TD 4, Exercice 5 La non-linéarité est nécessaire

UVSQ L3

1 Une idée informelle de ce qu'est la linéarité

Soit f une fonction dont les n entrées sont des bits $a_i, i \in [\![1, n]\!]$ et la sortie est un vecteur de m bits $f(a_1, \cdots, a_n) \in \{0,1\}^m$. On dira que f est linéaire si elle peut s'exprimer uniquement comme somme de certaines entrées. Autrement dit, la seule opération autorisée est le XOR: $\oplus$.

Exemple 1. $f\colon \{0,1\}^2 \to \{0,1\} \quad (x,y) \mapsto x$ *est une fonction linéaire.*
$g\colon (x1, \cdots, x_n) \mapsto (x_1 \oplus x_3 \oplus x_5 \oplus x_7,\ x_2 \oplus x_3,\ x_4 \oplus x_1)$ *est aussi une fonction linéaire.*

En revanche, $f\colon \{0,1\}^2 \to \{0,1\} \quad (x,y) \mapsto xy \oplus x$ *n'est pas linéaire car une multiplication entre deux variables apparait dans son expression.*

Deux observations sont importantes sur les fonctions linéaires:

Proposition 1. *Une somme de fonctions linéaires est une fonction linéaire.*

En effet, informellement, cela revient à considérer une somme de sommes qui est, a fortiori, une somme.

Exemple 2. *Soient* $f : (x,y,z) \mapsto x \oplus y$ *et* $g : (x,y,z) \mapsto z$. *Alors* $f+g : (x,y,z) \mapsto x \oplus y \oplus z$ *et* $f + g$ *est bien linéaire.*

Proposition 2. *Une composition de fonctions linéaires est une fonction linéaire.*

En effet, informellement, cela revient à remplacer les termes d'une somme en des sommes. On obtient finalement encore une somme.

Exemple 3. *si* $f : (a,b) \mapsto a \oplus b$ *et* $g : (x,y,z,t) \mapsto (x \oplus z, y \oplus t)$ *alors:*

$$f \circ g : (x,y,z,t) \mapsto f(g(x,y,z,t)) = f(x \oplus z, y \oplus t) = (x \oplus z) \oplus (y \oplus t) = x \oplus y \oplus z \oplus t.$$

Les preuves formelles de ces deux énoncés ne sont pas beaucoup plus compliquées que les exemples ci-dessus.

Pour finir, voici la définition formelle d'une fonction linéaire. On sera amené à l'utiliser plus bas.

Definition 1 (Définition formelle de la linéarité)**.** Soit $f\colon \{0,1\}^n \to \{0,1\}^m$ une fonction. f est dite linéaire si elle vérifie la propriété suivante:

$$\forall\ X, Y \in \{0,1\}^n \quad f(X \oplus Y) = f(X) \oplus f(Y).$$

Autrement dit, si f est linéaire si "l'image d'une somme est la somme des images".

2 L'énoncé

Considérons un système de chiffrement par bloc qui suit le schéma de Feistel et dont la fonction f utilisée à chaque tour est constituée d'une transformation linéaire A, suivie d'une addition bit à bit avec la clé, puis d'une seconde transformation linéaire B:

$$f(x,k) = B(A(x) \oplus k).$$

Montrer comment il est possible d'attaquer un tel système.

3 Une idée de preuve possible

On se place dans le cadre d'une attaque à clair et chiffré connus: les entrées et les sorties du chiffrement sont connues de l'attaquant. L'attaquant connaît aussi les spécifications du chiffrement (en particulier f).

Puisque B est linéaire, observons, d'après la dernière propriété, que l'on peut réecrire:

$$f(x,k) = B(A(x) \oplus k) = B(A(x)) \oplus B(k).$$

3.1 Schéma de Feistel à un tour

Considérons pour l'instant un seul tour de Feistel.
Observons tout d'abord que R_1 peut se réécrire sous la forme:

$$R_1 = L_0 \oplus f(R_0, K_0) = L_0 \oplus B(A(R_0)) \oplus B(K_0).$$

En additionnant par $L_0 \oplus B(A(R_0))$ de chaque côté de l'équation, on obtient:

$$R_1 \oplus L_0 \oplus B(A(R_0)) = B(K_0).$$

La partie de gauche de l'équation est totalement connue de l'attaquant car celui-ci connaît le clair (L_0, R_0), le chiffré (L_1, R_1), et les fonctions A, B. Il s'agit donc d'une constante lorsqu'on remplace L_0, R_0, L_1, R_1 par leurs valeurs connues.
Dans la partie de droite, on observe en revanche une fonction (connue) qui dépend linéairement de la clé K_0 inconnue.
Ainsi, si l'on regarde cette équation "bit à bit", celle-ci donne lieu à un système linéaire de $n/2$ équations dont les inconnues sont les bits de K_0.

Un adversaire obtient donc un système linéaire dont les inconnues sont des bits de clé. Une résolution de ce système (avec un pivot de Gauss), permet à l'adversaire de retrouver la clé K_0[1].

Remarque 4. On n'utilise ici qu'une des deux égalités du schéma de Feistel. La deuxième, $L_1 = R_0$, est indépendante de la clé et ne peut donc permettre de la retrouver. C'est un cas particulier du schéma à un tour.

[1] Si le nombre d'équations est insuffisant, il suffit à l'adversaire d'utiliser non pas un, mais quelques couples clairs/chiffrés différents.

3.2 Schéma de Feistel à deux tours

Considérons désormais deux tours de Feistel. On a alors

$$L_2 = R_1$$

$$R_2 = L_1 \oplus f(R_1, K_1) = L_1 \oplus B(A(R_1)) \oplus B(K_1).$$

On peut remplacer, dans les expressions de L_2 et R_2, les valeurs de L_1 et R_1 par leurs formes développées pour obtenir:

$$L_2 = L_0 \oplus B(A(R_0)) \oplus B(K_0)$$

$$R_2 = R_0 \oplus B\left(A\left(L_0 \oplus B\left(A(R_0)\right) \oplus B(K_0)\right)\right) \oplus B(K_1).$$

En utilisant la propriété de la définition formelle de linéarité, on obtient:

$$L_2 = L_0 \oplus B(A(R_0)) \oplus B(K_0)$$

$$R_2 = R_0 \oplus B(A(L_0)) \oplus B(A(B(A(R_0)))) \oplus B(A(B(K_0))) \oplus B(K_1).$$

On peut alors, de la même manière, séparer les termes dépendant du clair ou du chiffré, des termes dépendant de la clé.

$$L_2 \oplus L_0 \oplus B(A(R_0)) = B(K_0)$$

$$R_2 \oplus R_0 \oplus B(A(L_0)) \oplus B(A(B(A(R_0)))) = B(A(B(K_0))) \oplus B(K_1).$$

A nouveau, les membres de gauche des équations sont complétement connus d'un attaquant (car ils ne dépendent que des entrées et des sorties de la fonction de chiffrement), et les membres de droites sont des équations linéaires (comme somme et compositions de fonctions linéaires) dépendant uniquement des bits des sous-clés K_0 et K_1.

La résolution d'un tel système (obtenu en regardant les égalités "bit à bit") permet à un attaquant de retrouver les bits de K_0 et K_1 (quitte à utiliser quelques couples clairs/chiffrés pour obtenir plus d'équations).

3.3 Schéma de Feistel à r tours

De manière générale, on peut montrer que, quelque soit le nombre de tours du schéma de Feistel, on peut obtenir, en développant toutes les expressions des variables intermédiaires, deux égalités qui ressemblent à celles obtenues dans les cas de un ou deux tours:

$$g_1(L_0, R_0, L_r, R_r) = h_1(K_0, K_1, \cdots, K_{r-1})$$

$$g_2(L_0, R_0, L_r, R_r) = h_2(K_0, K_1, \cdots, K_{r-1})$$

où g_1, h_1, g_2, h_2 sont des fonctions linéaires connues de l'attaquant.

De nouveau l'attaquant connait L_0, R_0, L_r, R_r et g_1, g_2 donc il connait les parties de gauche. Une fois remplacées les valeurs connues du clair et du chiffré, les parties de gauche des équations sont donc constantes.

A droite, on observe des équations linéaires connues (l'attaquant connait h_1, h_2) dépendant uniquement des bits inconnus des sous-clés $K_0, \cdots, K_{r-1}$.

Ainsi, en regardant les égalités “bit à bit”, l'adversaire peut monter un système linéaire en les bits inconnus des sous-clés $K_0, \cdots, K_{r-1}$.

Même si les équations linéaires semblent plus compliquées, la résolution par pivot de Gauss reste toujours possible. Pour rappel, un système linéaire à s inconnues et s équations peut être résolu par pivot de Gauss en un nombre d'opérations de l'ordre de s^3. Choisir des fonctions linéaires comme fonctions de tour est donc une très mauvaise idée !...