Sécurité des chiffres symétriques en bloc

Bruno MARTIN,
Université Côte d'Azur

Rappels de probabilités

Probabilités

- **Ensemble fondamental :** S ensemble fini dont les éléments sont des événements élémentaires assimilables à la réalisation d'une expérience
- **Evénement :** sous-ensemble de S :
  - $\varnothing$ est l'événement vide
  - S l'événement certain
- Deux événements sont en **exclusion mutuelle** ssi leur intersection est vide

Distribution de probabilités, V.A.

Fonction de $X \subset S \to [0,1] =$ **distribution de probabilités** si :

1. $\forall$ événement $A \in X$, $Pr(A) \geq 0$
2. si $A \cap B = \varnothing$, $Pr(A \cup B) = Pr(A) + Pr(B)$
3. Pr(S)=1

$X : S \to \mathbb{R}$ une **V.A. discrète** si l'événement $x \in X$ est l'ensemble $\{s \in S : X(s) = x\}$; la proba associée est :

$$Pr(X = x) = \sum_{s \in S : X(s) = x} Pr(s)$$

Exemple

Une paire de D6, V.A. = maximum des valeurs. $Pr(X = 3) = \frac{5}{36}$

Dé1 = 3 Dé2=1,2,3 ou Dé2=3 Dé1=1,2,3 on retire Dé1=3=Dé2.

Probabilité conditionnelle et indépendance

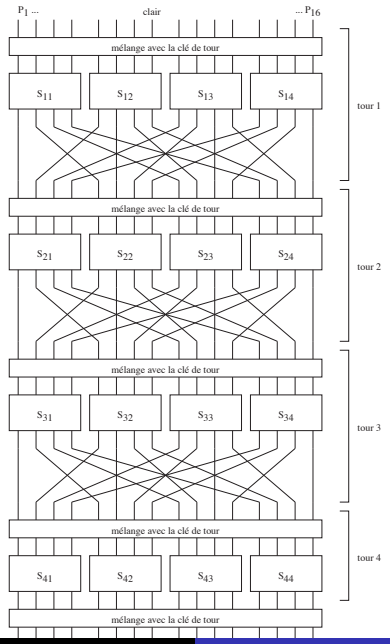
$$Pr(A|B) = \frac{Pr(A \cap B)}{Pr(B)}$$

- A, B 2 événements indép. ssi $Pr(A \cap B) = Pr(A).Pr(B)$; si $Pr(B) \neq 0$, A, B indép. $\Leftrightarrow Pr(A|B) = Pr(A)$
- Formule de Bayes :

$$\Pr(A \cap B) = Pr(B \cap A) = Pr(B)Pr(A|B) = Pr(A)Pr(B|A)$$

et si $Pr(B) \neq 0$, on a

$$Pr(A|B) = \frac{Pr(A)Pr(B|A)}{Pr(B)}$$



Cryptanalyse différentielle

Présentation du chiffre utilisé

Réseau de substitution/permutation simplifié [Heys, 2002].
Entrée : 16 bits ; fonctionne en 4 tours constitués de :

- **mélange** : xor entre la clé de tour et le bloc d'entrée du tour. Opération répétée à l'issue du dernier tour.
- **substitution**. 16 bits scindés en 4 sous-blocs qui entrent dans 4 boîtes-S identiques. (MSB à gauche)

in	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
out	E	4	D	1	2	F	B	8	3	A	6	C	5	9	0	7

- **transposition** : relie *la i^e sortie de la j^e boîte-S à la j^e entrée de la i^e boîte-S*

Boîte S

0	0000	1110	e	8	1000	0011	3
1	0001	0100	4	9	1001	1010	a
2	0010	1101	d	a	1010	0110	6
3	0011	0001	1	b	1011	1100	c
4	0100	0010	2	c	1100	0101	5
5	0101	1111	f	d	1101	1001	9
6	0110	1011	b	e	1110	0000	0
7	0111	1000	8	f	1111	0111	7

Bref historique

- Inventée à la fin des années 1980
  par [Biham and Shamir, 1990] pour la cryptanalyse de DES (succès mitigé).
- DES conçu pour résister à la cryptanalyse différentielle ?
- vulnérabilité de chiffres de la même époque (p.e. FEAL)
- présentation de la cryptanalyse différentielle de [Heys, 2002].

Principe de la cryptanalyse

$n = 16$ entrées $X = [X_1 \dots X_n]$, n sorties $Y = [Y_1 \dots Y_n]$;
Différences $\Delta X = X' \oplus X''$ (entrées) et $\Delta Y = Y' \oplus Y''$ (sorties).
On étudie proba d'apparition d'occurrences de ΔX et de ΔY à l'entrée du dernier tour du chiffre. $(\Delta X, \Delta Y) \stackrel{\Delta}{=}$ **différentiel**.
On exploite les apparitions d'un ΔY particulier pour un certain ΔX avec forte proba p_D.
Chiffre idéal : $\Pr(\Delta Y | \Delta X)$ devrait valoir $1/2^n$.
Attaque CPA : Choix de paires d'entrées (les **bonnes paires**) $X', X'' = \Delta X$ tq le ΔX considéré mène avec une forte proba à un ΔY particulier.

Principe de l'attaque

Trouver les $(\Delta X, \Delta Y)$ les plus probables :

1. examiner les propriétés des boîtes-S et construire *la table des différentiels* qui résume toutes les probabilités
2. considérer les ΔX et ΔY des boîtes-S pour maximiser les probabilités associées
3. combiner l'information sur les boîtes-S pour construire une approximation globale (caractéristique différentielle)

Analyse d'une boîte S -1-

On examine tous les $(\Delta X, \Delta Y)$ et on cherche la proba d'apparition de ΔY étant donné ΔX fixée :

- énumérer 16 valeurs pour X',
- $X'' = X' \oplus \Delta X$ est fixé par la valeur de ΔX fixée

Exemple

on cherche les valeurs les plus fréquentes de ΔY pour chaque paire $(X', X' \oplus \Delta X)$ pour les valeurs $\Delta X = 1011, 1000, 0100$ (0xB, 0x8, 0x4). Celles qui apparaissent avec le plus d'écart sont $\Delta Y = 0010$ pour $\Delta X = 1011$ (8/16), $\Delta Y = 1011$ pour $\Delta X = 1000$ (4/16) et $\Delta Y = 0110$ pour $\Delta X = 0100$ (6/16).

Détail des calculs pour $\Delta X = 1011$

Pour ΔX=1011

hex	dec	bin	image	ΔX = 11		ΔY
X			S(X)	X'=X XOR ΔX	S(X')	S(X) XOR S(X') bin
0	0	0000	14	11	12	2 0010
1	1	0001	4	10	6	2 0010
2	2	0010	13	9	10	7 0111
3	3	0011	1	8	3	2 0010
4	4	0100	2	15	7	5 0101
5	5	0101	15	14	0	15 1111
6	6	0110	11	13	9	2 0010
7	7	0111	8	12	5	13 1101
8	8	1000	3	3	1	2 0010
9	9	1001	10	2	13	7 0111
a	10	1010	6	1	4	2 0010
b	11	1011	12	0	14	2 0010
c	12	1100	5	7	8	13 1101
d	13	1101	9	6	11	2 0010
e	14	1110	0	5	15	15 1111
f	15	1111	7	4	2	5 0101

Exemple (suite) – synthèse pour les 3 ΔX

X'	Y	ΔY		
		$\Delta X = 1011$	$\Delta X = 1000$	$\Delta X = 0100$
0000	1110	0010	1101	1100
0001	0100	0010	1110	1011
0010	1101	0111	1011	0110
0011	0001	0010	1101	1001
0100	0010	0101	0111	1100
0101	1111	1111	0110	1011
0110	1011	0010	1011	0110
0111	1000	1101	1111	1001
1000	0011	0010	1101	0110
1001	1010	0111	1110	0011
1010	0110	0010	1011	0110
1011	1100	0010	1101	1011
1100	0101	1101	0111	0110
1101	1001	0010	0110	0011
1110	0000	1111	1011	0110
1111	0111	0101	1111	1011
		8	4	4

Analyse d'une boîte S –2–

Boîte-S parfaite, tous les éléments du tableau devraient être égaux à 1 et les proba = 1/16.

La **table des différentiels** résume toutes les possibilités. Chaque case représente le nombre d'occurrences de ΔY étant donnée ΔX.
La somme par ligne ou par colonne vaut $2^n = 16$.
Tous les éléments sont pairs (la différence est symétrique).
Si ΔX=0, ΔY aussi.

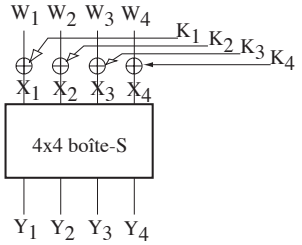
Exemple

Maximale pour ΔX=B et ΔY=2.
La proba pour que ΔY=2 pour une paire d'entrées telle que ΔX=B est de 1/2.

Table des différentiels

$\Delta Y >$	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	2	0	0	0	2	0	2	4	0	4	2	0	0
2	0	0	0	2	0	6	2	2	0	2	0	0	0	0	2	0
3	0	0	2	0	2	0	0	0	0	4	2	0	2	0	0	4
4	0	0	0	2	0	0	6	0	0	2	0	4	2	0	0	0
5	0	4	0	0	0	2	2	0	0	0	4	0	2	0	0	2
6	0	0	0	4	0	4	0	0	0	0	0	0	2	2	2	2
7	0	0	2	2	2	0	2	0	0	2	2	0	0	0	0	4
8	0	0	0	0	0	0	2	2	0	0	0	4	0	4	2	2
9	0	2	0	0	2	0	0	4	2	0	2	2	2	0	0	0
A	0	2	2	0	0	0	0	0	6	0	0	2	0	0	4	0
B	0	0	8	0	0	2	0	2	0	0	0	0	0	2	0	2
C	0	2	0	0	2	2	2	0	0	0	0	2	0	6	0	0
D	0	4	0	0	0	0	0	4	2	0	2	0	2	0	2	0
E	0	0	2	4	2	0	0	0	6	0	0	0	0	0	2	0
F	0	2	0	0	6	0	0	0	0	4	0	2	0	0	2	0

Influence de la clé sur un différentiel



X entrée de la boîte-S (sans action de la clé) Y la sortie.
$W = X \oplus K$ entrée boîte-S avec l'action de la clé K,

Evaluons $\Delta W_i = W_i' \oplus W_i''$:
$W_i' \oplus W_i'' = (X_i' \oplus K_i) \oplus (X_i'' \oplus K_i) = X_i' \oplus X_i'' = \Delta X_i$ (car $K_i \oplus K_i = 0$)
Les bits de clé n'ont pas d'influence sur les différentiels.

Construire la caractéristique différentielle du chiffre

Une fois obtenue la table des différentiels des boîtes-S, il faut construire la caractéristique différentielle du chiffre par concaténation des paires de différences adéquates entre les boîtes-S.

Exemple

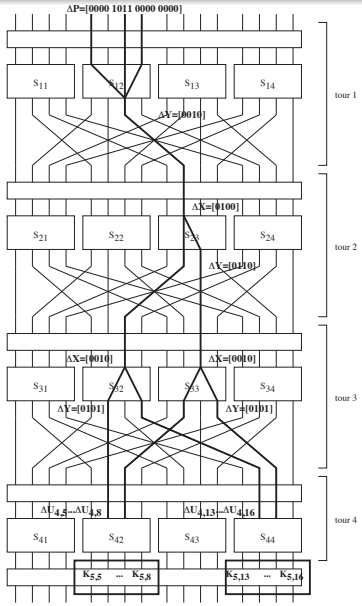
On utilise les paires de différences suivantes :

$$S_{12} : \Delta X = B \rightarrow \Delta Y = 2 \quad \text{proba } 8/16$$
$$S_{23} : \Delta X = 4 \rightarrow \Delta Y = 6 \quad \text{proba } 6/16$$
$$S_{32} : \Delta X = 2 \rightarrow \Delta Y = 5 \quad \text{proba } 6/16$$
$$S_{33} : \Delta X = 2 \rightarrow \Delta Y = 5 \quad \text{proba } 6/16$$

Différence à l'entrée du chiffre : $\Delta P = [0000\ 1011\ 0000\ 0000]$; à l'entrée du dernier tour, $\Delta U_4 = [0000\ 0110\ 0000\ 0110]$ de proba $(6/16)^2$ étant donné ΔU_3 de proba $6/16$, étant donné ΔU_2 de proba $8/16$ étant donné ΔP.
Sachant qu'on a entré ΔP, la proba d'avoir

$$\Delta U_4 = [0000\ 0110\ 0000\ 0110] \qquad (1)$$

est le produit des probas : $(6/16)^2(6/16)(8/16) = \frac{27}{1024} = 0,026$.
(On suppose l'indépendance).



Bonnes et mauvaises paires

Lors du calcul de la caractéristique différentielle, les couples (X', X'') qui donnent le ΔY maximal sont des **bonnes paires**. [Beneteau, 2010, King,]

Exemple

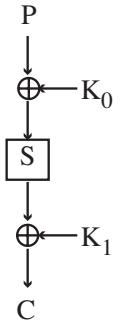
Pour $(\Delta X, \Delta Y) = (b, 2)$, *les bonnes paires sont :*

$$X' = (0,1,3,6,8,a,b,d) \quad X'' = (b,a,8,d,3,1,0,6)$$
$$Y' = (e,4,1,b,3,6,c,9) \quad Y'' = (c,6,3,9,1,4,e,b)$$

Les autres sont des **mauvaises paires**.

Choix des clairs – chiffre simplifié –

P_0 choisi au hasard et fixe la valeur $P_1 = P_0 \oplus \Delta X = 1011$.
On calcule $C_0 \oplus C_1$, $C_0 = \{P_0\}_{K_0 K_1}$ $C_1 = \{P_1\}_{K_0 K_1}$



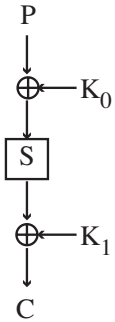
$P_0 = b = 1011$ $\quad$ $P_1 = 0 = 0000$

$C_0 = a = 1010$ $\quad$ $C_1 = 8 = 1000$

Si $C_0 \oplus C_1 = \Delta Y = 0010$, on a trouvé une bonne paire P, C (sinon on recommence).

Retrouver la clé

On "devine" que l'entrée de la boîte-S est 3 pour P_0 (choisi dans les 8 bonnes paires), d'où l'entrée correspondante 8 de P_1. (sinon, il reste 7 choix possibles)



$P_0 = b = 1011$ $\quad$ $P_1 = 0 = 0000$

$3 = 0011$ $\quad$ $8 = 1000$

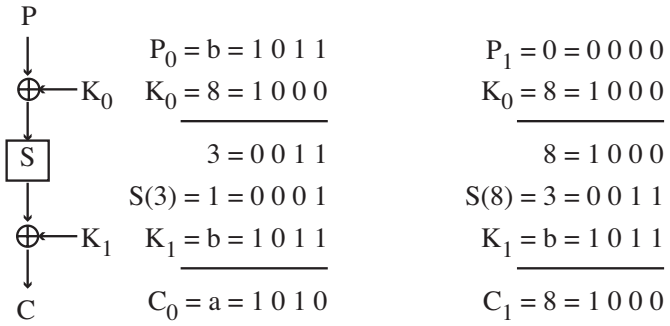
$S(3) = 1 = 0001$ $\quad$ $S(8) = 3 = 0011$

$C_0 = a = 1010$ $\quad$ $C_1 = 8 = 1000$

On déduit que $P_0 + K_0 = 3 \Rightarrow K_0 = 3 + P_0 = 1000 = 8$
Et que $C_0 = S(P_0 + K_0) + K_1 \Rightarrow K_1 = 1 + C_0 = 1011 = b$

Vérification

On vérifie $K_0 = 1000$ et $K_1 = 1011$ compatibles avec d'autres (P, C)



Pour un mauvais choix d'entrée, il y en a encore 7 autres possibles.
Le nombre de choix de la clé à faire équivaut à la proba du différentiel. Ici il y avait 8 clés possibles ($p_D = 8/16$). Utiliser une p_D inférieure aurait rendu la recherche plus courte mais trouver une bonne paire aurait été plus difficile.

En Python

```
def EM(P,K):
    S=[14,4,13,1,2,15,11,8,3,10,6,12,5,9,0,7]
    Kb=bin(K)[2:].zfill(8)
    K0,K1=int(Kb[:4],2),int(Kb[4:],2)
    P1=int(hex(P),16)^int(hex(K0),16)
    P2=int(hex(S[P1]),16)^int(hex(K1),16)
    return P2

C0 = bin(EM(11,139))[2:].zfill(4)
C1 = bin(EM(0,139))[2:].zfill(4)
hex(int(C0,2)),hex(int(C1,2))
> ('0xa', '0x8')
bin(int(C0,2)^int(C1,2))[2:].zfill(4)
> '0010'
```


Retrouver des bits de la clé – chiffre SPN –

Avec la caractéristique différentielle de $R-1$ tours d'un chiffre à R tours et une proba suffisante, on tente de retrouver des bits de la dernière clé de tour (ici K_5). Ils sont déterminés par les 2 dernières boîtes-S du dernier tour et les bits obtenus par la caractéristique différentielle.
On fait une cryptanalyse exhaustive partielle du dernier tour : pour toutes les valeurs possibles des bits cherchés de la dernière clé de tour, on xor avec les bits correspondants du chiffré et on les fait passer à l'envers dans les deux boîtes-S concernées pour obtenir les entrées des boîtes-S au dernier tour.
On effectue cette opération pour un grand nombre de clairs/chiffrés et on compte le nombre de fois où la caractéristique différentielle est vérifiée.
La clé qui satisfait l'expression le plus souvent est la plus vraisemblable.
Les autres bits de la clé sont déterminés par recherche exhaustive.

Sur l'exemple

La caractéristique différentielle affecte les entrées de S_{42} et S_{44}.
Pour tous les couples clair/crypto, on cherche les 256 valeurs possibles des bits de K_5 concernés : $[K_{5,5} \dots K_{5,8}, K_{5,13} \dots K_{5,16}]$.
Pour chaque valeur de clé, on compte le nombre de fois où (1) est satisfaite en calculant les valeur de $[\Delta U_{4,5} \dots \Delta U_{4,8}, \Delta U_{4,13} \dots \Delta U_{4,16}]$.
On choisit la clé qui a maximisé cette valeur.
Expérimentation : avec 5000 couples, en appliquant la technique, la clé la plus vraissemblable est (en hexadécimal) 24.

Complexité de l'attaque

Quantité de données pour une attaque fructueuse : N_D clairs.
Expérimentalement, on évalue

$$N_D \approx c/p_D$$

où p_D est la probabilité de la caractéristique différentielle des $R-1$ tours et c une petite constante.

Cas du DES

Dans [Biham and Shamir, 1990], il est dit qu'un DES limité à 6 tours peut être attaqué avec succès en 0,3 s avec 240 clairs.
Le DES à 16 tours nécessite 2^{58} étapes de calcul (ce qui est supérieur à la recherche exhaustive).

Et la cryptanalyse linéaire ?

Cryptanalyse linéaire

- Schéma général comparable à la cryptanalyse différentielle
- attaque des boîtes S en cherchant des relations linéaires ou affines qui s'écartent d'une distribution uniforme.
- relation linéaire(/affine) entre un sous-ens des entrées et un sous-ens des sorties
- Problème : combiner ces relations (utilisation du pilling-up lemma de Matsui [Matsui, 1993])

Principe de la peau d'oignon

Outil de base pour concaténer les approximations linéaires des boîtes-S.
Soient X_1 et X_2 2 VA binaires, l'expression linéaire
$X_1 \oplus X_2 = 0 \Leftrightarrow X_1 = X_2$ et l'expression affine $X_1 \oplus X_2 = 1 \Leftrightarrow X_1 \neq X_2$.
Les distributions de proba sont : $\begin{cases} Pr(X_i = 0) = p_i \\ Pr(X_i = 1) = 1 - p_i \end{cases}$.
Les deux VA sont supposées indépendantes et

$$Pr(X_1 = i, X_2 = j) = \begin{cases} p_1 p_2 & i = j = 0 \\ p_1(1-p_2) & i = 0, j = 1 \\ (1-p_1)p_2 & i = 1, j = 0 \\ (1-p_1)(1-p_2) & i = j = 1 \end{cases}$$

$Pr(X_1 \oplus X_2 = 0) = Pr(X_1 = X_2) = p_1 p_2 + (1-p_1)(1-p_2) = \frac{1}{2} + 2\varepsilon_1\varepsilon_2 = \frac{1}{2} + \varepsilon_{1,2}$

si $p_i = 1/2 + \varepsilon_i$, ε_i est le biais de X_i et $\varepsilon_{1,2}$ le biais de $X_1 \oplus X_2 = 0$.

Généralisation à plusieurs VA

La propriété précédente se généralise à plusieurs variables.

Lemme

Soient $X_1, \ldots, X_n$ n VA binaires indépendantes.

$$Pr(X_1 \oplus \ldots \oplus X_n = 0) = \begin{cases} \frac{1}{2} + 2^{n-1} \prod_{i=1}^{n} \varepsilon_i \\ \frac{1}{2} + \varepsilon_{1,2,\ldots,n} \end{cases}$$

Observons que si $p_i = 0$ (resp. 1) pour tout i, alors
$Pr(X_1 \oplus \ldots \oplus X_n = 0)$ (resp. 1) et si un seul des $p_i = 1/2$,
$Pr(X_1 \oplus \ldots \oplus X_n = 0) = 1/2$.

Application au DES

- Plus efficace que la cryptanalyse différentielle : au départ, 2^{47} couples (P, C), améliorations jusqu'à 2^{43}.
- DES pas robuste à une cryptanalyse linéaire
- En savoir plus sur la cryptanalyse linéaire : voir [Matsui, 1993, Heys, 2002, Musa et al., 2003, Mansoori and Bizaki, 2007].

Théorie de Galois

Analogie avec $\mathbb{R}$

On construit $\mathbb{C}$ à partir de $\mathbb{R}$ en lui ajoutant une racine i, racine de $x^2+1=0$ qui n'a pas de racine dans $\mathbb{R}$. On note $\mathbb{C}=\mathbb{R}(i)$.

On part

- d'un corps de base $\mathbb{F}_2=(\{0,1\},+,.)$. Les éléments du corps sont $\{0,1\}$, il a deux opérations $+,.$ et vérifie que tout élément a un opposé pour $'+'$ et un inverse pour $'.'$
- d'un polynôme p **irréductible**, i.e. qui n'a pas d'autre diviseurs que 1 et lui-même dans $\mathbb{F}_2[x]$. De plus, ni 0 ni 1 ne sont racine de ce polynôme.

On construit ensuite un **corps d'extension** de $\mathbb{F}_2[x]/p=\mathbb{F}_2(\eta)$ si η est racine de p.

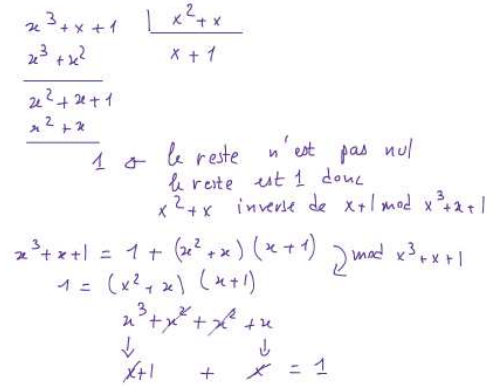
Exemple

On peut prendre $p(x)=x^3+x+1$ qui est irréductible.

Vérifier l'irréductibilité

On vérifie que $p(x)=x^3+x+1$ est irréductible :

- ni 0 ni 1 ne sont racine
- aucun des polynômes de degré inférieur ne le divisent : $\{x+1, x^2, x^2+1, x^2+x, x^2+x+1\}$



Eléments du corps d'extension

L'équation associée au polynôme $p(x)=x^3+x+1=0$ définit une relation de réécriture (d'équivalence) :

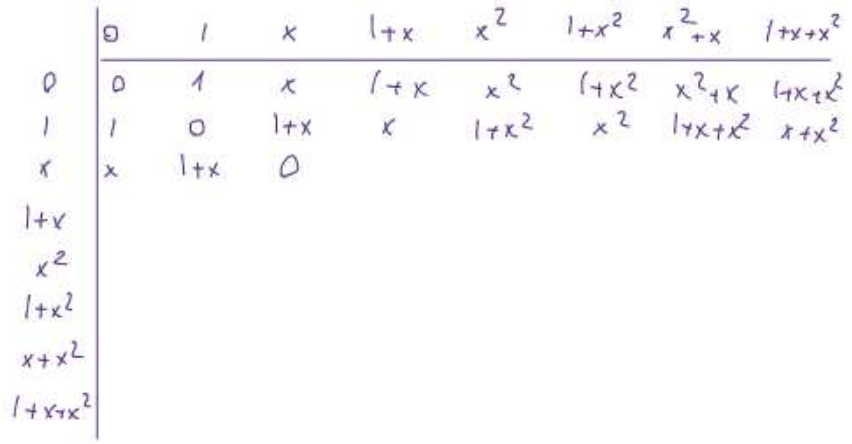
$$x^3=x+1$$

On construit l'extension $\mathbb{F}_2[x]/_{x^3+x+1=0}\simeq\mathbb{F}_8=GF(8)$ d'éléments :

polynôme	0	1	x	$x+1$	x^2	x^2+1	x^2+x	x^2+x+1
binaire	000	001	010	011	100	101	110	111

On vérifie qu'on a bien un corps en écrivant les tables de $'+'$ et $'.'$ sur ces éléments en faisant les calculs modulo p.

Addition



Produit

	0	1	x	$1+x$	x^2	$1+x^2$	x^2+x	$1+x+x^2$
0	0	0	0	0	0	0	0	0
1	0	1	x	$1+x$	x^2	$1+x^2$	x^2+x	$1+x+x^2$
x	0	x	x^2	$x+x^2$				
$1+x$	0	$1+x$	$x+x^2$					
x^2	0	x^2						
$1+x^2$	0	$1+x^2$						
$x+x^2$	0	$x+x^2$						
$1+x+x^2$	0	$1+x+x^2$						

Conclusion

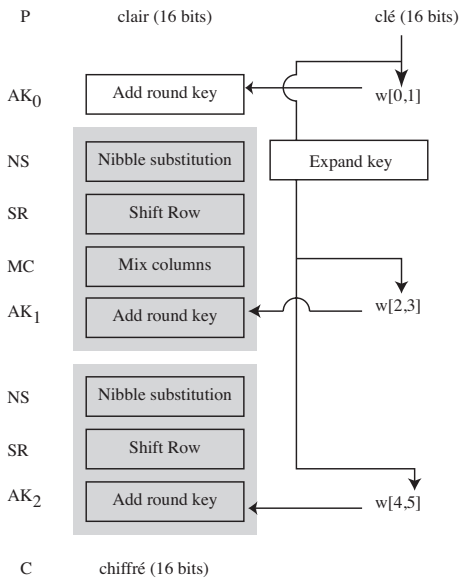
On a défini un corps d'extension à 8 éléments à partir de $\mathbb{F}_2$ et d'un polynôme irréductible de degré 3 (notez que $8 = 2^3$). Ce corps est unique (à isomorphisme près) et noté $\mathbb{F}_8$.
Ses éléments de base sont

polynôme	0	1	x	$x+1$	x^2	x^2+1	x^2+x	x^2+x+1
binaire	000	001	010	011	100	101	110	111

Les opérations internes sont définies en effectuant les calculs modulo le polynôme irréductible (en utilisant la règle de réécriture).

Construction algébrique de plus en plus utilisée en cryptographie.

(S)-AES



State matrix & nibbles

- 1 nibble = mot de 4 bits (E/S des composants de SAES)

$$\begin{array}{c|c} b_0 b_1 b_2 b_3 & b_8 b_9 b_{10} b_{11} \\ \hline b_4 b_5 b_6 b_7 & b_{12} b_{13} b_{15} b_{15} \end{array} = \begin{array}{c|c} S_{0,0} & S_{0,1} \\ \hline S_{1,0} & S_{1,1} \end{array}$$

- représentation de la clé :

$$\underbrace{k_0 k_1 \dots k_7}_{w[0]} \quad \underbrace{k_8 \dots k_{15}}_{w[1]}$$

Opérations dans $GF(16) \simeq \mathbb{F}_2[x]/x^4+x+1$

- $m(x) = x^4 + x + 1$ est un irréductible de $\mathbb{F}_2$
- éléments : nibble $b_0 b_1 b_2 b_3 \overset{\theta}{\leftrightarrow} b_0 x^3 + b_1 x^2 + b_2 x + b_3$
- addition : par addition des coefficients : $(x^3+x+1)+(x^2+1)$
- multiplication : produit des polynômes $\mod m(x)$
- codage octet : dans extension quadratique $\mathbb{F}_{16}[z]/z^2+1$
- attention ! z^2+1 non inversible dans $GF(16)$
- Rappel : trouver l'inverse d'un élément : Euclide étendu sur polynômes $(x+1, m) = (x^3+x^2+x)(x+1) + 1 \cdot m$

Inverses dans $\mathbb{F}_{16}$

1	0001	1	1	0001	1
2	0010	x	x^3+1	1001	9
3	0011	$x+1$	x^3+x^2+x	1110	e
4	0100	x^2	x^3+x^2+1	1101	d
5	0101	x^2+1	x^3+x+1	1011	b
6	0110	x^2+x	x^2+x+1	0111	7
7	0111	x^2+x+1	x^2+x	0110	6
8	1000	x^3	x^3+x^2+x+1	1111	f
9	1001	x^3+1	x	0010	2
a	1010	x^3+x	x^3+x^2	1100	c

Boîte S utilisée dans Nibble substitution

$i\downarrow$	00	01	10	11
00	9	4	a	b
01	d	1	8	5
10	6	2	0	3
11	c	e	f	7

$=$

1001	0100	1010	1011
1101	0001	1000	0101
0110	0010	0000	0011
1100	1110	1111	0111

- $\forall$ nibble : $\underbrace{b_0 b_1}_{\text{ligne}} \quad \underbrace{b_2 b_3}_{\text{colonne}}$: 00 01 $\overset{S}{\to}$ 01 00
-

$$\begin{array}{c|c} 0001 & 0001 \\ \hline 1100 & 1110 \end{array} \overset{S}{\to} \begin{array}{c|c} 0100 & 0100 \\ \hline 1100 & 1111 \end{array} = \begin{array}{c|c} 4 & 4 \\ \hline c & f \end{array}$$

Retrouver la boîte S algébriquement

1. initialiser la boîte S avec les nibbles rangés en tableau 1D ligne à ligne
2. convertir chaque nibble en polynôme
3. inverser chaque nibble dans $\mathbb{F}_{16}$
4. associer à l'inverse son polynôme dans $\mathbb{F}_2[y]/y^4-1=N(y)$
5. calculer $a(y)N(y)+b(y) \mod y^4+1$ avec $a=y^3+y^2+1$ et $b=y^3+1$

Exemple

$S(0011)=1011 \equiv S(3)=b$.

$S(0011)=1011 \equiv S(3)=b$

1. initialiser la boîte S avec les nibbles rangés en tableau 1D ligne à ligne $(b_0b_1b_2b_3=0011=x+1)$
2. convertir chaque nibble en polynôme $S(0011)=x+1$
3. inverser chaque nibbble dans $\mathbb{F}_{16}$: $(x+1)^{-1}=x^3+x^2+x$
4. associer à l'inverse son polynôme dans $\mathbb{F}_{16}[y]/y^4-1=N(y)$: $N(y)=y^3+y^2+y$
5. calculer $a(y)N(y)+b(y) \mod y^4+1$ avec $a=y^3+y^2+1$ et $b=y^3+1$:
   $(y^3+y^2+y)(y^3+y^2+1)+y^3+1 \mod y^4+1$
   $\cancel{y^6}+\cancel{y^5}+\cancel{y^3}+\cancel{y^5}+\cancel{y^4}+\cancel{y^2}+\cancel{y^4}+\cancel{y^3}+y+y^3+1$
   $\mod y^4+1 = y+y^3+1 = y^3+y+1$

Normalement $S(0011)=1011 \equiv S(3)=b$

En python/sympy

```
from sympy import *
init_printing()
from sympy.polys.domains import ZZ
from sympy.polys.galoistools import gf_mul, gf_add, gf_gcdex,
    gf_rem
def NS(nib):
    polym=ZZ.map([1,0,0,1,1])
    polya=ZZ.map([1,1,0,1])
    polyb=ZZ.map([1,0,0,1])
    polymod=ZZ.map([1,0,0,0,1])
    invnib,t,g=gf_gcdex(nib,polym,2,ZZ)
    return(gf_rem(gf_add(gf_mul(invnib,polya,2,ZZ),polyb,2,ZZ),
           polymod,2,ZZ))
> NS(ZZ.map([1,1]))
[1, 0, 1, 1]
```

Attention, le nibble se lit en image miroir

Autres transformations

- Shift row : transposition bits nibble : $b_0b_1\,b_2b_3 \mapsto b_2b_3\,b_0b_1$.
  $\begin{array}{c|c} 4 & 4 \\ \hline c & f \end{array} \mapsto \begin{array}{c|c} 4 & 4 \\ \hline f & c \end{array}$
- Mix columns : modification de la représ. pol. des colonnes de l'état $\begin{array}{c|c} N_i & . \\ \hline N_j & . \end{array}$; on associe $c(z)=N_iz+N_j \in \mathbb{F}_2[z]/z^2+1$; calculer $c(z).(x^2z+1) \mod z^2+1$.

Exemple

Pour $4f \leftrightarrow 0100\ 1111 \mapsto c(z)=x^2z+x^3+x^2+x+1$ *:*
$(x^3+x^2+1)z+(x^3+x^2)=N_kz+N_\ell \leftrightarrow 1101\ 1100$ *car* $z^2=1, x^4=x+1$ *et* $x^5=x^2+x$.

Mix columns (version matrice)

On travaille directement sur l'état :

$$\begin{pmatrix} 1 & x^2 \\ x^2 & 1 \end{pmatrix} \cdot \begin{pmatrix} S_{0,0} & S_{0,1} \\ S_{1,0} & S_{1,1} \end{pmatrix} =_{\mathbb{F}_{16}} \begin{pmatrix} 1 & 4 \\ 4 & 1 \end{pmatrix} \cdot \begin{pmatrix} S_{0,0} & S_{0,1} \\ S_{1,0} & S_{1,1} \end{pmatrix}$$

Exemple

$$\begin{pmatrix} 1 & x^2 \\ x^2 & 1 \end{pmatrix} \cdot \begin{pmatrix} x^2 & x^2 \\ x^3+x^2+x+1 & x^3+x^2 \end{pmatrix} = \begin{pmatrix} x^3+x^2+1 & 1 \\ x^3+x^2 & x^3+x^2+x+1 \end{pmatrix} = \begin{pmatrix} d & 1 \\ c & f \end{pmatrix}$$

Séquencement de la clé

- initialisation : $w[0] = k_0 \dots k_7 \qquad w[1] = k_8 \dots k_{15}$
- $2 \leq i \leq 5$

$$\begin{cases} w[i] = w[i-2] \oplus \text{RCON}(i/2) \oplus \text{SubNib}(\text{RotNib}(w[i-1])) & i \text{ pair} \\ w[i] = w[i-2] \oplus w[i-1] & i \text{ impair} \end{cases}$$

Avec

- RCON[i]=RC[i]0000
- RC[i] $= x^{i+2} \in \mathbb{F}_{16}$ (RC[1]=$x^3 \leftrightarrow$ 1000)
- RotNib(N_0N_1)=N_1N_0
- SubNib(N_0N_1)=$S(N_0)S(N_1)$ où S est la S-box

Exemple

avec $w[0]w[1] = 0101\ 1001\ 0111\ 1010$, *on a* $w[2] = 1101\ 1100$, $w[3] = 1010\ 0101$, $w[4] = 0110\ 1100$ *et* $w[5] = 1100\ 1010$

Pourquoi SAES ?

- introduit dans [Stallings, 2006] pour des raisons pédagogiques
- version simplifiée de AES utilisable "à la main"
- permet de voir le fonctionnement des cryptanalyses
- illustre tous les principes de fonctionnement de AES

Et le vrai AES ?

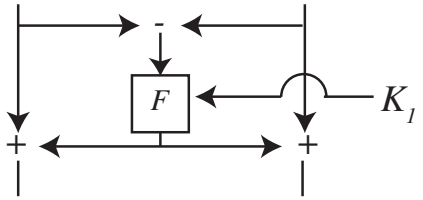
Fonctionne avec r un nombre de tours variable, fonction des tailles de clés et de blocs :

clés\ blocs	128	192	256
128	10	12	14
192	12	12	14
256	14	14	14

On travaille dans GF(256), engendré par le polynôme $x^8 + x^4 + x^3 + x + 1$. Sinon tout fonctionne comme pour SAES.

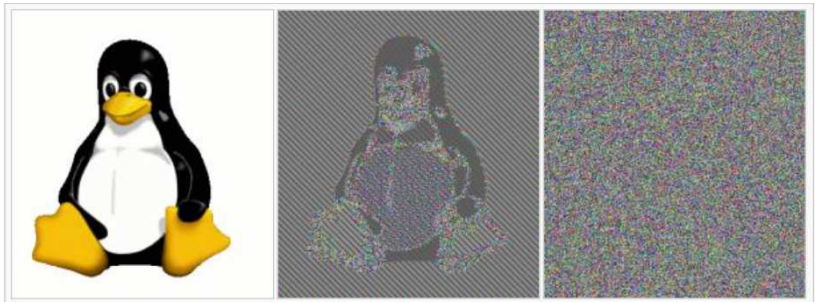
Encore d'autres ?

- dernière catégorie : type Lai-Massey ilustrée par IDEA -un des candidats- au concours AES
- strucure comparable à celle de Feistel
- ajout d'opérations algébriques $-, +$



Modes d'utilisation de ces chiffres

Importance des modes d'utilisation



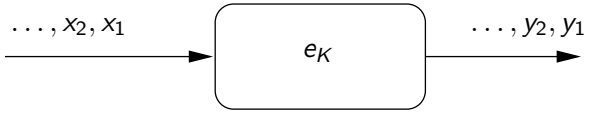
Original image | Encrypted using ECB mode | Modes other than ECB result in pseudo-randomness

The image on the right is how the image might appear encrypted with CBC, CTR or any of the other more secure modes—indistinguishable from random noise. Note that the random appearance of the image on the right does not ensure that the image has been securely encrypted; many kinds of insecure encryption have been developed which would produce output just as "random-looking".

http://en.wikipedia.org/wiki/Block_cipher_mode_of_operation

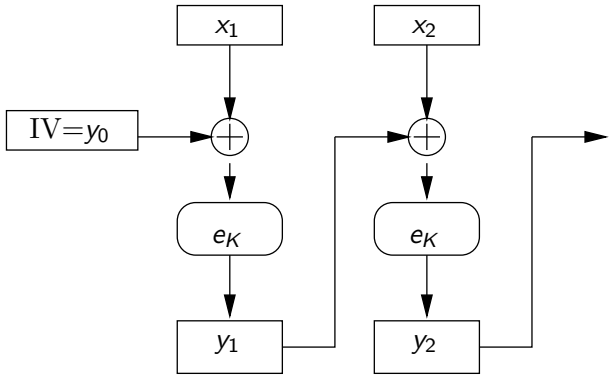
Mode ECB : electronic codebook mode

Celui décrit précédemment, étant donné un clair, chaque bloc x_i est chiffré avec la clé K, donnant le chiffré $y_1 y_2 \ldots$

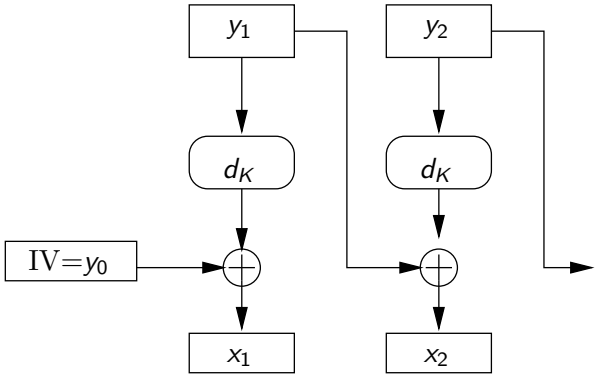


Mode CBC : cipher block chaining mode

Chaque chiffré y_i agit sur le bloc de clair suivant x_{i+1} avant son chiffrement par une opération de ou exclusif.



Mode CBC – Déchiffrement

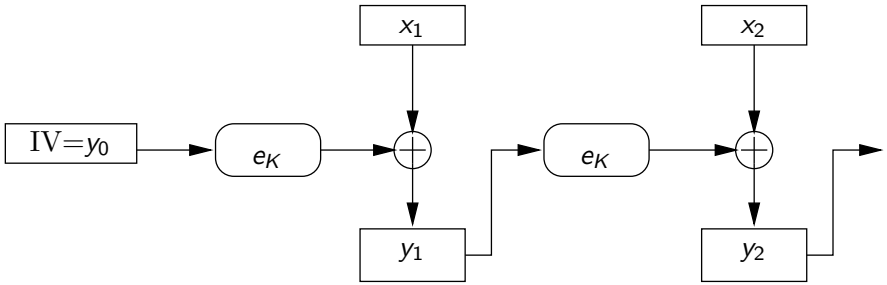


Modes OFB (output feedback mode) et CFB (cipher feedback mode)

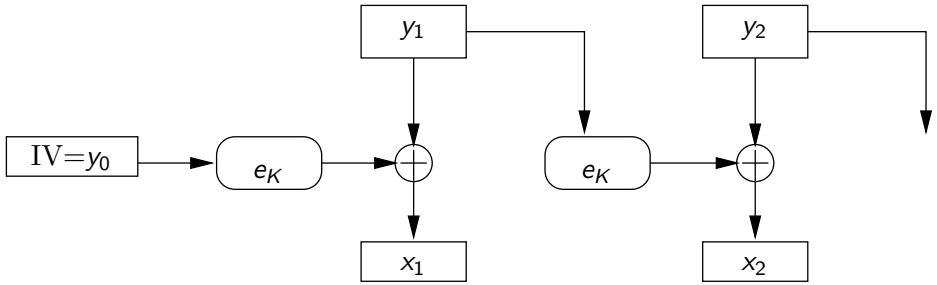
Le chiffrement du clair se fait par une suite de ou exclusifs avec des clés qui sont issues du chiffrement par le DES.

- **OFB** : suite des clés est un chiffrement itéré d'une valeur initiale IV de 64 bits. On définit z_0=IV et on calcule la suite $z_1 z_2 \ldots$ par $z_i = e_K(z_{i-1})$. Le clair est chiffré par $y_i = x_i \oplus z_i$
- **CFB** : on commence avec y_0=IV (bloc de 64 bits) ; la clé suivante est produite en chiffrant le chiffré précédent $z_i = e_K(y_{i-1})$. Sinon, identique au mode OFB

Chiffrement CFB

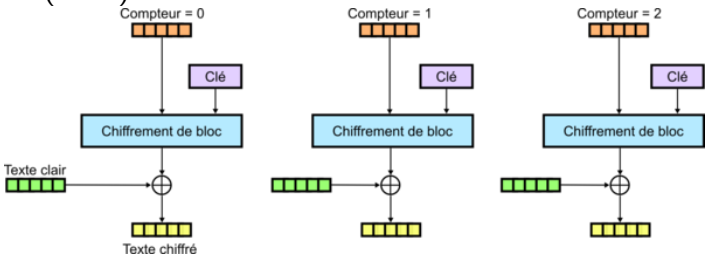


Déchiffrement CFB



Mode CTR

Le flux de clé est obtenu en chiffrant itérativement les valeurs d'un compteur (CTR)



Mode très utilisé car permet un chiffrement par flot et autorise les pré-calculs.

MAC-MDC

Pour Message Authentication Code (Modification Detection Code), également appelé empreinte du message (MAC=MDC+IV$\neq$ 0).
Possible avec CBC et CFB.
On démarre avec IV dont tous les bits sont nuls. On construit ensuite les chiffrés $y_1 \dots y_n$ avec la clé K en mode CBC et le MAC est le bloc y_n. Alice transmet donc le message $x_1 \dots x_n$ avec comme empreinte le MAC y_n.
Quand Bob reçoit $x_1 \dots x_n$, il construit $y_1 \dots y_n$ en utilisant la clé K et vérifie que y_n est identique au MAC reçu.

Beneteau, T. (2010).
Differential cryptanalysis of a very simple block cipher.

Biham, E. and Shamir, A. (1990).
Differential cryptanalysis of DES-like cryptosystems.
In *Crypto'90*.

Heys, H. M. (2002).
A tutorial on linear and differential cryptanalysis.

King, A.
Differential cryptanalysis tutorial.
http://www.theamazingking.com/crypto-diff.php.

Mansoori, S. and Bizaki, H. (2007).
On the vulnerability of simplified aes algorithm against linear cryptanalysis.
International Journal of computer science and network security, 7(7) :257–263.

Matsui, M. (1993).
Linear cryptanalysis method for DES cipher.
In *Eurocrypt'93*.

Musa, M., Schaefer, E., and Wedig, S. (2003).
A simplified aes algorithm and its linear and differential cryptanalyses.
Cryptologia, 17(2) :148–177.

Stallings, W. (2006).
Cryptography and Network Security.
Prentice-Hall, 4th. edition.