TD6 - Cryptanalyse Linéaire

Responsable : M. Bombar

1 Introduction

On s'intéresse au chiffrement par blocs suivant : Il s'agit d'un chiffrement SPN opérant sur des blocs de 32 bits, et sur r tours. La structure des tours est identique, la seule chose qui diffère d'un tour à l'autre est la clé. Une représentation du chiffrement sur 2 tours est donnée en Figure 1. Celles-ci sont obtenues à travers un algorithme de diversification de clé à partir d'une clé maître de 32 bits. On note K_0 la clé initiale et K_i pour $1 \leq i \leq r$ les clés de tours. Les messages, chiffrés et tout état intermédiaires sont représentés par des éléments de $\mathbb{F}_2^{32}$, et les clés sont soit représentées par des entiers entre 1 et $2^{32}-1$, ou bien comme des éléments non nuls de $\mathbb{F}_2^{32}$ donnant leur représentation binaire, avec bit de poids faible à droite, et éventuellement complétée par des 0 à gauche pour former 32 bits.

(Q1) Implémentez une fonction `int_to_vect(x, n)` qui prend en arguments deux entiers x et n où $0 \leq x \leq 2^n - 1$, et renvoie la représentation de x comme un élément de $\mathbb{F}_2^n$ avec bit de poids faible à droite comme décrit ci-dessus. Par exemple, `int_to_vect(4, 4)= (0, 1, 0, 0)`.

(Q2) Implémentez la fonction inverse `vect_to_int(x)` qui prend un argument un vecteur binaire, et renvoie un entier dont la représentation binaire est x.

1.1 Un Chiffrement sur 2 tours

Pour simplifier, on va supposer que le chiffrement ne possède que 2 tours. Initialement, la clé $K_0 \in \mathbb{F}_2^{32}$ est ajoutée au message $m \in \mathbb{F}_2^{32}$ (*i.e.*, xor bit à bit), puis on applique deux fonctions de tours paramétrées par K_1 et K_2. Ces fonctions de tours sont de la forme suivante :

1. **Substitution :** Les blocs de 32 bits sont découpés en 8 blocs de 4 bits. La boîte S est donnée par la table suivante :

x	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
$S(x)$	7	3	6	1	13	9	10	11	2	12	0	4	5	15	8	14

TABLE 1 – Description de la boîte S.

Par exemple, $S((0,1,0,0)) = S[4] \mapsto 13 = (1,1,0,1)$.

2. **Permutation linéaire :** La phase de diffusion est assurée par une simple rotation circulaire de 2 bits vers la droite.
3. **Ajout de clé :** Finalement on ajoute la clé de tour.

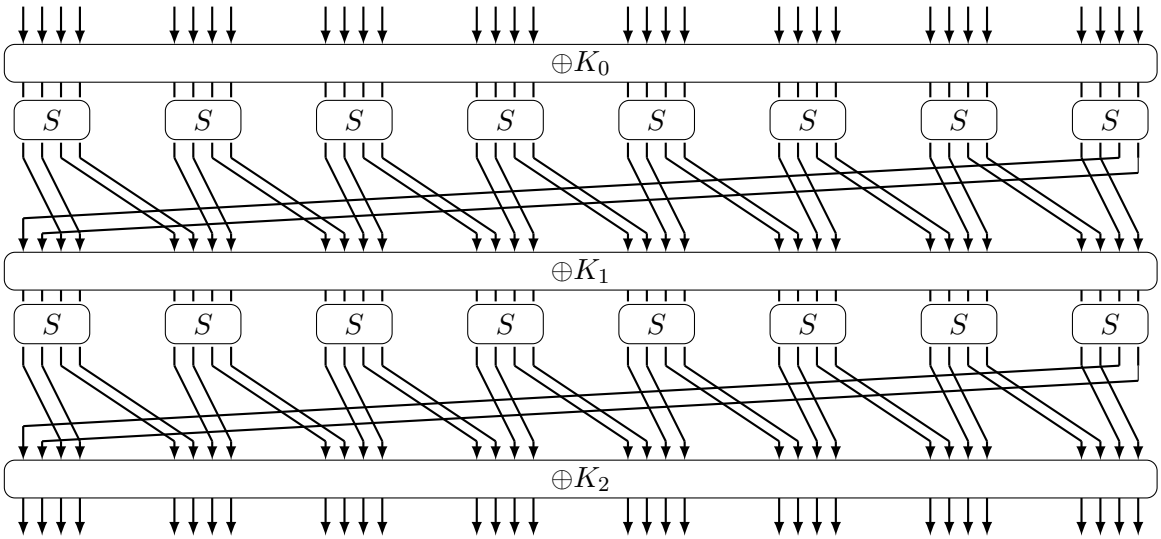

FIGURE 1 – Schéma du chiffrement sur 2 tours

(Q3) Implémentez la fonction de tour de ce chiffrement, ainsi que la fonction de tour inverse.

(Q4) Implémentez le chiffrement et le déchiffrement. Essayez d'être le plus flexible possible sur le nombre de tours, mais si vous préférez, limitez-vous à 2 tours.

Pour tester votre fonction, vous pouvez vérifier que le déchiffrement est bien l'inverse du chiffrement et que le chiffrement de $(0, 0, \ldots, 0, 0)$ avec les clés $K_0 = (1, 0, 0, \ldots, 0, 0, 1)$, $K_1 = (1, 1, \ldots, 1, 1)$ et $K_2 = (0, 1, 1, \ldots, 1, 1, 0)$ est

$$(0, 0, 1, 1, 1, 0, 0, 1, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 1, 0).$$

2 Cryptanalyse Linéaire

On rappelle que la qualité d'une approximation linéaire (α, β) pour une fonction $S : \mathbb{F}_2^n \mapsto \mathbb{F}_2^n$ est mesurée par le nombre de solutions

$$\sigma_{\alpha,\beta} \stackrel{\text{def}}{=} \#\Big\{\mathbf{x} \mid \langle \alpha, \mathbf{x} \rangle + \langle \beta, S(\mathbf{x}) \rangle = 0\Big\}.$$

Plus précisément, la probabilité que l'approximation soit bonne sur une entrée uniformément aléatoire $x \leftarrow \mathbb{F}_2^n$ est donnée par

$$p_{\alpha,\beta} \stackrel{\text{def}}{=} \frac{\sigma_{\alpha,\beta}}{2^n} = \frac{1}{2}\Big(1 + \varepsilon_{\alpha,\beta}\Big).$$

En général, pour monter une attaque linéaire, on représente la qualité de l'approximation par la matrice des approximations linéaires définie par

$$\mathrm{LAT}[\alpha][\beta] = \sigma_{\alpha,\beta} - 2^{n-1}.$$

(Q5) Rappelez la relation entre $\mathrm{LAT}[\alpha][\beta]$ et le biais $\varepsilon_{\alpha,\beta}$.

(Q6) Calculez la matrice des approximations linéaires de S.

(Q7) Déterminez la liste de toutes les approximations linéaires (α, β) qui maximisent (en valeur absolue) le biais $\varepsilon_{\alpha,\beta}$.

Pour monter notre cryptanalyse, on choisit l'approximation donnée par $(\alpha, \beta) = (0100, 1000)$.

Si $x \in \mathbb{F}_2^{32}$, on écrit $x^{(0)}, x^{(1)}, \ldots, x^{(7)}$ les 8 blocs de 4 bits qui décrivent x. Par ailleurs, on définit

$$A = (\alpha, 0, 0, 0, 0, 0, 0, 0) \in \mathbb{F}_2^{32}$$

et

$$B = (\beta, 0, 0, 0, 0, 0, 0, 0) \in \mathbb{F}_2^{32},$$

c'est-à-dire tels que $A^{(0)} = \alpha$ (respectivement $B^{(0)} = \beta$), et $A^{(i)} = 0$ (respectivement $B^{(i)} = 0$) pour $1 \leq i \leq 7$. On note F la fonction de tour du chiffrement, ainsi que P la permutation linéaire interne, et on définit

$$\begin{cases} x_0 = m + K_0 \\ x_1 = F(x_0, K_1) \\ x_2 = F(x_1, K_2) \end{cases}$$

On rappelle qu'une boîte S d'indice i au dernier tour est dite **active** si le bloc $P(B)^{(i)}$ (qui correspond à l'entrée de la boîte $S^{(i)}$, avant l'addition de la clé) n'est pas égal à $(0, 0, 0, 0)$.

(Q8) Vérifiez que (α, β) est bien dans votre liste construite en (Q7), et qu'elle permet de n'avoir qu'une seule boîte S active au deuxième tour.

(Q9) En déduire qu'on pourra retrouver les bits de la clé K_2 par groupe de 4 bits.

(Q10) Montrez que $\langle A, m\rangle = \langle P(B), x_1\rangle$ avec probabilité $\frac{1}{2} + \frac{3}{8}$. Vérifiez expérimentalement cette probabilité en se donnant des clés de tour K_0, K_1, K_2 et en testant avec un grand nombre de m aléatoires.

Générez un triplet de clés (K_0, K_1, K_2), ainsi que deux listes `Plaintext` et `Ciphertext` de longueurs **suffisamment grandes**, telles que `Plaintext[i]` soit un élément uniformément aléatoire dans $\mathbb{F}_2^{32}$ et `Ciphertext[i]` soit le chiffré correspondant.

(Q11) Utilisez l'équation décrite en (Q10) ainsi que vos paires (clair-chiffré) pour retrouver les bits $\{2, 3, 4, 5\}$ de la clé K_2.

(Q12) Adaptez votre attaque pour retrouver tous les bits de K_2.

3 Cryptanalyse Différentielle

Pour vous entraîner, vous pouvez aussi monter une cryptanalyse **différentielle** sur le chiffrement donné dans ce TD.

4 Challenge

On considère le chiffrement à deux tours présenté dans ce TP, où les trois clés de 32 bits K_0, K_1 et K_2 ont été obtenues à partir d'une clé maître K de 32 bits elle-aussi via un algorithme de diversification de clé (*key-schedule*) simple qui consiste à sélectionner pour chaque clé K_i certains bits de K, dont la table est donnée dans la Table 2 ci-dessous. On doit comprendre par exemple que le bit d'indice 0 de K_0 est le bit d'indice 23 de K.

clé	bits de K utilisés
K_0	23, 6, 24, 10, 3, 21, 11, 25, 20, 27, 11, 5, 24, 26, 10, 14, 13, 30, 23, 9, 26, 11, 16, 26, 25, 26, 27, 28, 2, 11, 20, 20
K_1	10, 19, 5, 2, 9, 6, 22, 0, 13, 1, 6, 3, 9, 31, 5, 28, 16, 31, 29, 9, 20, 27, 9, 29, 31, 22, 2, 17, 6, 4, 6, 31
K_2	21, 22, 16, 7, 24, 28, 5, 6, 28, 30, 21, 16, 28, 18, 10, 5, 6, 26, 31, 26, 21, 2, 29, 0, 19, 6, 8, 6, 18, 12, 14, 20

TABLE 2 – Table représentant l'algorithme de diversification de clé

Téléchargez sur cet URL couplesChallenge.sage une liste de couples clairs-chiffrés, et retrouvez la clé maître K que j'ai utilisée pour les générer. Vous pouvez représenter K comme un entier entre 1 et $2^{32} - 1$.