La Cryptanalyse Linière

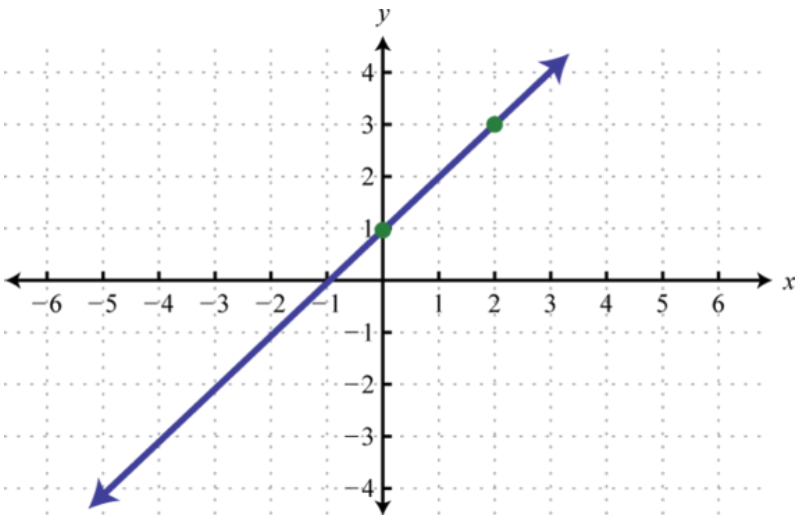
Module : Sécurité Informatique
Groupe : 1
Membres :

- Benkedadra Mohamed
- Benkorreche Mohamed El Amine

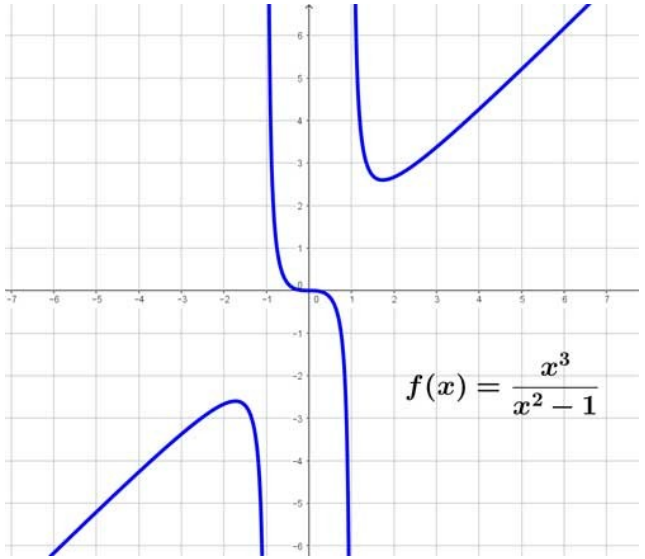
LINÉARITÉ

La Linéarité:

- La Linéarité en cryptographie est un concept un peux difficile a comprendre
- Quelque chose est linière donc quelque chose est prédictible
- Exemples :



Graphe 01



Graphe 02

MANIPULATION DES BITS

Le Masquage:

- Un maque : est un nombre en bit qui est utilisé pour choisir un sous ensemble d'un nombre en bit du même taille
- Chaque nombre de n-bit a $2^n - 1$ masques possibles (nb de 4bit a $2^4 - 1$ donc 15)
- Un nombre est masqué a l'aide des opérations binaires (AND, OR, XOR ..etc)
- On prend des 1 pour les cases qu'on veux et des 0 pour les autres.
- Un exemple de masquage :
    - x = 101101101001 → longeur = 12 bits → 4095 masques possibles
    - si on veux prendre le sous ensemble des cinq premiers nombres a gauche, on utilise l'opération AND entre le masque y et la valeur x :

      y = 111110000000

      d'où :

      x AND y = **10110**0000000

PARITÉ

La Parité:

- La parité d'un nombre est un bit, utilisé pour être sure qu'un nombre est pair ou impair
- Pour la calculer :
  - 1re méthode :
    - On transforme le nombre en base 2 (en bit)
    - On compte le nombre des 1, la parité égale à :
      - 1 si le nombre compter est impair
      - 0 sinon
    - Par exemple : 10110 → p = 1
  - 2eme méthode :
    - On fait un ⊕ (xor) entre chaque pair de bit
    - Par exemple :

$10 \rightarrow 1 \oplus 0 = 1$

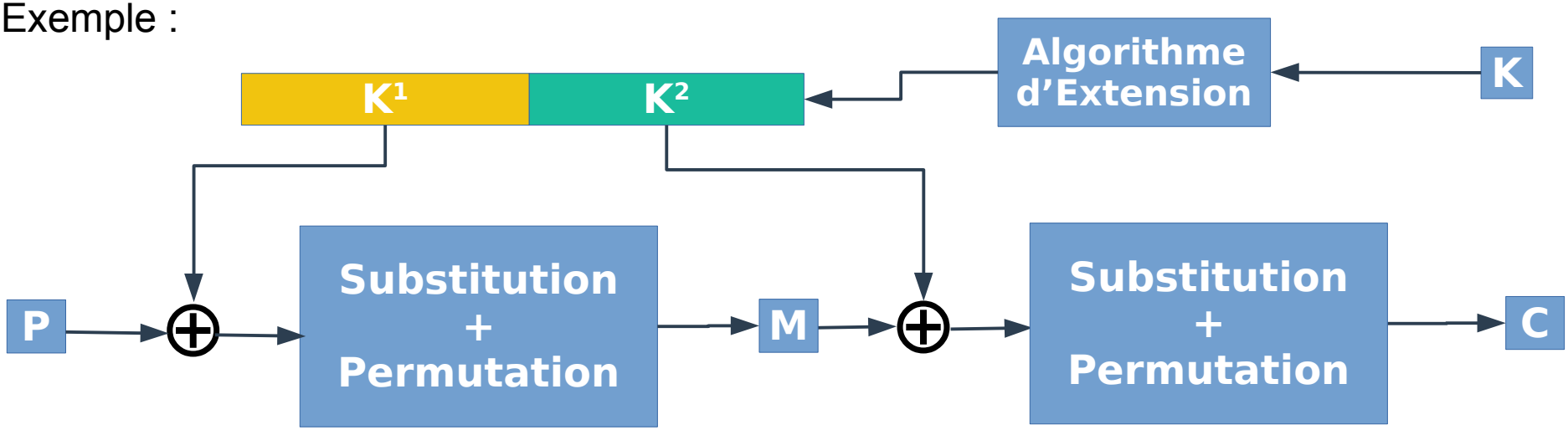
$1010 \rightarrow (1 \oplus 0) \oplus (1 \oplus 0) = 0$

$11100 \rightarrow ((1 \oplus 1) \oplus (1 \oplus 0)) \oplus 0 = 1$

CHIFFREMENT PAR BLOC

SPN:(Réseaux de Substitution-Permutation - Substitution-Permutation Network)

- Une des méthodes les plus facile pour approcher le chiffrement par bloc
- Existence R-1 messages semi-chiffré (M)
- Exemple :



- Fonctions des Rounds :
  - $W^1 = G(W^0, K^1) \Rightarrow M = P\text{-}BOX[\ S\text{-}BOX\ [\ P \oplus K^0\ ]\ ]$
  - $W^2 = G(W^0, K^2) \Rightarrow C = P\text{-}BOX[\ S\text{-}BOX\ [\ M \oplus K^1\ ]\ ]$

$$W^r = P\text{-}BOX\ [\ S\text{-}BOX\ [\ W^{r-1}, K^r\ ]\ ]$$

Cryptanalyse

Cryptanalyse:

- une technique dans le domaine de cryptographie, qui consiste de casser un algorithme
- Il y existe quatre type d'attaque dans la cryptanalyse :
  - Attaque sur un texte chiffré seul : l'attaquer a accès a des exemples du texte chiffré. Et essaye de trouvé le texte clair a l'aide de quelques informations.
  - Attaque à texte clair connu : l'attaquer a accès a des exemples de texte clair et leurs textes chiffrés, et il essaye de trouvé la clé de chiffrement (comme la cryptanalyse linière).
  - Attaque à texte clair choisi : l'attaquer peux choisir du texte clair et le chiffré au but de trouvé des méthodes pour faiblir l'algorithme
  - Attaque à texte chiffré choisi : l'attaquer peux choisir du texte chiffré puis demande son texte clair. Après, il les utilise afin de faiblir l'algorithme (l'inverse de l'attaque a texte clair choisi avec le même but).

Cryptanalyse Linière

Cryptanalyse Linière:

- Introduit par « Mitsuru Matsui » lors du « EUROCRYPT » de 1993, comme une meilleur méthode pour cassé le DES.
- Appartient au « Attaques a Texte Clair Connu »
- Essaye de trouvé la clé de chiffrement a l'aide des exemplaires P/C (Clair/Chiffré)
- Utilisé généralement avec les algorithmes de chiffrement par bloc
- Seul, elle n'ai pas très efficace pour les algorithmes de chiffrement moderne
- Idéal pour la cryptanalyse académique, ou en sais tous sur l'algorithme.
- Consiste de :
  - Trouver les parties non-linières de l'algorithme de chiffrement par bloc
  - Introduire la linéarité au parties non-linières a l'aide des approximations linières
  - Augmenter la précision de chaque approximation linière.
  - Trouver les meilleurs approximations pour chaque partie non-linière du l'algorithme

Cryptanalyse Linière

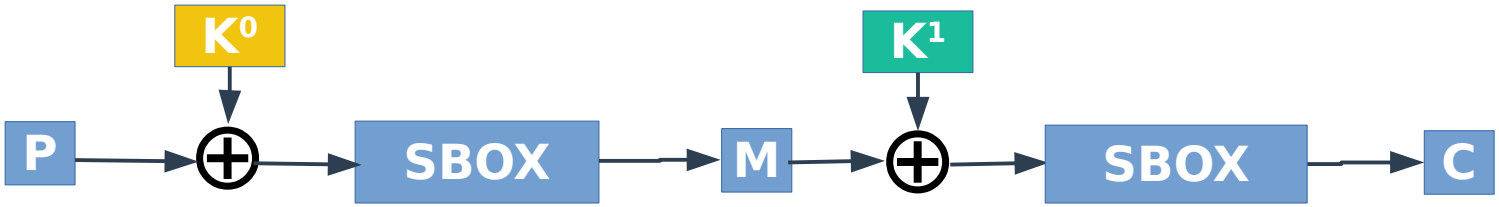
Implémentation:

- Bien analyser l'algorithme, et comprendre comment il marche.
- Trouver toutes les parties (opérations) non-linières .
- Pour chaque partie-non linière :
    - Faire des approximations linières .
    - Trouvé la meilleur approximation .
- Pour chaque round $W^R = G (W^{R-1} , K^R)$:
    - trouver les messages semi-chiffrés M pour chaque K^r des K^r possible.
    - Trouvé les meilleurs K^r (le K^r qui calcule le plus grand nombre de M correctement)
    - Essayer de deviner K^{r+1}
- Après qu'on trouve tous les K^r :
    - Tester tous les couples P/C avec l'ensemble (K^0, K^1, …., K^r)
    - Si tous les couples sont correcte donc les clés sont correcte.

Cryptanalyse Linière

Exemple :

- prenant l'algorithme suivant :



les P, K, C, M … etc ont une longueur de 4bit

disant qu'on a 16 P et leurs 16 C , chiffré a l'aide de l'algorithme

- 1- analyse de l'algorithme :
  - XOR est linière, parce que :

    $A = B \oplus C \rightarrow B = A \oplus C \rightarrow C = A \oplus B$ d'où le xor est prédictible
  - La seul partie non-linière est le S-BOX. d'où, il faut qu'on fait l'approximation Linière du S-BOX.

Cryptanalyse Linière

Approximation Linière :

- Considérant le S-BOX suivant :
- On trouve tous les masques possible du S-BOX,

  dans notre cas, l'entrée du S-BOX est de 4 bit, donc les masques d'entrée possibles sont [0001,0010,….,1111] , et la même chose pour les masques de sortie .

- On dessine un tableau ou les lignes sont les masques d'entrées et les colonnes sont les masques de sorties, et on initialise les cases par 0.

S-BOX :

0 → 9	8 → 13
1 → 11	9 → 7
2 → 12	10 → 3
3 → 4	11 → 8
4 → 10	12 → 15
5 → 1	13 → 14
6 → 2	14 → 0
7 → 6	15 → 5

- On masque chaque entrée possible (de 0000 a 1111) par un masque d'entrée et on calcule ça parité p1. On masque ça sortie par un masque de sortie et en calcule ça parité p2.

  si p1 égale a p2 → ajouter 1 dans la case [masque entrée][masque sortie]

  sinon → rien faire

Cryptanalyse Linière

	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	
01	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
02	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
03	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
04	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
05	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
06	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
07	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
08	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
09	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
10	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
11	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
12	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
13	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
14	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	
15	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	

Tableau d’approximation Initial

Cryptanalyse Linière

	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	
01	08	08	08	10	10	06	06	06	10	06	10	12	08	08	12	
02	04	06	06	08	08	10	06	04	08	10	10	08	08	10	06	
03	08	10	06	06	10	08	08	06	10	08	08	04	04	06	10	
04	06	10	12	08	06	06	08	06	08	04	10	06	08	08	06	
05	06	10	04	06	04	08	10	08	06	06	08	10	08	08	10	
06	06	08	06	08	10	08	10	10	12	06	08	06	12	10	08	
07	10	12	06	10	08	10	08	08	10	08	10	10	08	06	04	
08	10	08	06	10	04	06	04	08	10	08	06	06	08	10	08	
09	10	08	06	08	10	08	06	06	04	06	08	06	12	06	08	
10	06	10	08	14	08	08	10	08	06	10	08	06	08	08	10	
11	10	06	08	08	06	10	08	10	08	08	14	06	08	08	10	
12	08	10	10	06	06	08	08	06	10	12	08	08	12	06	10	
13	08	10	10	08	08	14	06	08	08	06	06	08	08	10	10	
14	08	04	08	10	06	10	10	06	10	06	06	08	08	04	08	
15	04	08	08	08	08	08	04	12	08	08	08	08	08	04	08	

Tableau d’approximation final

Cryptanalyse Linière

Meilleur Approximation Linière :

- On trouve la meilleur approximation par trouvé la case qui contient la plus grande valeur. On prend ça colonne et ça ligne comme « meilleur approximation ».
- Par exemple dans notre cas :
    - La valeur 14 est la plus grande

      Les cases ([masque entrée][masque sortie]) contenant cette valeur sont :

      [11][11]

      [10][04]

      [13][06]

      on prend une. par exemple [11][11].
- Remarque :

  si on utilise XOR dans notre algorithme, il est conseiller de prendre aussi les cases contenant les valeur ≤ (nombre des entrées possibles – plus grande valeur trouvé)

  dans ce cas par exemple, (16 – 14) = 2 , alors tous les valeurs ≤ 2

Cryptanalyse Linière

Attaque Linière :

- On commence par prendre tous les valeurs possibles de K^0
- On dessine un tableau a une dimension, la taille du tableau = nombre de K^0 possibles.
- initialiser chaque case par le 0
- Pour chaque K^0 :
    - On calcule la série des M des P qu'on a déjà
    - On passe chaque M dans le S-BOX
    - On masque chaque M par le masque d'entrée de la meilleur approx et en calcule la parité p1
    - On masque chaque sortie de M du S-BOX par le masque de sortie de la meilleur approx et en calcule la parité p2
        - Si p1 égale à p2 on ajoute 1 dans la case du K^0 utiliser pour la calculer
        - Sinon on soustrait 1 de la case du K^0 utiliser pour la calculer

Cryptanalyse Linière

→ Tableau Initial des Score des K^0 : [0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

→ Tableau final des Score des K^0 : [0, 4, 0, 12, -4, 0, -12, 0, 0, -12, 0, -4, 12, 0, 4]

- On trouve la plus grande valeur dans le tableau : dans ce cas c'est la valeur 12
- Les meilleurs K^0 sont les cases qui ont la valeur 12, donc 3 et 12
- Pour chaque K^0 trouvé, on calcule un M a l'aide d'un des P qu'on a déjà(P4 par exemple). Maintenant, on a un M calculer a l'aide de $K^0 = 3$ et un autre a l'aide de $K^0 = 12$.
- Pour chaque M calculer on calcule K^1 a l'aide du C du P utiliser pour calculer ce M (dans notre cas, c'est C4)

K^1 = S-BOX-INVERSE (C4) ⊕ M = 8 → pour $K^0 = 3$

K^1 = S-BOX-INVERSE (C4) ⊕ M = 5 → pour $K^0 = 12$

d'où on as trouvé K = (3,8) ou K = (5,12)

- Pour chaque couple K on chiffre tous Les P qu'on a déjà et on teste si il sont égaux a leurs C. Si oui, on a trouvé le couple (K^0,K^1). Si non, l'attaque linière a échoué.
- Dans notre cas, on trouve K = (3,12) .

Cryptanalyse Linière

Code :

- Python : https://github.com/LogX7/linear-crypto
- C : http://theamazingking.com/linear1.c

Ressources :

- The Amazing King Blog : theamazingking.com
- Wikipedia : Linière Cryptanalysis / Cryptanalysis / Block Ciphers.
- StackExchange : crypto.stackexchange.com