Cryptanalyse des chiffrements à clef secrète par blocs

Anne Canteaut

INRIA - Projet CODES

Anne.Canteaut@inria.fr

http://www-rocq.inria.fr/~canteaut/

1 Le chiffrement à clef secrète par blocs

Parmi toutes les fonctionnalités offertes par la cryptographie, une des principales et des plus anciennes est la protection de la confidentialité de l'information. Pour mettre des données hors de portée des oreilles indiscrètes, il suffit de les rendre incompréhensibles (sauf pour leur destinataire légitime) au moyen d'un algorithme de chiffrement. Chiffrer un message consiste donc à le transformer en un texte chiffré par un procédé qui dépend d'un paramètre appelé la clef de chiffrement. Un interlocuteur privilégié peut alors déchiffrer le message en utilisant la fonction de déchiffrement s'il connaît la clef de déchiffrement correspondant. Un tel système n'est sûr que s'il est impossible à un intrus de déduire le texte clair du message chiffré, et a fortiori de retrouver la clef de déchiffrement.

Il y a maintenant plus d'un siècle, les cryptographes ont pris conscience que la sécurité d'un procédé de chiffrement devait uniquement reposer sur le secret de la clef de déchiffrement utilisée. En effet, il est à la fois irréaliste et dangereux de fonder la sécurité du système sur l'hypothèse qu'un attaquant n'a pas connaissance de la méthode utilisée. La publication récente sur Internet des spécifications d'algorithmes propriétaires, tel celui utilisé dans le système GSM, nous a encore montré qu'il est impossible de conserver un algorithme secret à long terme. Par ailleurs, le fait de rendre publiques les méthodes de chiffrement et de déchiffrement offre une certaine garantie sur la sécurité d'un système, dans la mesure où tout nouvel algorithme cryptographique est immédiatement confronté à la sagacité de la communauté scientifique.

Les *algorithmes de chiffrement à clef secrète* (ou *symétriques* ou encore *conventionnels*) sont ceux pour lesquels émetteur et destinataire partagent une même clef secrète — autrement dit, les clefs de chiffrement et de déchiffrement sont identiques. L'emploi d'un algorithme à clef secrète lors d'une communication nécessite donc l'échange préalable d'un secret entre les deux protagonistes à travers un canal sécurisé ou au moyen d'autres techniques cryptographiques. La découverte en 1976 des systèmes à clef publique a permis de s'affranchir de cette contrainte, mais elle n'a pas pour autant apporté de solution parfaite dans la mesure où tous les algorithmes de chiffrement à clef publique, de par leur lenteur, ne permettent pas le chiffrement en ligne. Les techniques de chiffrement à clef secrète ne sont donc pas tombées en désuétude car elles seules permettent actuellement d'atteindre des débits très élevés.

Seules les techniques de chiffrement à clef secrète dites *par blocs* sont envisagées ici. Un système de chiffrement est dit par blocs s'il divise le texte clair en blocs de taille fixe (généralement 64 ou 128 bits) et chiffre un bloc à la fois avec la même clef. Les exemples les plus connus de chiffrement par blocs sont le DES (Data Encryption Standard) qui fut adopté comme standard américain pour les communications commerciales en 1977 et qui est aujourd'hui vulnérable à la cryptanalyse, et son successeur, l'AES (Advanced Encryption Stan-

dard), choisi au terme d'un concours en octobre 2000 (standard FIPS-197 disponible sur `http://csrc.nist.gov/encryption/aes/`).

2 La recherche exhaustive de la clef

Un paramètre essentiel pour la sécurité d'un système à clef secrète est la taille de l'espace des clefs secrètes. En effet, il est toujours possible de mener sur un algorithme de chiffrement une attaque dite *exhaustive* pour retrouver la clef secrète. Cette attaque consiste simplement à énumérer toutes les clefs possibles du système et à essayer chacune d'entre elles pour décrypter un message chiffré. Si l'espace des clefs correspond à l'ensemble des mots de k bits, le nombre moyen d'appels à la fonction de déchiffrement requis dans une attaque exhaustive est égal à 2^{k-1}. Une telle attaque devient donc hors de portée dès que l'espace des clefs est suffisamment grand. Au vu de la puissance actuelle des ordinateurs, on considère qu'une clef secrète doit comporter au minimum 80 bits. On recommande l'emploi de clefs de 128 bits dès que l'on souhaite une sécurité à relativement long terme. Notons que cette limite évolue avec la technologie. Pour donner un ordre de grandeur, une attaque exhaustive du système de chiffrement DES, qui utilise une clef secrète de 56 bits, a été réalisée en janvier 1998 en 39 jours sur 10 000 Pentium en parallèle, puis en 56 heures en juillet 1998 à l'aide d'une machine dédiée comportant 1500 composants DES (`http://www.eff.org/descracker.html`). Le temps de calcul nécessaire à une attaque exhaustive est évidemment exponentiel en la taille de la clef secrète. Il est 2^{64} fois, c'est-à-dire 18446744073709551616 fois plus dur de casser un système possédant une clef de 128 bits que de casser un système avec une clef de 64 bits (ce qui est déjà très difficile).

3 Les attaques sur le dernier tour

Il existe d'autres types d'attaques sur les systèmes de chiffrement à clef secrète par blocs, qui consistent à exploiter certaines structures particulières de l'algorithme. On considère généralement qu'un chiffrement à clef secrète présente une bonne sécurité s'il n'est pas vulnérable à une attaque qui soit sensiblement plus efficace que la recherche exhaustive de la clef secrète.

La plupart des techniques de cryptanalyse sur les chiffrements par blocs reposent sur le fait qu'il s'agit de chiffrements itératifs. En effet, une idée naturelle et communément employée pour construire un algorithme de chiffrement qui soit à la fois rapide et solide est de répéter un certain nombre de fois une transformation relativement simple. On espère alors que plusieurs itérations de cette fonction la rendront suffisamment inextricable pour assurer la sécurité du système. De façon plus formelle, pour chiffrer un bloc de message, on lui applique une fonction F paramétrée par une quantité secrète k_1 (la sous-clef du premier tour), puis on applique au résultat la même fonction F paramétrée par une autre valeur, k_2 (la sous-clef du deuxième tour)... Après r itérations, on obtient alors le texte chiffré correspondant (cf. Figure 1). Les r sous-clefs $k_1, \ldots, k_r$ sont généralement obtenues à partir de la clef secrète du système K au moyen d'un algorithme de cadencement de clef. Ainsi, le DES comporte 16 itérations d'une même fonction ; les 16 sous-clefs comportent chacune 48 bits et sont dérivées d'une même clef secrète de 56 bits. De même, l'AES utilisant une clef de 128 bits est constitué de 10 tours, chacun d'entre eux étant paramétré par une sous-clef de 128 bits.

Les attaques classiques sur les chiffrements par blocs exploitent donc cette structure itérative. Elles sont appelées *attaques sur le dernier tour* car elles ont pour but de retrouver

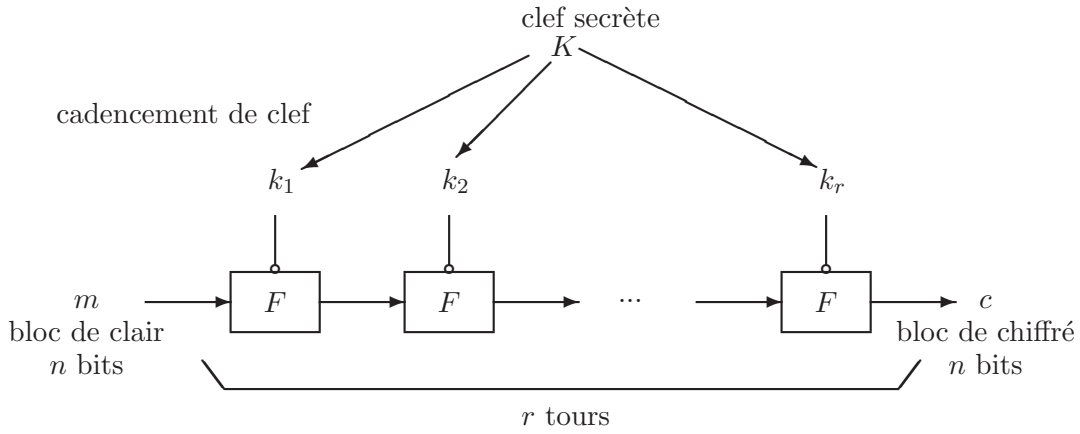

FIG. 1 – *Principe d'un chiffrement itératif*

la valeur de la sous-clef utilisée à la dernière itération. Après avoir mené une attaque sur le dernier tour, l'attaquant peut ensuite essayer de calculer la valeur de la clef secrète K à partir de la dernière sous-clef. Il peut également retrouver successivement les autres sous-clefs k_{r-1}, ..., k_1. En effet, la connaissance de la sous-clef k_r lui permet de supprimer la dernière itération du chiffrement et de se ramener à l'attaque d'un chiffrement à $r-1$ tours. Il est alors possible de retrouver k_{r-1} en appliquant une attaque sur le dernier tour à ce chiffrement réduit (puis les autres sous-clefs en répétant ce procédé).

Les attaques sur le dernier tour sont des attaques à clair connu ou à clair choisi. Cela signifie qu'elles nécessitent la connaissance d'un certain nombre de couples message clair - message chiffré par le système. Dans le cas d'une attaque à clair choisi, il faut que ces couples correspondent en plus à des messages clairs particuliers choisis par l'attaquant. Pour mesurer la complexité d'une attaque de ce type, on prend donc en compte son temps de calcul mais aussi le nombre de couples clairs - chiffrés nécessaires pour la mener à bien.

Toutes les attaques sur le dernier tour sont fondées sur une étude théorique du chiffrement réduit, c'est-à-dire de la fonction de chiffrement amputée de sa dernière itération. Elles reposent sur le principe suivant : il est possible de retrouver la sous-clef utilisée au dernier tour (ou une information sur sa valeur) dès lors que l'on dispose d'un moyen pour distinguer le chiffrement réduit (c'est-à-dire $r-1$ itérations de F) d'une permutation aléatoire. Plus précisément, un détecteur de chiffrement réduit est un moyen permettant, à partir de la donnée d'un certain nombre de couples (x_i, y_i), de déterminer si les valeurs y_i correspondent aux images des x_i par un chiffrement réduit, ou si ces y_i sont des valeurs quelconques (autrement dit, si les y_i sont les images des x_i par une permutation aléatoire). Au moyen d'un tel détecteur, un attaquant peut alors mener une recherche exhaustive sur la sous-clef du dernier tour. Pour cela, il suffit qu'il connaisse des couples clairs - chiffrés (m_i, c_i). Pour chacune des valeurs possibles k, il détermine alors si cette valeur k correspond à la sous-clef k_r utilisée au dernier tour en calculant les quantités $y_i = F_k^{-1}(c_i)$ où F_k^{-1} est l'inverse de la permutation F_k et les c_i sont les chiffrés connus. Ensuite, l'attaquant applique le détecteur aux couples (m_i, y_i) où les m_i sont les textes clairs correspondant aux chiffrés c_i. Si la sous-clef k essayée par l'attaquant correspond à la dernière sous-clef k_r, les y_i sont bien les images des m_i par le chiffrement réduit. En effet, pour $k = k_r$, on a

$$y_i = F_k^{-1} \circ F_{k_r} \circ F_{k_{r-1}} \circ \ldots \circ F_{k_1}(m_i) = F_{k_{r-1}} \circ \ldots \circ F_{k_1}(m_i) \ .$$

Par contre, si la sous-clef k essayée n'est pas correcte, alors les y_i sont les images des m_i par

une fonction qui correspond au chiffrement réduit suivi d'une application de F puis d'une application de F^{-1} :

$$y_i = F_k^{-1} \circ F_{k_r} \circ F_{k_{r-1}} \circ \ldots \circ F_{k_1}(m_i) \ .$$

La sous-clef k n'ayant aucun rapport avec la clef k_r utilisée, les y_i se comportent plus ou moins comme les images des m_i par une permutation aléatoire. On constate donc que les $y_i = F_k^{-1}(c_i)$ sont les images de m_i par le chiffrement réduit si la valeur k essayée est la bonne, et qu'ils sont obtenus par une permutation aléatoire sinon. Tout procédé permettant de distinguer le chiffrement réduit d'une permutation aléatoire permet donc, de cette façon, de déterminer si la sous-clef essayée par l'attaquant correspond à la sous-clef du dernier tour ou non. En résumé, à partir d'un détecteur, l'attaque se déroule de la manière suivante :

Entrée : N couples clairs - chiffrés $(m_1, c_1), \ldots, (m_N, c_N)$.

Sortie : candidats possibles pour la sous-clef du dernier tour k_r.

Algorithme.

Pour toute valeur k possible pour k_r
Pour i de 1 à N, $y_i \leftarrow F_k^{-1}(c_i)$.
Appliquer le détecteur aux couples $(m_1, y_1), \ldots, (m_N, y_N)$.
Si le chiffrement réduit est détecté, alors k est un candidat pour k_r.

Toute attaque sur le dernier tour nécessite donc pour chacune des sous-clefs k essayées, le calcul des N valeurs y_i (c'est-à-dire N évaluations de la fonction F^{-1}) et un appel au détecteur. Le nombre d'opérations nécessaires pour la mener à bien est donc de l'ordre de $2^{n_k}(N + D)$, où n_k est le nombre de bits des sous-clefs et D est le coût d'un appel au détecteur.

On voit donc que le problème essentiel à résoudre pour attaquer de cette manière un algorithme de chiffrement donné est de trouver un détecteur efficace pour le chiffrement réduit. Efficace signifie ici que le détecteur est rapide et également qu'il nécessite uniquement la connaissance d'un petit nombre de couples entrées - sorties. Les attaques classiques (cryptanalyse différentielle, cryptanalyse linéaire...) diffèrent donc par le type de détecteur utilisé. Chacun de ces détecteurs tente d'exploiter une faiblesse particulière du chiffrement réduit.

4 La cryptanalyse différentielle

La toute première méthode d'attaque sur le dernier tour, la cryptanalyse différentielle, a été publiée en 1991 par Biham et Shamir. Il s'agit d'une attaque à clair choisi qui nécessite la connaissance des chiffrés correspondant à des couples de messages clairs dont la différence est fixée. Elle peut être mise en œuvre dès lors que le chiffrement réduit présente la faiblesse suivante : il existe un couple de différences, (a, b), tel que la différence entre les images par le chiffrement réduit de deux entrées dont la différence vaut a est égale à b avec une probabilité élevée. La différence entre deux blocs de n bits, ici notée $\oplus$, est généralement le ou exclusif bit à bit (xor). Autrement dit, l'attaque nécessite que, pour toutes les valeurs possibles de $k_1, \ldots, k_{r-1}$, la fonction de chiffrement réduit $G = F_{k_{r-1}} \circ \ldots \circ F_{k_1}$ vérifie $G(x \oplus a) + G(x) = b$ pour une grande proportion des valeurs de x. On peut alors détecter le chiffrement réduit à partir de la connaissance des valeurs prises par la fonction pour des entrées dont la différence vaut a. Le détecteur associé considère donc des couples d'entrées - sorties de la forme $(x_1, y_1), (x_1 \oplus a, y_1'), (x_2, y_2), (x_2 \oplus a, y_2'), \ldots$ et il compte le nombre de couples (y_i, y_i') qui

vérifient $y_i \oplus y'_i = b$. Si les y_i et les y'_i ont été obtenus en appliquant le chiffrement réduit respectivement aux x_i et aux $x_i \oplus a$, ce nombre est élevé. Sinon, la proportion de (y_i, y'_i) qui diffèrent de b est proche de $1/2^n$ (puisque, dans ce cas, $y_i \oplus y'_i$ peut prendre n'importe quelle valeur de n bits avec la même probabilité). Pour que ce détecteur puisse fonctionner, il faut que le nombre N de valeurs (x_i, y_i) et $(x_i \oplus a, y'_i)$ connues soit suffisamment grand. Il doit être supérieur à l'inverse de $(p - \frac{1}{2^n})$ où p est la probabilité que la différence des images de deux éléments qui diffèrent de a soit égale à b.

Toute la difficulté pour concevoir une attaque différentielle réside évidemment dans la recherche d'un couple de différences (a, b) qui soit propagé par le chiffrement réduit avec une probabilité élevée. Cette recherche nécessite une étude très fine de la fonction F itérée par le chiffrement.

Exemple : attaque différentielle du DES à 4 tours

Voyons par exemple comment on peut mener une attaque différentielle sur le DES à 4 tours — il s'agit d'un exemple illustratif puisque le DES comporte 16 tours. La fonction qui est itérée dans le DES est représentée à la figure 2. Elle consiste à séparer l'entrée composée de

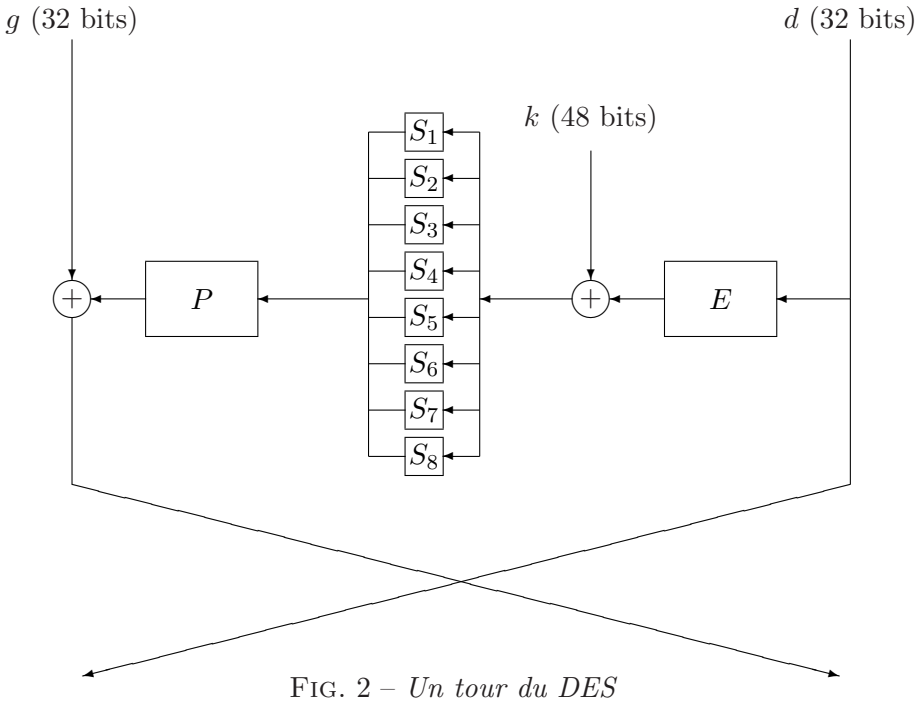

FIG. 2 – *Un tour du DES*

64 bits en deux mots de 32 bits, g et d, qui correspondent respectivement aux 32 bits de gauche et de droite de l'entrée. Ensuite, on transforme d en un mot de 48 bits par une fonction E qui consiste à dupliquer certains bits de d. Puis, on additionne par un xor la sortie de E à la sous-clef du tour. On découpe le résultat en 8 mots de 6 bits, $b_1, \ldots, b_8$. Chaque b_i rentre dans une boîte S_i qui le transforme en un mot de 4 bits. Les 8 mots de 4 bits obtenus en sortie des boîtes S sont regroupés en un seul mot de 32 bits dont les bits sont ensuite permutés suivant une permutation P. Le mot obtenu est finalement additionné (xor) avec la moitié gauche g de l'entrée, ce qui produit la moitié droite de la sortie de la fonction. La moitié gauche de la sortie

est, elle, simplement égale à la moitié droite de l'entrée, d. La description complète des fonctions E, P et des boîtes-S est publique et disponible par exemple dans le chapitre 7 du *Handbook of Applied Cryptography* (`http://cacr.math.uwaterloo.ca/hac/about/chap7.pdf`). Dans le DES, le premier tour est précédé d'une permutation des bits du message clair, et le dernier tour est suivi de la permutation inverse. Nous ferons abstraction de ces deux transformations ici dans la mesure où elles n'ont aucune influence sur la cryptanalyse.

Dans le but de trouver une différence a qui se propage à travers plusieurs tours du DES, nous commençons par nous intéresser à un seul tour. Dans toute la suite, les bits d'un mot sont numérotés de 1 à n à partir de la gauche. Considérons les images de deux entrées (g, d) et (g', d') dont les 32 bits de gauche diffèrent d'une constante α_g et dont les 32 bits de droite diffèrent de la constante hexadécimale 60000000. Autrement dit, tous les bits de d et d' sont égaux sauf les bits en deuxième et troisième positions à partir de la gauche. La figure 3 montre la propagation de la différence $d \oplus d' =$ 60000000 à travers un tour du DES. La transformation

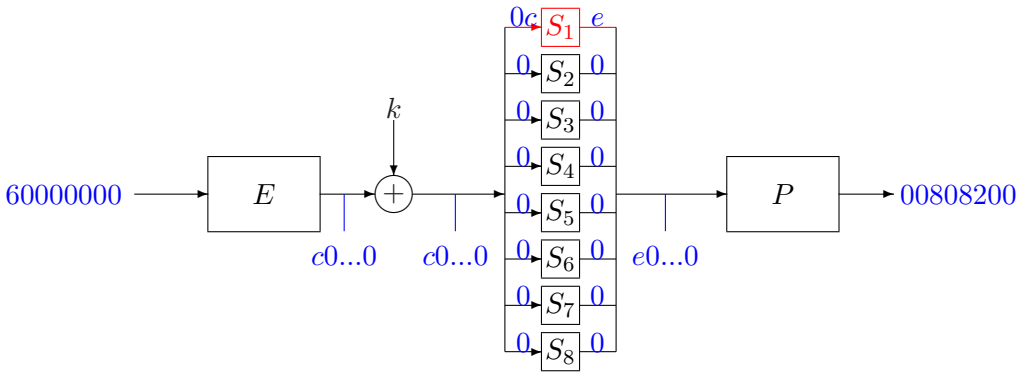

FIG. 3 – *Propagation sur un tour de la différence* 60000000 *entre les moitiés droites des entrées*

E envoie les bits 2 et 3 de son entrée sur les bits 3 et 4 de sa sortie. On en déduit que, si les entrées de E diffèrent uniquement sur les bits 2 et 3, alors les sorties ne diffèrent que sur les bits 3 et 4. Le fait d'ajouter la sous-clef k ne change pas la valeur de la différence. Ainsi, à l'entrée des boîtes S, les deux mots que l'on considère ne diffèrent que sur leurs 3^{e} et 4^{e} bits. Ces deux bits n'interviennent qu'en entrée de la boîte S_1. Les entrées des 7 autres boîtes S sont les mêmes pour les deux messages. Donc, leurs sorties seront les mêmes ; autrement dit, la différence des sorties des boîtes S est un mot de 32 bits dont les octets 2 à 8 sont nuls. Intéressons-nous maintenant à la boîte S_1. Nous avons deux mots de 6 bits, x et x', en entrée de cette boîte qui diffèrent en position 3 et 4 ($x' = x \oplus 001100$ en binaire) et nous voulons avoir des informations sur la valeur de $S_1(x) \oplus S_1(x')$. Une étude détaillée de S_1 permet de constater que, sur les 64 valeurs possibles pour x (tous les mots de 6 bits), 14 d'entre elles vérifient

$$S_1(x) \oplus S_1(x \oplus 001100) = 1110 \ .$$

Les deux octets qui vont sortir de S_1 auront donc une différence égale à la constante hexadécimale e avec une probabilité 7/32. On en déduit donc que les deux mots de 32 bits auxquels on va appliquer la permutation P diffèrent de e sur leur premier octet, et ont leurs 7 autres octets égaux ; ces deux mots diffèrent donc uniquement sur leurs positions 1, 2 et 3. La permutation P envoie les bits 1, 2 et 3 en positions 9, 17 et 23. Donc, en sortie de P, la différence entre nos deux messages va être égale à la constante de 32 bits dont tous les bits sauf les 9^{e}, 17^{e} et

23^{e} sont nuls, c'est-à-dire en hexadécimal à 00808200. Maintenant, on ajoute à ces mots de 32 bits, x et $x' = x \oplus 00808200$, la partie gauche des entrées de la fonction, g et g'. Comme $g' = g \oplus \alpha_g$, on en déduit

$$(x \oplus g) \oplus (x' \oplus g') = (x \oplus x') \oplus (g \oplus g') = 00808200 \oplus \alpha_g \ .$$

Il s'ensuit que les parties droites des sorties de la fonction diffèrent de $00808200 \oplus \alpha_g$ avec une probabilité 7/32. La différence entre les parties gauches des sorties est égale à la différence entre les parties droites des entrées, c'est-à-dire à 60000000. La conclusion de cette étude est donc que, pour deux entrées (g, d) et (g', d') de la fonction telles que $g \oplus g' = \alpha_g$ et $d \oplus d' = 60000000$, la différence entre les sorties va être égale à 6000000 sur la moitié gauche et à $00808200 \oplus \alpha_g$ sur la moitié droite, avec une probabilité 7/32.

Grâce à cette étude, il est maintenant possible de trouver un couple de différences qui se propage sur 3 tours du DES avec une probabilité élevée (cf. Figure 4). En effet, supposons

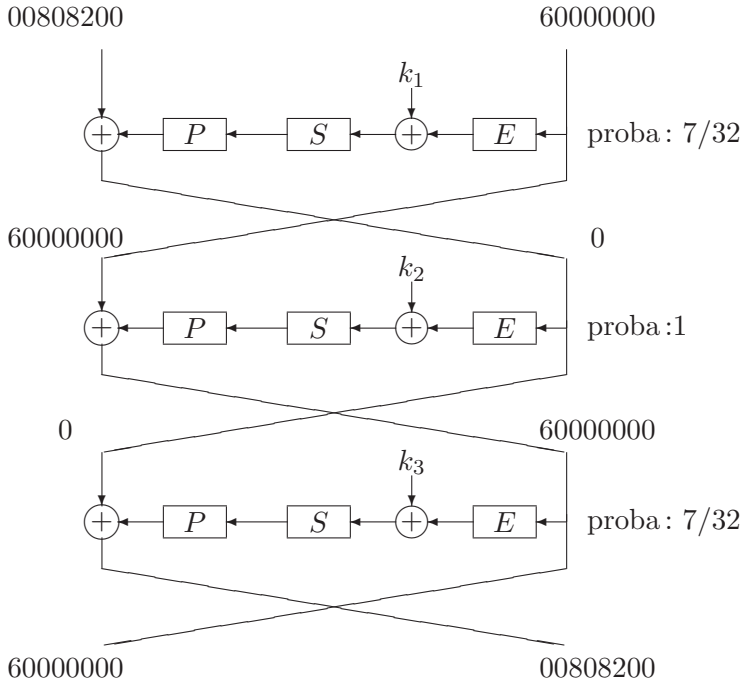

FIG. 4 – *Propagation de la différence* $(00808200, 60000000)$ *sur trois tours*

que l'on applique 3 tours du DES à deux messages qui diffèrent de 00808200 sur leurs 32 bits de gauche et de 60000000 sur leurs 32 bits de droite. En utilisant le résultat précédent avec $\alpha_g = 00808200$, on obtient que les sorties du premier tour vont différer de 60000000 sur la partie gauche et de 0 sur la partie droite, avec une probabilité 7/32. Les entrées du deuxième tour coïncident donc sur leurs moitiés droites. Il s'ensuit que les sorties de la fonction P au deuxième tour sont les mêmes (puisqu'on a appliqué la même transformation à deux valeurs égales). Donc, la différence entre les moitiés droites des sorties du deuxième tour est égale à la différence entre les moitiés gauches des entrées du deuxième tour, c'est-à-dire à 60000000,

avec une probabilité 1. La différence des moitiés gauches des sorties du deuxième tour est évidemment égale à la différence des entrées droites, c'est-à-dire à 0. En entrée du troisième tour, nous avons donc deux messages dont la différence vaut 0 sur la partie gauche et 60000000 sur la partie droite. En utilisant le résultat de la figure 3 avec $\alpha_g = 0$, on trouve donc que les sorties du troisième tour diffèrent de 60000000 sur leurs moitiés gauche et de 00808200 sur leurs moitiés droites. La probabilité que l'on obtienne cette différence en sortie du troisième tour si l'on part en entrée de la différence $(00808200, 60000000)$ est donnée par le produit des probabilités obtenues à chaque tour :

$$\frac{7}{32} \times 1 \times \frac{7}{32} \simeq 0,048 \ .$$

Par ce raisonnement, nous avons construit un détecteur du DES à 3 tours : ce détecteur prend en entrée 100 couples d'entrées - sorties d'une fonction de la forme $(x_1, y_1), (x_1 \oplus a, y'_1), \ldots, (x_{50}, y_{50}), (x_{50} \oplus a, y'_{50})$ où a est la constante hexadécimale $(00808200, 60000000)$. Il compte combien, parmi les 50 couples (y_i, y'_i), sont tels que $y_i \oplus y'_i = (60000000, 00808200)$. Si ce nombre est proche de $0,048 \times 50 \simeq 2,4$, alors les valeurs présentées au détecteur sont bien des couples d'entrées - sorties d'un DES à 3 tours. Par contre, si les couples présentés au détecteur sont obtenus à partir d'une permutation aléatoire, il n'y a généralement aucun couple (y_i, y'_i) dont la différence soit égale à $(60000000, 00808200)$: cela n'arrive qu'avec une probabilité $(1/2^{64}) \times 50 < 10^{-17}$. Ce détecteur permet donc de bâtir une attaque sur le DES à 4 tours qui permet de retrouver la clef utilisée au dernier tour à partir de la connaissance de 100 couples clairs - chiffrés du système.

Pour attaquer le DES complet, il faut trouver une différence qui se propage sur 15 tours du DES avec une probabilité élevée, ce qui est malheureusement beaucoup plus difficile. La meilleure attaque différentielle connue sur le DES complet nécessite la connaissance de 2^{47} couples clairs - chiffrés.

5 La cryptanalyse linéaire

Une seconde catégorie d'attaques sur le dernier tour, la cryptanalyse linéaire, a été proposée par Matsui en 1993. Il s'agit d'une attaque à clair connu qui peut être menée dès que le chiffrement réduit possède la faiblesse suivante : il existe un ensemble de positions $i_1, i_2, \ldots, i_u$ du mot entré et un ensemble de positions $j_1, j_2, \ldots, j_v$ de la sortie tels que la somme (xor) des bits $i_1, i_2, \ldots$ de l'entrée plus la somme des bits $j_1, j_2, \ldots$ de la sortie prend la même valeur pour la plupart des entrées. Autrement dit, si on note $x[i]$ le i^{e} bit de x, cela signifie que, pour toutes les valeurs possibles de $k_1, \ldots, k_{r-1}$, la fonction de chiffrement réduit $G = F_{k_{r-1}} \circ \ldots \circ F_{k_1}$ vérifie

$$x[i_1] \oplus x[i_2] \oplus \ldots \oplus x[i_u] \oplus G(x)[j_1] \oplus G(x)[j_2] \oplus \ldots \oplus G(x)[j_v] = \varepsilon$$

pour la plupart des valeurs de x, où ε est une constante binaire indépendante de x mais qui peut dépendre des $k_1, \ldots, k_{r-1}$. On peut alors exploiter cette propriété pour construire un détecteur. À partir de la connaissance de N couples d'entrées - sorties x, y, on compte le nombre de ces couples qui vérifient l'équation linéaire

$$x[i_1] \oplus \ldots \oplus x[i_u] \oplus y[j_1] \oplus \ldots \oplus y[j_v] = 0 \ .$$

Si les y ont été obtenus en appliquant le chiffrement réduit aux messages x, alors l'expression linéaire prend généralement la même valeur pour tous les x. Si cette valeur est 0, alors le

nombre de couples (x, y) qui vérifie la propriété est très grand ; si c'est 1, ce nombre est très petit. Par contre, si les y sont obtenus à partir d'une permutation aléatoire, alors l'expression linéaire va prendre la valeur 0 avec une probabilité proche de $1/2$, c'est-à-dire pour à peu près la moitié des couples (x, y).

Pour que le détecteur fonctionne avec une probabilité de succès très élevée, il faut que le nombre de couples (x, y) connus soit de l'ordre de l'inverse de $(p - \frac{1}{2})^2$, où p est la probabilité que l'équation linéaire soit satisfaite.

Exemple : attaque linéaire du DES à 4 tours

Pour mener à bien la cryptanalyse linéaire du DES à 4 tours, il faut trouver une équation linéaire du DES à 3 tours qui soit satisfaite avec une probabilité élevée. Recherchons tout d'abord une telle équation pour un tour du DES, le tour numéro i. Soit d_i la partie droite de l'entrée du i^{e} tour. Le bit 26 de la sortie de la fonction E correspond au bit 17 de d_i. Si x est

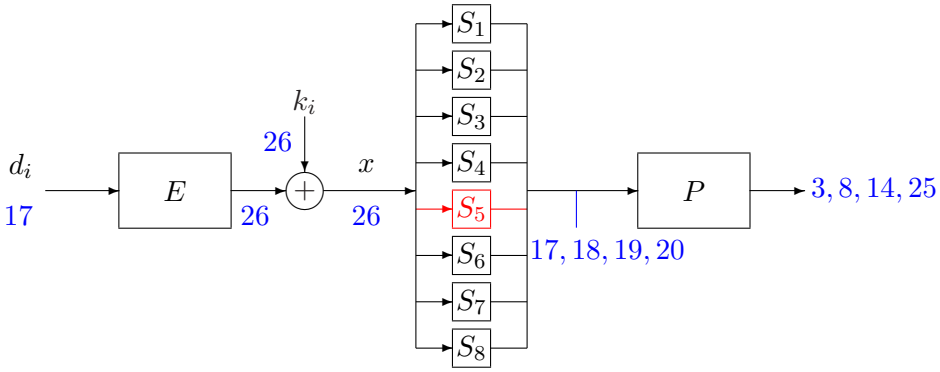

FIG. 5 – *Approximation linéaire du i^{e} tour du DES*

le mot de 48 bits en entrée des boîtes S, on a donc que le bit 26 de x correspond au bit 17 de d_i additionné au bit 26 de la sous-clef k_i. Le bit 26 de x intervient en entrée de la boîte S_5 (il s'agit du deuxième bit de l'entrée de S_5). Or, une étude précise de S_5 montre que la somme du deuxième bit de son entrée et de tous les bits de sa sortie vaut 0 pour 12 des 64 entrées possibles (c'est-à-dire avec une probabilité 3/16). Les bits de sortie de S_5 sont les bits 17 à 20 du mot de 32 bits $S(x)$ obtenus en regroupant les sorties des 8 boîtes S. On a donc

$$S(x)[17] \oplus S(x)[18] \oplus S(x)[19] \oplus S(x)[20] = x[26] = k_i[26] \oplus d_i[17]$$

avec une probabilité 3/16. La permutation P transforme les bits 17, 18, 19 et 20 de $S(X)$ en 3, 8, 14 et 25. Or, la sortie de P est égale à $d_{i+1} \oplus g_i$, où g_i est la moitié gauche de l'entrée du tour et d_{i+1} la moitié droite de la sortie (qui est l'entrée du tour suivant). On en déduit donc l'équation linéaire suivante liant les bits de l'entrée (g_i, d_i) du tour et ceux de la partie droite d_{i+1} de sa sortie :

$$g_i[3] \oplus g_i[8] \oplus g_i[14] \oplus g_i[25] \oplus d_{i+1}[3] \oplus d_{i+1}[8] \oplus d_{i+1}[14] \oplus d_{i+1}[25] \oplus d_i[17] \oplus k_i[26] = 0$$

avec une probabilité 3/16.

Écrivons cette expression pour les premier et troisième tours ($i = 1$ et $i = 3$). On a avec une probabilité 3/16

$$g_1[3] \oplus g_1[8] \oplus g_1[14] \oplus g_1[25] \oplus d_2[3] \oplus d_2[8] \oplus d_2[14] \oplus d_2[25] \oplus d_1[17] \oplus k_1[26] = 0 \ ,$$

$$g_3[3] \oplus g_3[8] \oplus g_3[14] \oplus g_3[25] \oplus d_4[3] \oplus d_4[8] \oplus d_4[14] \oplus d_4[25] \oplus d_3[17] \oplus k_3[26] = 0 \ .$$

Or, la partie droite de l'entrée du deuxième tour, d_2 est exactement égale à la partie gauche de sa sortie g_3. De même la partie droite de l'entrée du troisième tour d_3 est égale à la partie gauche de sa sortie, g_4. En effectuant la somme des deux expressions de gauche, on obtient

$$g_1[3] \oplus g_1[8] \oplus g_1[14] \oplus g_1[25] \oplus d_1[17] \oplus d_4[3] \oplus d_4[8] \oplus d_4[14] \oplus d_4[25] \oplus g_4[17] \oplus k_1[26] \oplus k_3[26] = 0 \ .$$

Comme il s'agit d'une expression binaire, cette somme vaut 0 si les deux termes sont égaux à 0 ou si les deux sont égaux à 1. C'est donc vrai avec une probabilité $(3/16)^2 + (1 - 3/16)^2 \simeq 0,7$. Nous avons donc obtenus une équation linéaire liant certains bits de l'entrée (g_1, d_1) et certains bits de sa sortie (g_4, d_4) qui prend une valeur constante (égale à $k_1[26] \oplus k_3[26]$) avec une probabilité relativement élevée. Nous venons ainsi de construire un détecteur du DES à 3 tours : ce détecteur prend en entrée 26 couples d'entrées - sorties d'une fonction, $(x_1, y_1), \ldots, (x_{26}, y_{26})$ et il compte le nombre de couples (x, y) parmi ces 26 qui vérifient

$$x[3] \oplus x[8] \oplus x[14] \oplus x[25] \oplus x[49] \oplus y[35] \oplus y[40] \oplus y[46] \oplus y[57] \oplus y[17] = 0 \ ,$$

puisque le bit i de d_1 correspond au bit $(i + 32)$ de x. Si ce nombre est proche de 18 (dans le cas où $k_1[26] \oplus k_3[26] = 0$) ou s'il est proche de 8 (cas où $k_1[26] \oplus k_3[26] = 1$), alors les valeurs présentées au détecteur sont bien des couples d'entrées - sorties d'un DES à 3 tours. Par contre, si ces couples sont obtenus à partir d'une permutation aléatoire, le nombre de couples (x, y) vérifiant l'équation sera proche de la moitié, c'est-à-dire de 13.

Pour attaquer le DES complet sur 16 tours, il faut de la même façon trouver une équation linéaire qui lie les entrées et les sorties de 15 tours du DES et qui soit satisfaite avec une probabilité élevée. La meilleure approximation connue est satisfaite avec une probabilité de l'ordre $1/2 + 2^{-22}$. Elle conduit à une attaque nécessitant la connaissance de 2^{43} couples clairs-chiffrés.

6 Conclusion

La cryptanalyse différentielle et la cryptanalyse linéaire sont des techniques maintenant bien connues et on sait aujourd'hui comment choisir la fonction itérée pour que le chiffrement résiste à ces deux attaques. Le nouveau standard de chiffrement par blocs, l'AES, a été conçu de cette manière. Mais cela ne garantit pas pour autant la sécurité du système. Il existe en effet d'autres attaques sur le dernier tour, plus récentes, (cryptanalyse différentielle d'ordre supérieur, Square attack...) dont les détecteurs exploitent d'autres types de faiblesses du chiffrement réduit. Avant d'utiliser un chiffrement à clef secrète, il convient donc de s'assurer qu'il résiste à toutes les attaques connues. Une liste des principaux chiffrements par blocs avec la complexité des meilleures attaques connues est maintenue par L. Knudsen et V. Rijmen sur `http://www.ii.uib.no/~larsr/bc.html`. Enfin, il faut avoir conscience que la sécurité d'un système de chiffrement n'est jamais garantie car elle évolue dans le temps et que l'on ne peut pas exclure l'apparition de nouvelles attaques efficaces.