beta = g^{n_b})$ . La clef secrète commune est alors:

$$K_{AB} = g^{n_a n_b} = \alpha^{n_b} = \beta^{n_a}$$

Ce protocole est attaquable par l'homme du milieu:

1.  $A \rightarrow O || B : A, \alpha$

1'.  $O_A \rightarrow B : A, \gamma = g^{n_o}$

2'.  $B \rightarrow O || A : B, \beta$

2.  $O_B \rightarrow A : B, \gamma = g^{n_o}$

car Oscar partage une clef avec Alice et une autre clef avec Bob.

Cette base est en général modifiée par l'usage d'une authentification accompagnant la délivrance de clef publique avec des envois du type:

1.  $A \rightarrow B : A, \alpha$

2.  $B \rightarrow A : B, \beta, \text{sign}_B(\beta, \alpha)$

### 8.5.4 Protocole d'établissement de clef avec tierce partie

1.  $A \rightarrow T : A, B$

2.  $A \rightarrow B : A, g^{n_A}$

3.  $T \rightarrow B : \{A, B, K_T\}_{K_{BT}}$

4.  $T \rightarrow A : \{A, B, K_T\}_{K_{AT}}$

5.  $B \rightarrow A : A, g^{n_B}$

obtenant ainsi une clef de session  $g^{n_A \cdot n_B \cdot K_T}$  suivi de la destruction des noms et des clefs éphémères de  $A$

### 8.5.5 Protocole STS (Station To Station)

(Diffie-Oorschot-Wiener)

Pour  $g$  une base d'un logarithme discret et  $f$  une fonction définie hors protocole et connue de  $A$  et  $B$ :

1.  $A \rightarrow B : g^{n_A}$
2.  $B \rightarrow A : g^{n_B}, \{sign_B(g^{n_A}, g^{n_B})\}_{K_{AB}}$
3.  $B \rightarrow A : \{sign_A(g^{n_A}, g^{n_B})\}_{K_{AB}}$

où  $K_{AB} = f(g^{n_A \cdot n_B})$  et  $g^{n_A \cdot n_B}$  est la nouvelle clef fraîche de session, confirmée par l'usage de la clef temporaire  $K_{AB}$ .

C'est une authentification de  $B$  par  $A$  qui n'est pas forte car vulnérable à une attaque à base de réflexion entreprise par l'homme du milieu:

1.  $A \rightarrow O||B : g^{n_A}$
- 1'.  $O \rightarrow B : g^{n_A}$
- 2'.  $B \rightarrow O : g^{n_B}, \{sign_B(g^{n_A}, g^{n_B})\}_{K_{AO}}$
2.  $O_B \rightarrow A : g^{n_B}, \{sign_B(g^{n_A}, g^{n_B})\}_{K_{AB}}$
3.  $A \rightarrow O||B : \{sign_A(g^{n_A}, g^{n_B})\}_{K_{AB}}$

car  $K_{AO} = K_{AB}$

Du point de vue de la clef: la clef de  $A$  est bonne et a été confirmée mais il n'y a pas de croyance mutuelle.