

Notes de cours de CRYPTOLOGIE

rédigées par Julien Audebert et Lionel Rivière

16 avril 2010

Table des matières

1	Introduction	3
1.1	Chiffrement	3
1.2	Système de Chiffrement (Chiffre)	3
1.2.1	Chiffrement de César	3
1.2.2	Chiffrement par substitution	4
1.2.3	Chiffrement par permutation (transposition)	4
1.3	Loi (Principe) de Kerckhoffs	4
2	Contribution de Shannon	4
2.1	Rappels de probabilité	4
2.1.1	Probabilité conditionnelle	4
2.1.2	Variables aléatoires	5
2.2	Crypto-système	5
2.2.1	Indépendance	5
2.3	Système "parfait" au sens de Shannon (confidentialite parfaite)	6
2.3.1	Caractère parfait	6
2.3.2	Extension du modèle de Shannon (Simmons)	7
3	Sécurité Calculatoire (chiffres par flot, générateurs pseudo-aléatoires)	8
3.1	Chiffre par flot(Stream Cipher)	9
3.1.1	Exemple : Von Neumann	10
3.2	Cas des générateurs linéaires.	12
4	Chiffres par blocs	15
4.0.1	Suggestion de Shannon	16
4.0.2	Idée de Feistel	16
4.1	DES	16

4.1.1	Successeurs du DES	17
4.2	AES (Advanced Encryption Sandard)	18
5	Fonctions sens unique (ou difficilement inversibles) one-way	20
5.1	Définition	20
5.2	Protection des mots de passe	21
5.3	Quelles fonctions peuvent être sens unique ?	21

1 Introduction

Les besoins de la cryptologie :

- Confidentialité (Secret)
- Intégrité (des messages)
- Authentification (émetteur, destinataire)
- Non-répudiation

Quelques exemples d'utilisation :

- Paiement
- Monnaie numérique
- vote électronique
- ventes aux enchères
- Pour garder l'ANONYMAT

1.1 Chiffrement

Modèle Standard

$$A, M \xrightarrow{C=f(M)} B, M = f^{-1}(C)$$

O

- A émetteur **chiffre**
- B récepteur **déchiffre**
- O observateur (passif) **décrypte**
- M message en clair
- C message chiffré (cryptogramme)
- f transformation secrète de chiffrement

Le secret est partagé entre A et B.

1.2 Système de Chiffrement (Chiffre)

$(f_k)_{k \in K}$ est l'ensemble des «clés».

(f_k) définit l'ensemble des fonctions permettant de chiffrer

1.2.1 Chiffrement de César

$$A \longrightarrow D$$

$$B \longrightarrow E$$

$$C \longrightarrow F$$

1.2.2 Chiffrement par substitution

(f_σ) où σ est une permutation de l'ensemble $\{A, B, C, \dots, Z\}$

CRYPTO $\rightarrow f(CRYPTO) = \sigma(C)|\sigma(R)|\sigma(Y)|\sigma(P)|\sigma(T)|\sigma(O)$

1.2.3 Chiffrement par permutation (transposition)

1	2	3	4	5	6
C	R	Y	P	T	O

$$\sigma = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 3 & 5 & 4 & 6 & 1 & 2 \end{pmatrix}$$

$$C = YTPOCR$$

1.3 Loi (Principe) de Kerckhoffs

Le secret doit dépendre de la seule clé (et non pas du système entier).

2 Contribution de Shannon

$\langle M, K, C \rangle$ sont des *variables aléatoires*.

La quantité pertinente à étudier est : $P(M = m|C = c)$.

2.1 Rappels de probabilité

(Ω, P) est un espace probabilisé. La probabilité associée est une fonction $P : \mathcal{P}(\Omega) \rightarrow [0, 1]$

- si $A \cap B = \emptyset$, alors $P(A \cup B) = P(A) + P(B)$
- $P(\emptyset) = 0, P(\Omega) = 1$.

2.1.1 Probabilité conditionnelle

$P(A|B) = \frac{P(A \cap B)}{P(B)}$ Notation : $P(A \cap B) = P(A, B)$

2.1.2 Variables aléatoires

$X : \Omega \Rightarrow E$ $P(X = x) = P(X^{-1}(x))$ $P(X \in A) = P(X^{-1}(A))$

Loi de X ; la donnée des $P(X = x)$.

X prend ses valeurs dans E .

Loi uniforme si $P(X = x) = \frac{1}{\text{card}E}$ pour tout $x \in E$

2.2 Crypto-système

Un système crypto (chiffrement) est *parfait* si

$$\forall m, c, P(M = m|C = c) = P(M = m)$$

2.2.1 Indépendance

$A, B \in \Omega$ indépendants si

$$P(A, B) = P(A)P(B)$$

$$P(A|B) = P(A)$$

X, Y variables indépendantes

$$\forall x, y P(X = x, Y = y) = P(X = x)P(Y = y)$$

Exemple

	a	b	c
k_1	1	3	2
k_2	1	2	3
k_3	2	3	1

On fait l'hypothèse que K indépendant de M

$$P(M = a|C = 1) = \frac{P(M=a, C=1)}{P(C=1)}.$$

$$\begin{aligned} P(C = 1) &= P(M = a, K = k_1) + P(M = a, K = k_2) + P(M = c, K = k_3) \\ &= P(M = a) \frac{1}{3} + P(M = a) \frac{1}{3} + P(M = c) \frac{1}{3} \\ &= \frac{2}{3} P(M = a) + \frac{1}{3} P(M = c) \end{aligned}$$

$$\begin{aligned} P(M = a, C = 1) &= P(M = a, K = k_1) + P(M = a, K = k_2) \\ &= \frac{2}{3} P(M = a). \end{aligned}$$

$$P(M = a|C = 1) = \frac{\frac{2}{3}P(M = a)}{\frac{2}{3}P(M = a) + \frac{1}{3}P(M = c)} = \frac{1}{1 + \frac{1}{2} \frac{P(M=c)}{P(M=a)}}$$

2.3 Système "parfait" au sens de Shannon (confidentialité parfaite)

$$\forall m \in M, \forall c \in C, P(M = m|C = c) = P(M = m)$$

exemple : "One-time pad", système de Vernam (1926). Masque jetable.

$$\begin{array}{r} M = M_1 \ . \ . \ . \ M_n \\ + K = K_1 \ . \ . \ . \ K_n \\ \hline C = C_1 \ . \ . \ . \ C_n \end{array}$$

avec,

$$M_i \in (\mathbb{Z}/m\mathbb{Z}); (m = 2, m = 256, m = 26, \dots).$$

$$C_i = M_i + K_i \pmod{n}.$$

Les K_i sont aléatoires, uniformes : $P(K_i = k) = \frac{1}{m}$, indépendants entre eux, et indépendants de M . La clé K est secrète et utilisée qu'une seule fois.

2.3.1 Caractère parfait

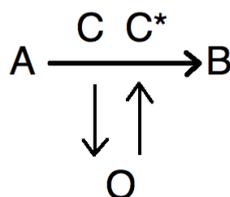
$$P(M = m | C = c) = \frac{P(M = m, C = c)}{P(C = c)} = \frac{P(M = m) \frac{1}{\#\mathcal{K}}}{\frac{1}{\#\mathcal{K}}} = P(M = m)$$

$$\begin{aligned} P(M = m, C = c) &= P(M = m, K = c - m) \\ &= P(M = m)P(K = c - m) \\ &= P(M = m) \frac{1}{\#\mathcal{K}} \\ &= P(M = m) \frac{1}{m^n} \end{aligned}$$

$$\begin{aligned} P(C = c) &= P\left(\bigcup_{k \in \mathcal{K}} \{K = k, M = c - k\}\right) \\ &= \sum_{k \in \mathcal{K}} P(K = k) P(M = c - k) \\ &= \frac{1}{\#\mathcal{K}} \sum_{k \in \mathcal{K}} P(M = c - k) \end{aligned}$$

$$= \frac{1}{\#\mathcal{K}}$$

2.3.2 Extension du modèle de Shannon (Simmons)



Il y a 2 stratégies pour l'observateur O :

- **Imposture** (Impersonation) : O n'observe pas de transmission mais essaie de faire accepter C^* . (Réussite de probabilité P_I).
 - **Substitution** : O observe et substitue $C^* \neq C$ (probabilité P_S).
- La probabilité de "tromperie" (deception) est $P_T = \max(P_S, P_I)$.

exemple : Il s'agit de chiffrer un ensemble de messages courts. $M = \{0, 1\}$.

I.

$K \ M$	0	1
00	00	11
01	01	10
10	10	01
11	11	00

- Confidentialité ?

$$\begin{aligned}
 P(M = 0 \mid C = 00) &= \frac{P(M = 0, C = 00)}{P(C = 00)} \\
 &= \frac{P(M = 0, K = 00)}{P(M = 0, K = 00) + P(M = 1, K = 11)}
 \end{aligned}$$

or K et M sont indépendantes donc

$$\begin{aligned}
 &= \frac{P(M = 0)P(K = 00)}{P(M = 0)(1/4) + P(M = 1)(1/4)} \\
 &= \frac{P(M = 0)(1/4)}{1/4} \\
 &= P(M = 0).
 \end{aligned}$$

Le résultat est le même pour les autres calculs. La confidentialité est donc parfaite. Cependant, l'observateur peut substituer : $C = 00 \rightarrow C^* = 11$, il n'y a aucune authenticité, $P_S = 1$.

On observe par ailleurs, $P_I = \frac{1}{4} + \frac{1}{4} = \frac{1}{2}$.

II. On peut faire mieux que le premier cas.

$K \ M$	0	1
00	00	10
01	01	00
10	11	01
11	10	11

• Confidentialité parfaite car le cas $P(M = 0 \mid C = 00) = P(M = 0)$ se généralise.

$$P_I = \frac{1}{2}.$$

On trouve $P_S = \max(P(M = 0), P(M = 1))$. Cette quantité peut être supérieure 1/2 et même être proche de 1. Peut-on faire mieux ?

III.

$K \ M$	0	1
00	00	10
01	01	11
10	00	11
11	01	10

• Confidentialité nulle.

$$P(K = 00 \mid C = 00) = P(K = 10 \mid C = 00) = \frac{1}{2} = P_S.$$

On a fait mieux que dans le cas précédent, mais pour descendre la probabilité de tromperie 1/2 il a fallu sacrifier la confidentialité.

Remarque III : $C = (M, \mathcal{S}(M, K))$, où \mathcal{S} est une signature numérique.

3 Sécurité Calculatoire (chiffres par flot, générateurs pseudo-aléatoires)

– Taille des clés (comment l'engendrer ?)

– Partage des clés ?

Idée simple : "Dégrader" le One-time pad en remplaçant K_1, \dots, K_n par une suite pseudo-aléatoire S_1, \dots, S_n, \dots engendrée par une clé secrète courte aléatoire.

exemple : s_1, \dots, s_n suite de bits $\{0, 1\}$.

$s_i = f(s_{i-1}, s_{i-2}, \dots, s_{i-m})$ dépend des m bits précédents f partiellement connue s_0, \dots, s_{m-1} , partie secrète.

exemple : Générateur linéaire

$s_i = s_{i-1}h_{m-1} + s_{i-2}h_{m-2} + \dots + s_{i-m}h_0 \pmod{2}$,

clé : $h_0, \dots, h_{m-1}, s_0, \dots, s_{m-1}$.

$c_i = M_i + s_i$ avec s suite chiffrante.

3.1 Chiffre par flot(Stream Cipher)

$$\begin{array}{l} m = m_1 \dots m_n \\ + "k" = S_1 \dots S_n \\ \hline C_1 \dots C_n \end{array}$$

$s = s_1 \dots s_n$ suite chiffrante Pseudo-aléatoire

$s = f(K)$ K clé courte

$K \in \{0, 1\}^m (= \{0, 1\}^{128})$

Sécurité calculatoire (\neq inconditionnelle)

Types d'attaques

- Texte chiffré seul
- Texte clair et chiffrés connus (ou partiellement)
- Texte clair choisi

Remarque : Pour le chiffrement par flot, les attaques "Clair choisi" et "Clair connu" sont équivalentes.

Le problème de la cryptanalyse, c'est de trouver la clé K partir d'un certain nombre de symboles de la suite chiffrée ($s_1 \dots s_n$) ou encore partir de $s_1 \dots s_n$, trouver s_{n+1}, s_{n+2}, \dots

De nos jours, on se place plutôt dans les 2 derniers cas plutôt que dans le cas de l'attaque par texte chiffré seul qui est trop restrictive.

Problème 1 Réaliser un générateur nombres pseudo-aléatoires $S_1 \dots S_n \dots$ imprévisible

$$s_{n+1} = f_k(s_n, s_{n+1}, \dots, s_{n-m+1})$$

m "mémoire" du générateur

$$(s_{n+m}, s_{n+m+1}, \dots, s_{n+1}) = F(s_n, s_{n+1}, \dots, s_{n-m+1})$$

"Bonnes propriétés statistiques" Remarque :

(S_i) est périodique

au plus 2^m "états" du générateur

m - uples $(S_n \dots S_{n-m+1})$

Période max $= 2^m$.

Question Que se passe t'il si la fonction f est choisi aléatoire ?

La suite ne peut être choisie aléatoirement mais le générateur oui.

$$f : \{0, 1\}^m \mapsto \{0, 1\}.$$

Si on choisit le générateur aléatoirement (la fonction f), on s'attend une période maxi nettement plus petite, elle sera en racine de 2^m .

3.1.1 Exemple : Von Neumann

Premier vouloir utiliser des générateurs pseudo-aléatoires. (1946)

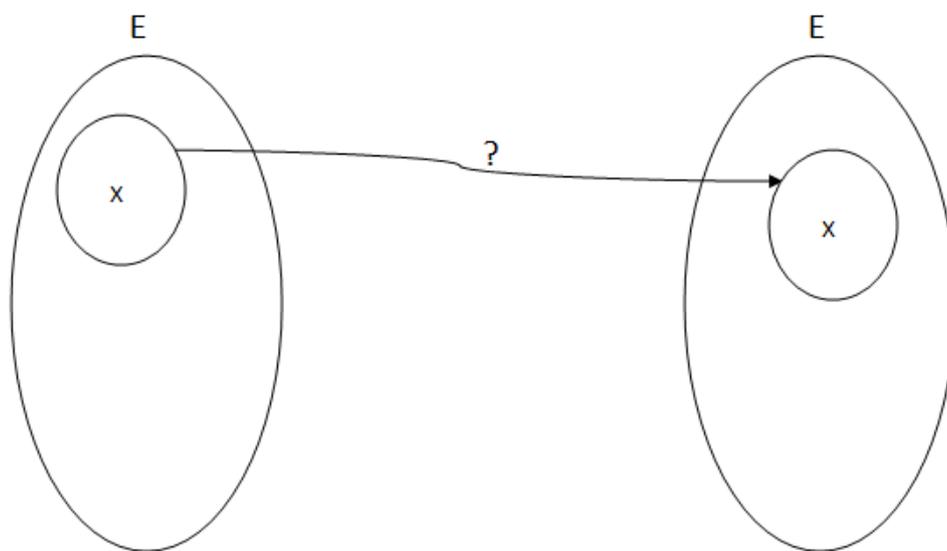
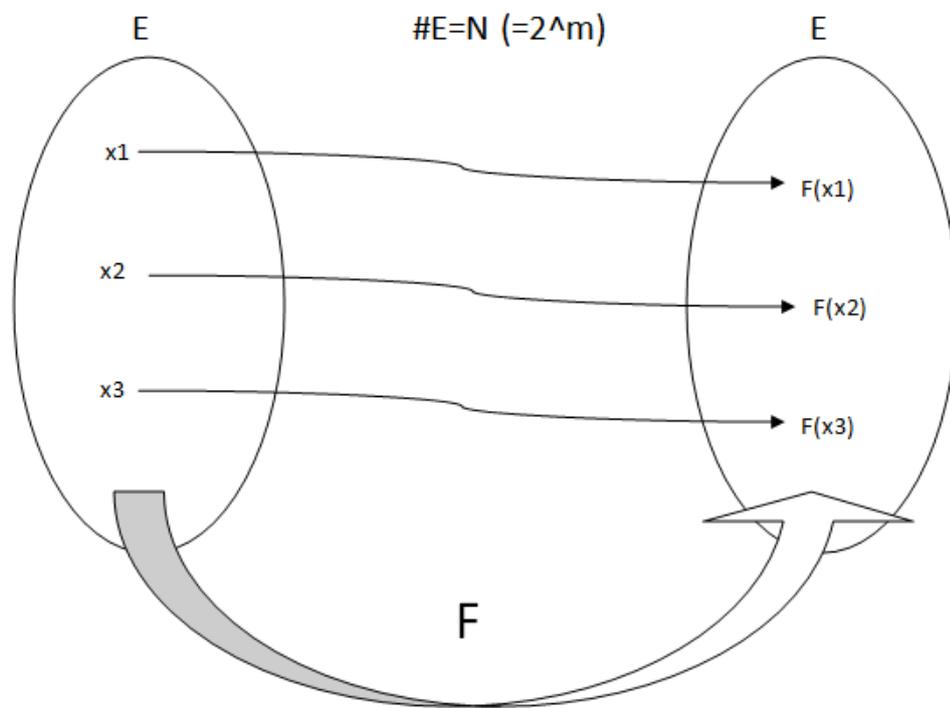
Il marche de la manière suivante :

$$f : \{0, 1\}^m \mapsto \{0, 1\} \leftrightarrow F : \{0, 1\}^m \mapsto \{0, 1\}^m$$

$$F : \{0, 99999\} \mapsto \{0, 99999\}$$

$$x \mapsto x^2$$

l'entier représenté par les 5 chiffres du milieu



$$Probabilit \mapsto (\forall i, F(x_i) \notin X_i) = \prod_{i=1}^{|X|} \left(1 - \frac{|X_i|}{N}\right)$$

$$\leq \left(1 - \frac{|X|}{2N}\right) \simeq 1 - \frac{|X|^2}{4N}$$

Problème quand $|X| \simeq \sqrt{N}$

Problème 2 $f : \{0, 1\}^m \mapsto \{0, 1\}$
 $\#f = 2^{2^m}$



3.2 Cas des générateurs linéaires.

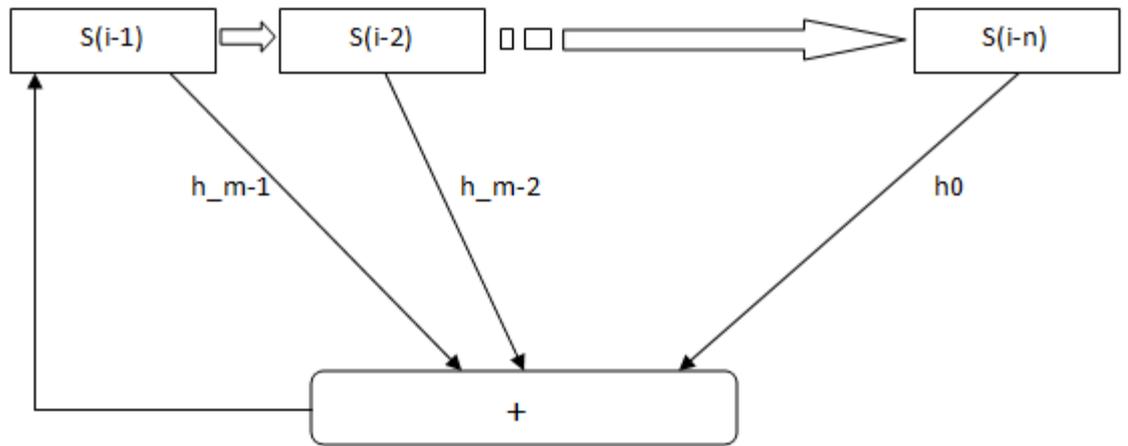
$$S_i = S_{i-1}h_{m-1} + S_{i-2}h_{m-2} + \dots + S_{i-m}h_0 \pmod{2}$$

Définition

$$h(X) = X^m + h_{m-1}X^{m-1} + \dots + h_1X + h_0 \in \mathbb{F}_2[X]$$

Polynôme de réfraction de $s = (s_i)$

LFSR (Linear Feedback Shift Register) Séquence



Théorème Si $h(X)$ est :

- 1) irréductible
- 2) primitif

Alors période (s) = $2^m - 1$

Primitif : X d'ordre $\frac{2^m-1}{e}$ dans $\mathbb{F}_2[X]/h(X)$

Plus petit entier e tel que $X^e - 1$ multiple de $h(X)$.

Nous avons de bonnes propriétés statistiques.

Mais suites très prévisibles.

car $2m$ symboles consécutifs de la suite (S_i) permettent de trouver les h_i coefficients du générateur.

h_0	$S_0 \dots S_{m-1}$	$S_m \dots S_{2m-1}$
h_1	$S_1 \dots S_m$	$S_{m+1} \dots S_{2m-1}$
h_2	$S_2 \dots S_{m+1}$	$S_{m+2} \dots S_{2m-1}$
	
	$S_m \dots S_{2m-1}$	

$$h_0 S_0 + \dots + h_{m-1} S_{m-1} = S_m$$

$$h_1 S_1 + \dots + h_{m-1} S_{m-1} = S_{m+1}$$

.....

m relations , m inconnues

Définition Notion de complexité linéaire d'une suite

Prenons une suite (S_i) périodique quelconque.

On appelle complexité linéaire de S , la mémoire minimale d'un générateur linéaire qui engendre S . $\lambda(S)$: la taille de la plus petite récurrence linéaire qui engendre S .

Remarque : $S_{i+\pi} = S_i$ avec π la période de la suite S est une récurrence linéaire qui engendre S .

Pour illustrer ce que l'on vient de dire, on veut (pour la cryptologie) un grand $\lambda(S)$. $\leq 2^{50}$.

$$\begin{array}{r} M = m_1 \ . \ . \ . \ m_n \ . \ . \ . \\ + S = s_1 \ . \ . \ . \ s_n \ . \ . \ . \\ \hline C = c_1 \ . \ . \ . \ c_n \ . \ . \ . \end{array}$$

$$S_i = f(x_i^1, \dots, x_i^n).$$

Exemple : Geffe.

$$\lambda(S) \ (c_i = c_i + \pi)$$

Stratégie de combinaison de LFSR

—————Schéma—————

Exemple de Geffe

—————Schéma—————

Théorème : s,t deux suites périodiques.

- $\lambda(s + t) \leq \lambda(s) + \lambda(t)$
- $\lambda(st) \leq \lambda(s)\lambda(t)$

Preuve : s engendré par récurrence linéaire, mémoire m

$$\textcircled{*} \ s_i = s_{i-1}h_{m-1} + \dots + s_{i-m}h_0 \quad \text{avec } m = \lambda(s)$$

L'ensemble des solutions E de $\textcircled{*}$ est stable par addition, donc c'est un espace vectoriel sur $\mathbb{F}_2 = (\{0, 1\}, +, x)$. Toute solution s de $\textcircled{*}$ s'écrit :

$$s = \alpha_1 e^1 + \alpha_2 e^2 + \dots + \alpha_k e^k$$

où (e_1, \dots, e_k) est une base de E,

$\alpha_i \in \mathbb{F}_2$ écriture unique,

$k=m$ ($\dim(E) = n$ car espace vectoriel de dimension k contient 2^k éléments

et on a 2^m suites possibles).

$$t = \beta_1 f_1 + \dots + \beta_{m'} f_{m'}$$

où $m' = \lambda(t)$

$s + t$ est dans l'espace engendré par $(e_1, \dots, e_m, f_1, \dots, f_{m'})$, $\dim(E + F) \leq m + m'$. \square

$\lambda(s + t)$ dimension de l'espace des décalés de $s + t$

$$\left\{ \begin{array}{l} s_0 \ s_1 \ s_2 \ \dots \\ s_1 \ s_2 \ s_3 \ \dots \\ \cdot \\ \cdot \\ \cdot \\ s_m \ s_{m+1} \ s_{m+2} \end{array} \right.$$

$$(s + t)_{i+d} = s_{i+d} + t_{i+d}$$

$$\begin{aligned} st &= (\alpha_1 e_1 + \dots + \alpha_m e_m)(\beta_1 f_1 + \dots + \beta_{m'} f_{m'}) \\ &= \sum_{i,j} \alpha_i \beta_j e_i f_j \end{aligned}$$

$(e_1 f_1, \dots, e_1 f_{m'}, e_2 f_1, \dots, e_2 f_{m'}, \dots, e_m f_1, \dots, e_m f_{m'})$ engendre $(s_{i+d} t_{i+d}) = ((st)_{i+d})$. \square

$$\lambda(a) = \lambda(b) = \lambda(c) = 128$$

$$\lambda(s) \leq \lambda(a) + 2 \times 128^2$$

Autre problème : Corrélation entre s_i et a_i .

– si $b_i = 0$, $s_i = a_i$

– si $b_i = 1$, $s_i = c_i = \alpha_i$ indépendant de a_i une fois sur deux.

– si $s_i = a_i$, 3 fois sur 4.

$s = f(x_1, \dots, x_n)$ doit résister aux corrélations s indépendantes de x_i .

4 Chiffres par blocs

$$\begin{aligned} f : \{0, 1\}^m \times \{0, 1\}^k &\longrightarrow \{0, 1\}^m \\ (M, K) &\longmapsto C \end{aligned}$$

4.0.1 Suggestion de Shannon

Diffusion

Confusion

Alterner des substitutions et des "transpositions" (ou applications linéaires).

m=64,128
———Schéma———

IBM : Lucifer
m=k=128
 $S_0, S_1 : 0, 1^8 \rightarrow 0, 1^8$.

4.0.2 Idée de Feistel

Idée de Feistel
———Schéma———

4.1 DES

DES qui est restée la principale norme de chiffrement (1973-1998), a nécessité des efforts importants en cryptanalyse. Rendu complètement public, beaucoup on essayé d'en montrer les faiblesses. Les attaques contre le DES donne la célébrité assurée.

Les espaces de clé n'étaient pas très grand $K \in \{0, 1\}^{56}$. Cet espace était TOUT juste trop (donc beaucoup de controverse). 2 attaques ont été trouvées. Ce sont des attaques susceptibles de fonctionner sur tout les modèles.

- La cryptanalyse différentielle (1991, Shamir, Biham)
C'est une attaque clair choisi qui ne fonctionne pas bien avec le système DES. Il s'agit d'une attaque statistique.
- La cryptanalyse linéaire (1994, Matsumoto)
C'est une attaque clair connu.
Idée : essayer de trouver une relation linéaire qui relie les bits du cryptogramme et les bits de la clé. Est ce que l'on peut trouver une relation du type $M_1 + M_3 + M_7 + \dots + M_{59} + K_2 + \dots + K_{32} + K_{35} + C_3 + C_{24} + \dots + C_{61} = 0$
 M_1 premier bit du message en clair
 K_2 bit de la clé

C_3 bit du cryptogramme

Il y a un bit qui se calcule en fonction des autres. A chaque fois, on divise par 2 l'espace des clés. Dis comme ça, c'est utopique car il n'existe pas de relation normalement, mais on peut espérer qu'il y ait de telles relations.

La probabilité d'avoir de telles relations est $\frac{1}{2} - \varepsilon$ ou $\frac{1}{2} + \varepsilon$ et existe forcément pour un $\varepsilon > 0$.

$C_1 = f(M, K)$ non linéaire, il y a un vrai espoir de trouver une relation. Si jamais on trouve une telle relation, au lieu d'avoir besoin d'un couple clair chiffré, il en faut plusieurs, par exemple, si cette relation est satisfaite avec une probabilité de ε , il faut regarder n couples M, C , et calculer $M_1 + M_3 + M_7 + \dots + M_{59} + K_2 + \dots + K_{32} + K_{35} + C_3 + C_{24} + \dots + C_{61} = 0$, de regarder combien de fois ça vaut 0, combien de fois ça vaut 1, si on trouve une majorité de 0, ça veut dire que la somme des bits de la clé K vaut 0, donc $K_2 + \dots + K_{32} + K_{35}$.

La question primordiale : Combien de couple (clair, chiffré) faut-il ?

Quel est l'ordre de grandeur de n pour être très sûr du résultat ?

Hypothèse : Valeurs indépendantes de la somme $\sum M_i + C_i$

Pour un n assez grand, la loi des grands nombres, on doit trouver $n(\frac{1}{2} - \varepsilon)$ mais en pratique, il y aura des fluctuations dont l'ordre de grandeur est : $O(\sqrt{n})$

Il faut que le $\sqrt{n} \leq n\varepsilon$

ou encore $\sqrt{n} \geq \frac{1}{2\varepsilon}$

$n \geq \frac{1}{\varepsilon^2}$

Avec le DES on a un ε qui vaut environ 2^{-21} avec un $n \approx 2^{43}$

4.1.1 Successeurs du DES

Triple DES

– $K \in \{0, 1\}^{3 \times 56} = \{0, 1\}^{168}$.

$f_k(M) = DES_{K_3}(DES_{K_2}(DES_{K_1}(M)))$

– $E_{K_1}(D_{K_2}(E_{K_1}(M))), K = (K_1, K_2) \in \{0, 1\}^{112}$

$K_1 = K_2 \Rightarrow$ DES simple

Chiffrement double

- $M \mapsto f_{K_2}(f_{K_1}(M)) = C$ on obtient un système de chiffrement à peine plus sûr que le simple
Attaque "par le milieu"
M,C clair-chiffré.

Admettons la table suivante

$f_K(M)$		$f_K^{-1}(C)$
\vdots		\vdots
\vdots		\vdots
\vdots		\vdots
$K \in \mathcal{K}$		$K \in \mathcal{K}$

Regarder dans les 2 tableaux quel est la valeur commune

4.2 AES (Advanced Encryption Standard)

créé par Rijndael

$$M, K \in \{0, 1\}^{128}$$

M

$$\oplus \leftarrow K_0 = K$$

Ronde1

$$\oplus \leftarrow K_1 = K$$

Ronde2

$$\oplus \leftarrow K_2 = K$$

\vdots

Ronde10

$$\oplus \leftarrow K_{10} = K$$

C

Ronde i : $M_i \rightarrow$ substitution \rightarrow application linéaire

Substitution : Une application $\{0, 1\}^8 \rightarrow \{0, 1\}^8$

M_i 16 octets

Application algébrique : $\mathbb{F}_2[X]/(X^8 + X^4 + X^3 + X + 1) \rightarrow \mathbb{F}_2[X]/(\)$

$$\sigma : P \mapsto P^{-1}$$

$$0 \mapsto 0$$

σ coïncide (conjecture) le moins souvent avec application affine.

$$S : \{0, 1\}^8 \rightarrow \{0, 1\}^8$$

$$x \mapsto A\sigma(x) + b$$

$$A = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ \vdots & \vdots \end{pmatrix} \longrightarrow b = \begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{pmatrix}.$$

Application linéaire.

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ & & & \\ & & & \\ a_{41} & & & a_{44} \end{pmatrix} \longrightarrow \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{22} & a_{23} & a_{24} & a_{21} \\ a_{33} & a_{34} & a_{31} & a_{32} \\ a_{44} & a_{41} & a_{42} & a_{43} \end{pmatrix}$$

2 applications successives :

- 1) Shift Rows
- 2) Mixcolumns

$$\begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix} \longleftrightarrow a(X) = a + bX + cX^2 + dX^3$$

$$a(X) \rightarrow a(X)x(X) \text{ mod } (X^4 + 1)$$

$$c(X) = (03)X^3 + X^2 + X + (02)$$

$$03 = [00000011] = (x + 1) \text{ mod } (x^8 + x^4 + x^3 + 1)$$

Ronde 10 : pas de Mixcolumns.

Clé K_i ?

$$K = K_0$$

//SCHEMA//

$$w(i) = w(i - 1) + w(i - 4) \text{ si } i \neq 0 \text{ mod } 4$$

$$w(i) = Ti(w(i - 1)) + w(i - 4) \text{ si } i = 0 \text{ mod } 4$$

Avec Ti :

$$\begin{pmatrix} a \\ b \\ c \\ d \end{pmatrix} \rightarrow \begin{pmatrix} b \\ c \\ d \\ a \end{pmatrix} \rightarrow \begin{bmatrix} S(b) \\ S(c) \\ S(d) \\ S(a) \end{bmatrix} \rightarrow \begin{bmatrix} S(b) + e_i \\ S(c) \\ S(d) \\ S(a) \end{bmatrix}$$

$$e_i = [00000010]^{(i-4)/4}$$

*Modes opératoires :

$$f_k : \{0, 1\}^N \rightarrow \{0, 1\}^N$$

$$M_1, M_2, \dots, M_i, \dots \xrightarrow{f_k} C_1, C_2, \dots, C_i, \dots$$

Si on observe $C_i = C_j$, alors on en déduit $M_i = M_j$.

– Mode ECB (electronic code book)

Pb : dictionnaire de chiffrés

– Mode CBC (cipher block chaining)

$$C_j = f_k(M_j + C_{j-1})$$

$C_0 = VI$ (= valeur aléatoire) \rightarrow communiqué en clair

– Mode CFB (cipher feedback)

$$C_j = M_j + f_k(C_{j-1})$$

– Mode OFB (output feedback)

$$X_0 = VI \text{ clair}$$

$$X_j = f_k(X_{j-1}) \text{ et } C_j = M_j + X_j$$

transforme le chiffre par blocs en chiffre par flot

5 Fonctions sens unique (ou difficilement inversibles) one-way

5.1 Définition

$$f : (K, VI) \rightarrow \{0, 1\}^n$$

Fonctions bijectives :

f "sens unique" s'il \exists un algorithme efficace qui calcule $f(x)$ et s'il \nexists un algorithme efficace qui calcule x partir de $f(x)$

$$f : \{0, 1\}^n \rightarrow \{0, 1\}^n$$

$$E \rightarrow E$$

$$x \rightarrow f(x)$$

5.2 Protection des mots de passe

$U \rightarrow m \rightarrow M$ avec $m \in \{m_0, m_1, \dots, m_k\}$

On stocke plutôt les $f(m_i)$.

La machine compare ensuite les $f(m_i)$ au $f(m)$ donné.

Pas d'accès possible si on ne sait pas inverser f .

Procédé de Lamport :

connexions distance

$U \rightsquigarrow M$

m_0

$m_1 = f(m_0)$ et $m_i = f(m_{i-1}) \rightarrow$ envoie m_i

U envoie m_{999}

M fait $m_{1000} = f(m_{999})$

si m_{1000} apparaît \rightarrow accepte puis remplace m_{1000} par m_{999}

U envoie m_{998} etc

5.3 Quelles fonctions peuvent être sens unique ?

Fonction $+$ dans $\mathbb{Z}/n\mathbb{Z} \rightarrow$ non (trop simple)

Fonction \times dans $\mathbb{Z}/n\mathbb{Z} \rightarrow$ non (inversible)

Fonction sens unique (one-way)

$$f : x \mapsto f(x)$$

\longleftarrow

Candidat :

$$f : x \mapsto \alpha^x \text{ mod}(n)$$

pas de f^{-1} de même nature β^{α^x}

$$y = \alpha^x \text{ mod}(n)$$

En général, le temps de calcul pour trouver x partir de $y = \alpha^x \text{ mod}(n)$

- Recherche exhaustive : $\sim n$ opérations dans $\mathbb{Z}/n\mathbb{Z} \rightarrow m = 2^{\log n}$

Meilleurs algorithmes : $2^{c\sqrt{\log n}}$ (sous-exp $\rightarrow 2^{c(\log n)^{\frac{1}{3}}}$)

Calculabilité de f Élévations au carré successives

$$x = \sum_{i \in I} 2^i$$

$$\alpha^x = \prod_{i \in I} \alpha^{2^i}$$

$$\alpha^{2^i} = ((\alpha^2)^2) \dots 2$$

Choix de f i.e de α , n

Par exemple, pour que f soit une bijection

$$\{0, 1, 2, \dots, n-2\} \xrightarrow{f} \{1, 2, \dots, n-1\}$$

$$x \mapsto \alpha^x \text{ mod}(n)$$

Théorème (Elément primitif) Si n premier, $\exists \alpha$ (primitif) t.q f est une bijection :

- $1, \alpha, \alpha^2, \dots, \alpha^{n-2}$ sont $2 \cdot 2 \neq$
- ordre de $\alpha = n - 1$

Trouver n, α adéquats ?? (Théorème de Lucas) n est premier ssi $\exists \alpha$ t.q

- $\alpha^{n-1} = 1 \text{ mod}(n)$
- $\forall q(n-1), q$ premier, $\alpha^{\frac{n-1}{q}} \neq 1 \text{ mod}(n)$

\Rightarrow si n premier . Th de l'élément primitif

\Leftarrow Si

$$\alpha^{n-1} = 1 \text{ mod}(n), \alpha \in (\mathbb{Z}/n\mathbb{Z})^*$$

Le nombre d'éléments dans le groupe x est $\varphi(n)$

$$\frac{\alpha^{\varphi(n)} = 1}{\alpha^{n-1} = 1} \alpha^{\text{pgcd}(n-1, \varphi(n))} = 1$$

$$\varphi(n) = n - 1 \bullet$$

2 idées clés.

- n, α au hasard

Justification Beaucoup de nombres premiers.

$$\pi(n) \sim \frac{n}{\ln n}$$

$$\text{Proba} \simeq \frac{1}{\ln n}$$

– imposer q : imposer les facteurs de $n - 1$

Choisir $q_1 \dots q_k$ premier

$$n = 1 + q_1^{m_1} \dots q_k^{m_k}$$

Concretement : Pour fabriquer des nombres premiers de plus en plus grand.

$$Q = \{2, 3, 5, \dots\}$$

$$- 1) q \simeq 2^{50}$$

$$- 2) q \simeq 2^{100}, 2^{150}, \dots$$

Chercher simultanément α, n .

Seule manière connue d'avoir α primitif mod (n)

$$x \mapsto \alpha^x \text{ mod } (n).$$

$$(\alpha^a)^b = (\alpha^b)^a.$$

Protocole de Diffie-Hellman(1976) Résoudre le problème du partage des clés.

ATTENTION SCHEMA DE ALICE ET BOB

A et B partagent α , p premier, α primitif mod p
(publie)

A choisi a secret, calcul $\alpha^a \text{ mod } (p)$, le donne B.

B choisi b secret, calcul $\alpha^b \text{ mod } (p)$, le donne A.

A et B conviennent de Partager

$$S = \alpha^{ab} \text{ mod } (p)$$

$$A : (\alpha^b)^a = S \text{ mod } (p)$$

$$B : (\alpha^a)^b = S \text{ mod } (p)$$

Conclusion : Si le problème de DH est algorithmiquement difficile, alors A et B savent partager un secret donc le protocole est efficace

Chiffrement Idée de clé "publique".

$$f_k : M \mapsto C \mapsto f_k^{-1}(C) = M$$

(f, f^{-1}) un algorithme \mathcal{A} pour calculer f et un algorithme \mathcal{B} pour calculer f^{-1} .

Idée : B garde \mathcal{B} secret (destinataire)
mais rend public \mathcal{A} (calcul f)

$$A : \mathcal{A}(M) \xrightarrow{C} B.\mathcal{B}(C) = M$$

Pour cela il faut :

f sens unique "avec trappe" (trapdoor) $\exists \mathcal{B}, \exists$ secret qui permet de calculer f^{-1}

Chiffrement El Gamal Publiques ; α, p , et $P = \alpha^s \text{mod}(p)$

Secret : s

Chiffrement : α, p, P .

A choisit k au hasard

$$C_1 = \alpha^k \text{mod}(p)$$

$$C = (C_1, C_2)$$

$$C_2 = MP^k$$

Déchiffrement : s

$$P^k = \alpha^{sk} = (\alpha^k)^s = C_1^s$$

$$M = C_2(C_1^s)^{-1} \text{mod}(p)$$

Il suffit d'un groupe commutatif G

$$x \mapsto \alpha^x \text{ a sens unique.}$$