

CRYPTOGRAPHIE

I – CRYPTOSYSTEMES :

1. CESAR :

$$C_i = m_i + \text{offset mod } (26)$$

$$M_i = c_i - \text{offset mod } (26)$$

Exemple (offset = 3) :

| | | | | | | | | |
|---|--------------|---|----|----|---|----|----|----|
| + | CLAIR | A | S | T | E | R | I | X |
| | | 0 | 19 | 20 | 4 | 18 | 8 | 23 |
| | CLEF | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| = | CRYPTOGRAMME | 3 | 21 | 23 | 7 | 21 | 11 | 0 |
| | | D | V | W | H | U | L | A |

→ Cryptanalyse :

- Analyse fréquentielle.
- Essais systématiques.

2. VIGENERE :

| | | | | | | | | |
|---|--------------|---|----|----|---|----|----|----|
| + | CLAIR | A | S | T | E | R | I | X |
| | | 0 | 19 | 20 | 4 | 18 | 8 | 23 |
| | CLEF | 3 | 1 | 4 | 3 | 1 | 4 | 3 |
| = | CRYPTOGRAMME | 3 | 20 | 24 | 7 | 19 | 12 | 0 |
| | | D | T | Y | H | S | M | A |

→ Cryptanalyse :

- Analyse fréquentielle, regarder les rangs modulo la longueur de la clef.

3. PLAYFAIR :

→ $26 = 5^2 + 1$. On choisit une lettre à ne pas coder (ex : Y).

| | | | | |
|---|---|---|---|---|
| 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 |

→ Le codage se fait par bi-gramme, et voici les trois cas possibles :

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Le bi-gramme détermine la diagonale d'un rectangle : Le bi-gramme du crypto est la diagonale inverse. | Le bi-gramme est sur la même ligne : Le bi-gramme du crypto est formé par le bi-gramme de mêmes colonnes, de la ligne suivante. | Le bi-gramme est sur la même colonne : Le bi-gramme du crypto est formé par le bi-gramme de mêmes lignes, de la colonne suivante. | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Exemple : | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <table border="1"> <tr><td>A</td><td>B</td><td>C</td><td>D</td><td>E</td></tr> <tr><td>F</td><td>G</td><td>H</td><td>I</td><td>J</td></tr> <tr><td>K</td><td>L</td><td>M</td><td>N</td><td>O</td></tr> <tr><td>P</td><td>Q</td><td>R</td><td>S</td><td>T</td></tr> <tr><td>U</td><td>V</td><td>W</td><td>X</td><td>Z</td></tr> </table> | 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 | <table border="1"> <tr><td>A</td><td>B</td><td>C</td><td>D</td><td>E</td></tr> <tr><td>F</td><td>G</td><td>H</td><td>I</td><td>J</td></tr> <tr><td>K</td><td>L</td><td>M</td><td>N</td><td>O</td></tr> <tr><td>P</td><td>Q</td><td>R</td><td>S</td><td>T</td></tr> <tr><td>U</td><td>V</td><td>W</td><td>X</td><td>Z</td></tr> </table> | 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 | <table border="1"> <tr><td>A</td><td>B</td><td>C</td><td>D</td><td>E</td></tr> <tr><td>F</td><td>G</td><td>H</td><td>I</td><td>J</td></tr> <tr><td>K</td><td>L</td><td>M</td><td>N</td><td>O</td></tr> <tr><td>P</td><td>Q</td><td>R</td><td>S</td><td>T</td></tr> <tr><td>U</td><td>V</td><td>W</td><td>X</td><td>Z</td></tr> </table> | 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 |
| 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 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 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 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 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 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| • (G,S) est codé en (I,Q) | • (G,I) est codé en (L,N) | • (G,Q) est codé en (H,R) | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

→ Cryptanalyse :

- Analyse fréquentielle par bi-gramme.
- Chaque lettre ne peut être codée qu'au maximum par 4 alternatives.

4. HILL :

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|-----|-----------------------------|-----|----------|----|---|---|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|---|---|----|----|----|-----|----|
| Le Crypto-gramme | | La clef (Une Matrice nxn) | | Le Clair | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <table border="1"> <tr><td>C1</td></tr> <tr><td>C2</td></tr> <tr><td>C3</td></tr> <tr><td>...</td></tr> <tr><td>Cn</td></tr> </table> | C1 | C2 | C3 | ... | Cn | = | <table border="1"> <tr><td>A11</td><td>...</td><td>...</td><td>...</td><td>A1n</td></tr> <tr><td>...</td><td>...</td><td>...</td><td>...</td><td>...</td></tr> <tr><td>...</td><td>...</td><td>...</td><td>...</td><td>...</td></tr> <tr><td>...</td><td>...</td><td>...</td><td>...</td><td>...</td></tr> <tr><td>An1</td><td>...</td><td>...</td><td>...</td><td>Ann</td></tr> </table> | A11 | ... | ... | ... | A1n | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | An1 | ... | ... | ... | Ann | X | <table border="1"> <tr><td>M1</td></tr> <tr><td>M2</td></tr> <tr><td>M3</td></tr> <tr><td>...</td></tr> <tr><td>Mn</td></tr> </table> | M1 | M2 | M3 | ... | Mn |
| C1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| C2 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| C3 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Cn | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| A11 | ... | ... | ... | A1n | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | ... | ... | ... | ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | ... | ... | ... | ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | ... | ... | ... | ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| An1 | ... | ... | ... | Ann | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| M1 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| M2 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| M3 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| ... | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Mn | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

→ Avec A, une matrice inversible dans l'anneau Z_{26} ($\Leftrightarrow \text{Dét}(A)$ non diviseur de 0).

→ Cryptanalyse :

- Si on dispose de n observations ($(C_i)_{i=1..n}$; $(M_i)_{i=1..n}$). On aura n^2 équations pour n^2 inconnues ($(A_{ij})_{i=1..n,j=1..n}$). En inversant le système on obtient A.

5. ENIGMA :

C'est une alternance entre VIGENERE et des permutations sur les mono-gramme.

6. CHE GUEVARA :

→ On n'associe plus aux lettres des nombres entre 0 et 25, mais des nombres de longueur variable !

Exemple :

| Codage | | Chiffrement | | | | | | |
|--------|-----|-------------|---|-----|-------|--|--|--|
| A | 1 | Clair : | A | B | C | | | |
| B | 45 | | 1 | 4 5 | 8 2 1 | | | |
| C | 821 | Clef : | 3 | 1 4 | 3 1 4 | | | |
| ... | ... | Crypto : | 4 | 5 9 | 1 3 5 | | | |

→ Cryptanalyse :

- Ainsi, pour le cryptanalyste, dans la séquence cryptée '459135', le monogramme '5', peut correspondre au crypto du deuxième monogramme de clair, ou du premier monogramme de clair ...
On ne peut plus raisonner par analyse fréquentielle, car un mono-gramme peut-être codé sur un poly-gramme !

II – CRYPTOGRAPHIE ARITHMETIQUE:

→ RAPPELS SUR LES ANNEAUX D'ENTRIERS :

- L'ordre d'un élément d'un groupe :

Soit un groupe multiplicatif (G,x) , et $A \in G$.
L'ordre de A est le plus petit entier e tel que $A^e = 1$.

→ L'existence des entiers e , tels que $A^e = 1$, est prouvée par le fait que G est fini.

- Théorème de LAGRANGE :

Si $A \in (G,x)$ fini (n éléments) et A d'ordre n .
Alors $A^n = 1$.

- Théorème de FERMAT :

Si p est premier alors : $\forall x \neq 0, x^{p-1} = 1 \pmod{p}$

→ (Démonstration : Si p est premier, alors l'anneau $(G,+x)$, fini à p éléments, est un corps. On applique alors le théorème de LAGRANGE, au groupe multiplicatif de ce corps : (G^*,x) , qui possède $p-1$ éléments ...)

- La fonction d'EULER :

La fonction d'Euler d'un entier $n \in \mathbb{N}$, est le nombre d'entiers inférieurs à n et premier avec lui. Elle est notée : $\varphi(n)$.

- Propriétés :

- Si p est premier : $\varphi(p) = p-1$.
- $\varphi(p \times q) = \varphi(p) \times \varphi(q)$.

- Théorème d'EULER :

Si $n \in \mathbb{N}$ et $\varphi(n)$ est sa fonction d'Euler
Alors $x^{\varphi(n)} = 1 \pmod{n}$.

→ Tests de Primalité :

Si la réciproque du théorème de Fermat, avait été vraie, ce théorème aurait été bien utile pour générer des nombres premiers avec une complexité correcte ... Une manière de rendre cette réciproque vraie est de caractériser les exceptions.

- Nombres Pseudo premiers faibles :

Un entier n, non premier est un pseudo premier faible de base a s'il vérifie la relation :

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

→ Il existe 1770, nombres pseudo premiers faibles pour les bases 2,3,5 et 7 et inférieurs à $25 \cdot 10^9$. La connaissance des ces nombres et un test de Fermat permettent d'établir la primalité, d'un nombre $< 25 \cdot 10^9$. Il est intuitif de penser que si on augmente le nombre de bases, on atteindra un nombre nul d'entiers non premiers et pseudo premiers faibles pour toute ces bases ... ET BIEN C'EST FAUX !

- Nombres de Carmichael :

Un nombre pseudo premier faible de toute base est dit nombre de Carmichael.

→ Caractérisation mathématique :

(n est un nombre de Carmichael)

↔

(n est non premier, sans facteurs carrés et $\forall p$, facteur de n, $p-1 \mid n-1$)

→ Il y'a 2163 nombres de Carmichael $< 25 \cdot 10^9$. Le plus petit d'entre eux est $561=3 \cdot 11 \cdot 17$.

Un critère plus puissant qui élimine les nombres de Carmichael est le suivant :

- Nombres pseudo premiers forts :

(Un nombre non premier n, tel que : $n-1=2^f \cdot d$, avec d impair, est un pseudo premier fort pour la base a)

↔

([$a^d = 1 \pmod{n}$] OU [$\exists s \in \{ 0,1,2, \dots, r-1 \}$, $S = d \cdot 2^s$ et tel que $a^S = -1 \pmod{n}$])

→ Il y'a seulement 13, nombres $< 25 \cdot 10^9$ et pseudo premiers forts pour les bases 2,3 et 5. Et si on complète les tests pour les bases 7 et 11, ce nombre tombe à 0 !!!
Aucun résultat n'a été prouvé, quant à la base à partir de laquelle le nombre des pseudo premiers forts (de taille inconnue), tombe à zéro ...

→ Un raisonnement de type probabilistique est alors adopté. En effet pour les longueurs d'entiers utilisés dans la crypto, le taux d'erreur dans le test de primalité de Fermat (+ test de pseudo primalité forte) est de l'ordre de 10^{-12} , c'est à dire de l'ordre du taux de modification d'un bit par rayonnement cosmique, ou du taux de défection d'un module mémoire ... Tout d'un coup la notion de nombre premier avec une certaine probabilité, n'ayant aucun sens mathématique, en admet un en informatique !

→ Exponentielle et Logarithme d'entiers :

On considère le corps \mathbb{F}_p à p éléments (p est forcément premier).

$$\begin{array}{lcl} \text{Exp :} & \mathbb{F}_p^* & \rightarrow \mathbb{F}_p^* \\ & x & \rightarrow a^x \text{ mod } (p) \end{array}$$

est une bijection et sa réciproque est appelée logarithme de base a :

$$\begin{array}{lcl} \text{Log}_a : & \mathbb{F}_p^* & \rightarrow \mathbb{F}_p^* \\ & y & \rightarrow x, \text{ tel que : } y = a^x \text{ mod } (p) \end{array}$$

La fonction exponentielle, parfaitement inversible au sens mathématique, ne l'est pas au sens informatique ... En effet la complexité de calcul de l'exponentielle d'un entier est de l'ordre de $O(\log_2(p))$ avec l'algorithme de Strassen (diviser pour régner), alors que celle de calcul du logarithme n'est pas en-dessous de $O(p)$...

→ Factorisation d'un entier :

Ce même problème est posé, dans le cas de la factorisation d'un entier :

On note $\mathbb{P}(N)$, l'ensemble des nombres premiers $< N$, et E l'ensemble des entiers se décomposant en un produit de deux nombres premiers (E est appelé ensemble RSA).

$$\begin{array}{lcl} \mathbb{P}^2(N) & \rightarrow & E \subset \mathbb{N} \\ (p,q) & \rightarrow & p \cdot q \end{array}$$

Cette application est une bijection, au sens mathématique, mais pas au sens informatique. En effet :

La complexité de la factorisation est de l'ordre de $O(N^{\log(\log(N))})$.

Alors que celle de son inverse (simple produit) est de l'ordre de : $O(1)$.

→ Ce sont ces propriétés de non inversibilité algorithmique qui vont garantir la solidité des algorithmes arithmétiques de cryptage. On distingue toutefois deux types de problèmes (de cryptanalyse) :

- **Problème NP Complet** : La théorie écarte toute possibilité d'amélioration de la complexité (Cas de l'algorithme d'entiers). La solidité d'un tel algorithme ne dépendra pas du temps.
- **Problème non NP Complet** : La théorie n'a pas encore dit son dernier mot sur la possibilité d'améliorer la complexité des calculs. La solidité d'un tel algorithme pourrait être compromise par la découverte de méthodes de complexité inférieure.

1. EL GAMAL :

A → B
A désire communiquer avec B.

→ ETABLISSEMENT DE CLEF :

- B choisit p (premier de l'ordre de 10^{150}).
- B choisit g avec $1 < g < p-1$.
- B choisit s avec $1 < s < p-1$.
- B calcule g^s .

Public : p, g, g^s .

Privé : s.

→ CHIFFREMENT : (Soit $1 < m < p$, le message à encrypter)

- A choisit un aléa $1 < r < p-1$ et calcule $C_1 = g^r \text{ mod } (p)$.
- A calcule $C_2 = m.(g^s)^r \text{ mod } (p) = m.g^{s.r} \text{ mod}(p)$.
- A envoie C_1 et C_2 .

→ DECHIFFREMENT :

- B reçoit C_1 et C_2 .
- B calcule $d = C_1^{-s} \text{ mod } (p) = g^{-r.s} \text{ mod } (p)$ et $C_2.d = m.g^{s.r}.g^{-r.s} \text{ mod}(p) = m \text{ mod } (p)$.

→ SOLIDITE :

La solidité d'EL GAMAL, réside dans la complexité du calcul du logarithme d'un entier : calculer s, sachant g et g^s .

→ CARACTERISTIQUES :

| | |
|--------------------|--|
| CHIFFREMENT | 2 Exponentielles en pré-calcul + 1 Produit. |
| DECHIFFREMENT | 1 Exponentielle en pré-calcul + 1 Produit. |
| TAUX D'EXPANSION : | $M \rightarrow (C_1, C_2)$. Débit divisé par 2. |

→ PERFORMANCE : Optimal, en différé, car possibilité de pré-calcul.

2. ELLIS / RSA (Rivest / Shamir / Adelman) :

A → B
A désire communiquer avec B.

→ ETABLISSEMENT DE CLEF :

- B choisit p,q (premiers de l'ordre de 10^{150}) et calcule $n=p.q$ et $\varphi(n) = (p-1).(q-1)$.
- B trouve k,k' tels que $k.k' = 1 \pmod{\varphi(n)}$.

Public : n, k.

Privé : $\varphi(n)$, k'.

→ CHIFFREMENT : (entier $m < n$ à chiffrer)

- A calcule $C(m) = m^k$ et l'envoi.

→ DECHIFFREMENT :

- B calcule $(C(m))^{k'} = m^{k.k'} \pmod{n} = m^{1+\lambda.\varphi(n)} \pmod{n} = m \pmod{n}$.

→ SOLIDITE :

THEOREME DE CRYPTOGRAPHIE :

Pour un système RSA, connaissant n et k. Retrouver k' équivaut à Factoriser n

Preuve : En considérant l'équation du second degré : $X^2 - SX + P$.

Avec $S = p+q$ et $P = p.q = n$ et $\varphi(n) = (p-1).(q-1) = n - S + 1$.

→ CARACTERISTIQUES :

| | |
|--------------------|--|
| CHIFFREMENT | 1 Exponentielle |
| DECHIFFREMENT | 1 Exponentielle |
| TAUX D'EXPANSION : | $M \rightarrow C(M)$. Débit conservé. |

→ PERFORMANCE : Optimal, en temps réel, car moins de calcul d'exponentielles.

→ PROBLEMES D'ETABLISSEMENT DE CLEFS RSA :

1. Lors de l'établissement de ces clefs, il est demandé de trouver deux entiers k et k' , tels que $k.k' = 1 \pmod{\varphi(n)}$ comment faire ?

ASTUCE :

$$k.k' = 1 + \lambda.\varphi(n) \Leftrightarrow 1 + \lambda.\varphi(n) = \text{mod}(k)$$

- On commence à $k = 257$.
- On fait tourner un programme, (une boucle de 256 itérations) et on trouve λ_0 .
- Alors $k' = (1 + \lambda_0.\varphi(n))/257$.

2. Comment généraliser pour des k et k' grands ?

ASTUCE :

$$k.k' = 1 \pmod{\varphi(n)} \Rightarrow k^i.k'^i = 1 \pmod{\varphi(n)}$$

Ainsi en partant de $k = 257$, et pour $i = 50$, on aura k et k' de l'ordre de 10^{150} !!!

III – CRYPTOSYSTEMES DES / IDEA / AES :

Le développement spectaculaire des cryptosystèmes à clef publique ne doit pas masquer le fait que les cryptosystèmes à clef secrète, sont très prisés pour leur rapidité et leur robustesse. Ils n'effectuent pas des opérations de chiffrement bit par bit, ce qui les rend plus appropriés pour des systèmes en ligne et à haut débit.

1. DES (DATA ENCRYPTION STANDARD) :

L'algorithme du DES chiffre des blocs de 64 bits à l'aide d'une clef k de 64 bits (dont 8 de parité, situés à la fin de chaque octet, c'est à dire sur les positions : 8 ,16, 24, 32, 40, 48,56,64 et qui sont ignorés par les calculs).

Il se déroule en trois étapes distinctes :

1. Une permutation IP, indépendante de la clef K , est appliquée au bloc de 64 bits.
2. 16 Itérations d'une structure de Feistel (définie à l'aide d'une fonction ϕ quelconque et de 8 S boxes), voici la description d'une itération :
 - Le premier argument de ϕ (32 bits de données), A est explosé en une chaîne de 48 bits : $E(A)$, par une permutation (indépendante des données et de la clef : aucun renforcement de la solidité), de ses bits et la répétition de 16 parmi eux.
 - Une chaîne J de 48 bits est obtenue des 56 bits significatifs de la clef K (par permutations circulaires sur cette dernière).
 - L'opération $E(A) \oplus J$ donne une chaîne B de 48 bits qui sont subdivisés en 8 blocs de 6 bits chacun : B_1, \dots, B_8 .
 - On utilise 8 S boxes pour contracter ces 8 blocs de 6 bits vers des blocs de 4 bits.
 - Une permutation est appliquée à la chaîne de 32 bits résultante.
3. La permutation IP^{-1} est appliquée au résultat.

L'étude de ce cryptosystème fait intervenir deux nouvelles notions en cryptologie :

1.1 Les structures de Feistel :

On se donne :

- Une fonction quelconque $\varphi() : |F^{32} \times |F^{48} \rightarrow |F^{32}$
- Une clef K de 64 bits.

| | | |
|---|---|---|
| On considère le bloc _i de données de 64 bits (on peut généraliser à n bits). | <div style="border: 1px solid black; width: 100%; height: 20px; margin-bottom: 5px;"></div> Bloc _i | |
| On le découpe en deux parties égales de 32 bits (n div 2). | <div style="border: 1px solid black; width: 50%; height: 20px; margin-bottom: 5px;"></div> G _i | <div style="border: 1px solid black; width: 50%; height: 20px; margin-bottom: 5px;"></div> D _i |
| On applique la transformation suivante sur les deux blocs G _i et D _i : - D _{i+1} = G _i ⊕ φ(D _i ,K _i) - G _{i+1} = D _i (Où les K _i sont obtenues à partir de la clef K) | <div style="border: 1px solid black; width: 50%; height: 20px; margin-bottom: 5px;"></div> D _i | <div style="border: 1px solid black; width: 50%; height: 20px; margin-bottom: 5px;"></div> G _i ⊕ φ(D _i ,K _i) |
| On obtient le bloc _{i+1} | <div style="border: 1px solid black; width: 100%; height: 20px; margin-bottom: 5px;"></div> Bloc _{i+1} | |

→ Ainsi on peut écrire $(G_{i+1}, D_{i+1}) = f_{\varphi}(G_i, D_i) = (D_i, G_i \oplus \varphi(D_i, K_i))$

L'intitulé du théorème de Feistel est que f est toujours inversible quelque soit la fonction φ choisie !!!

En effet : $f_{\varphi}(u,v) = (v, (\varphi(v,k) \oplus u)) \Rightarrow f_{\varphi}^{-1}(x,y) = ((\varphi(x,k) \oplus y), x)$

Dans le cas du DES, seules quelques fonctions particulières sont utilisées parmi toutes les S Boxes possibles, car seules ces 8 fonctions particulières assurent une parfaite solidité du cryptosystème.

1.2 Les S Boxes (Secret Boxes) :

Ce sont des fonctions de 6 bits sur 4 bits.

$$\begin{aligned} \text{S Boxe :} \quad & \{0,1\}^6 \rightarrow \{0,1\}^4 \\ & (x,y,z,t,u,v) \rightarrow (a,b,c,d) \end{aligned}$$

Il en existe 2^{256} fonctions différentes. Seulement les concepteurs du DES (IBM), n'en ont retenues que quelques unes, parce que pour les autres choix le cryptosystème était très vulnérable.

Les critères qui ont motivé ces choix pertinents d'IBM, restent inconnus ...

Le rôle joué par ces S boxes est essentiel : Il détruit la structure et rend donc des attaques par permutations circulaires parfaitement désuètes.

1.3 La solidité du DES :

- La richesse de la structure de Feistel (Un très grand nombre d'implantations possibles, grâce à la liberté de choix de ϕ ...). En plus l'ensemble des structures de Feistel est dense dans l'ensemble des fonctions aléatoires.
- Les S boxes restent un grand point d'interrogation. Elle permettent des attaques réussies par cryptanalyse différentielle pour des itérations à 8 étapes (Shamir). La non linéarité qu'elles introduisent étant assurés par un petit nombre de variables, peut permettre une cryptanalyse partielle, en figeant ces quelques variables (Matsui).
- La longueur de la clef peut permettre un balayage systématique de toutes les clefs possibles à un coût très abordable (expérience réussie dans un campus d'université en juin 1997)

1.4 Le DES en pratique :

Le DES a été réalisé pour être réalisé sur un circuit câblé. Néanmoins ses applications logicielles sont très performantes et donnent des réalisations performantes grâce aux précalculs. Ce cryptosystème est notamment utilisé dans les cartes bancaires, les DAB et dans la compensation bancaire.

1.5 Les modes d'opération du DES :

- Mode ECB : Chiffrement indépendant des blocs.
- Mode CFB/OFB : Chiffrement chaîné des blocs.
- Mode MAC : Le premier bloc est chiffré avec la clef, le deuxième avec le premier crypto etc. Le dernier crypto est la signature authentifiante du texte.

2. IDEA (International Data Encryption Algorithm) :

2.1 Description :

Il s'agit d'un algorithme par bloc de 64 bits en entrée et en sortie, avec une clef de 128 bits. Il comporte 8 itérations identiques, composées de transformations inversibles, plus une transformation finale.

2.2 Opérations :

Les données sur 64 bits sont subdivisées en 4 blocs de 16 bits chacun. Quatre opérations sur ces blocs, interviennent lors des itérations :

- XOR : Il s'agit du OU EXCLUSIF bit par bit, sans propagation de retenue.
- AND : Il s'agit du ET bit par bit.
- ADDMOD : Sur des entiers de 16 bits, il s'agit de l'addition modulo 2^{16} , qui est inversible.
- PRODMOD : On travaille dans le corps fini à $2^{16}+1$ éléments. Avec la convention de remplacer par 2^{16} , toute valeur nulle. Ainsi tous les éléments sont inversibles.

2.3 Algorithme :

- Entrée : $M = m_1, \dots, m_{64}$; $K = k_1, \dots, k_{128}$.
- Sortie : $Y = (Y_1, Y_2, Y_3, Y_4)$
- Calculer les clefs partielles de 16 bits chacune intervenant dans les itérations et la transformation finale : A chaque itération on partitionne la clef en blocs de 16 bits et on lui inflige une rotation vers la gauche de 25 bits.
- Calculer $X = (X_1, X_2, X_3, X_4)$, les 4 blocs de 16 bits consécutifs.
- **POUR** r allant de 1 à 8 **FAIRE** :
 - | $X_1 = \text{PRODMOD} (X_1, K_1^{(r)})$.
 - | $X_4 = \text{PRODMOD} (X_4, K_4^{(r)})$.
 - | $X_2 = \text{ADDMOD} (X_2, K_2^{(r)})$.
 - | $X_3 = \text{ADDMOD} (X_3, K_3^{(r)})$.
 - | $t_0 = \text{PRODMOD} (K_5^{(r)}, \text{XOR} (X_1, X_3))$.
 - | $t_1 = \text{PRODMOD} (K_6^{(r)}, \text{ADDMOD} (t_0, \text{XOR} (X_2, X_4)))$.
 - | $t_2 = \text{ADDMOD} (t_0, t_1)$.
 - | $X_1 = \text{XOR} (X_1, t_1)$.
 - | $X_4 = \text{XOR} (X_4, t_2)$.
 - | $a = \text{XOR} (X_2, t_2)$.
 - | $X_2 = \text{XOR} (X_3, t_1)$.
 - | $X_3 = a$.
- **FIN FAIRE**
- La transformation de sortie est :
 - $Y_1 = \text{PRODMOD} (X_1, K_1^{(9)})$.
 - $Y_4 = \text{PRODMOD} (X_4, K_4^{(9)})$.
 - $Y_2 = \text{ADDMOD} (X_2, K_2^{(9)})$.
 - $Y_3 = \text{ADDMOD} (X_3, K_3^{(9)})$.

2.4 Solidité :

- Il existe 2^{64} clefs menant d'un clair donné à un crypto donné, mais elles sont impossible à caractériser.
- Une attaque par essais systématiques des 2^{128} clefs, est encore hors de portée et pour plusieurs années encore.
- On a établi la fragilité de quelques clefs.

3. AES (Advanced Encryption System) de RIJNDAEL :

3.1 Description (non détaillée) :

Successeur du DES, il s'agit d'un algorithme à 8 itérations, plus une transformation finale. Les données sont représentées sous forme d'une matrice à coefficients dans GF(8). Chacune des itérations fait intervenir ces 4 fonctions :

- Bytesub : Transformation sur les octets (inversion + transformation linéaire affine).
- Shiftrow : Permutation de lignes.
- Mixcolumn : Permutation sur les colonnes.
- Addroundkey : Transformation qui fait intervenir la clef.

3.2 Solidité :

- Haute diffusion.
- Résistance aux attaques par linéarité de Matsui.