Alexandre UNG

Quentin RAND

Pierre N'GUESSAN

Cryptanalyse Différentielle

I) Introduction

La cryptanalyse différentielle a été inventé par Eli Biham et Adi Shamir en 1991. Il s'agit d'une attaque a texte clair choisi appliquée aux algorithmes de chiffrement itératifs par blocs ainsi que les algorithmes de chiffrement par flots et aux fonctions de hachage.

Elle consiste à étudier les données en sortie en fonction des modifications apportées en entrées, de plus, elle a été initialement conçue pour casser les algorithmes de chiffrement itératif par blocs comme le DES (Data Encryption Standard) ou encore le FEAL (Fast Data Enciphement Algorithm).

Il s'avère que le FEAL a été vraiment faible contre cette méthode (8 messages clairs choisis nécessaires pour briser FEAL-4 qui était d'époque) alors que le DES a été très résistant à cette méthode. Cette résistance du DES entraina la rumeur selon laquelle, cet algorithme aurait été créé de façon à résister à la T-Attack comme elle était surnommée en 1970 (Tickling Attack) puisqu'elle consiste à "*chatouiller*" l'entrée et d'observer le résultat en sorti.

Nous nous intéresserons seulement à la cryptanalyse différentielle d'origine et nous verrons notamment son efficacité contre l'algorithme de FEAL.

II) Principe de l'analyse différentielle

Le principe général de cette attaque consiste à considérer des couples de clairs X et X' présentant une différence ΔX fixée et à étudier la propagation de cette différence initiale à travers le chiffrement. On traite les couples d'entrée et de sortie comme des variables aléatoires que l'on note X, Y, ΔX, ΔY. Les différences sont définies par le XOR bit à bit.

L'attaque exploite la probabilité d'apparition d'occurrences de différences entre des clairs et d'occurrences de différences entre des chiffrés en entrée du dernier tour du chiffre.

Entrées X = [X1 . . .Xn] sorties Y = [Y1 . . .Yn] ; on étudie ΔX = X‘ ⊕ X’’ (2 entrées) et ΔY = Y‘ ⊕ Y’’ (2 sorties).

Si le système cryptographique était parfait alors la probabilité pour qu'un ΔY provienne d'un ΔX devrait être de 1/2^n où n est le nombre de bits de X. Ce qui n'est pas le cas (pseudo-aléatoire)

La cryptanalyse exploite les apparitions d'un ΔY particulier pour un certain ΔX avec forte une probabilité p.

C'est une attaque à clair choisi : on choisit des paires d'entrées X', X'' = ΔX tel que le ΔX considéré mène avec une forte probabilité à un ΔY particulier.

Comment faire l'attaque :

- Trouver les (ΔX, ΔY) les plus probables. On appelle le couple (ΔX, ΔY) différentielle
- Examiner les propriétés des boîtes-S et construire les caractéristiques différentielles
- Considérer les ΔX et ΔY des boîtes-S pour trouver les plus fréquentes
- Combiner l'information sur les boîtes-S pour construire une approximation globale du chiffre

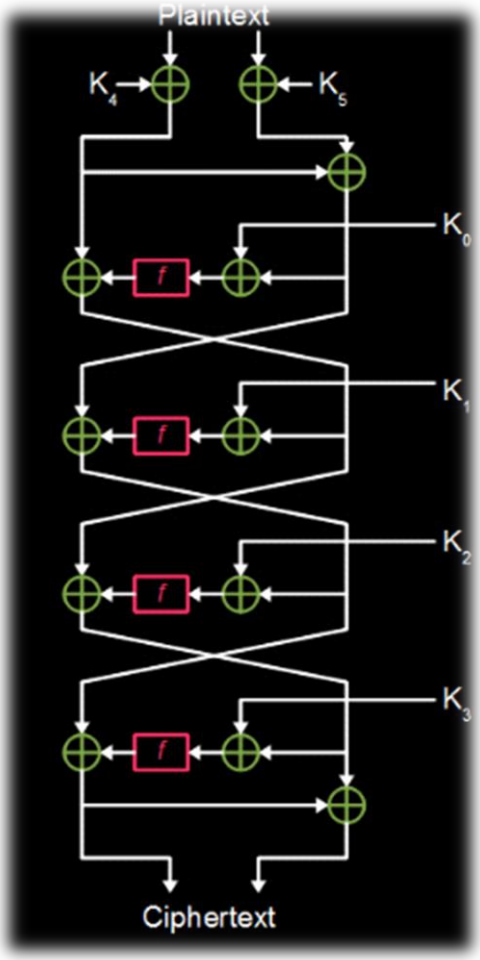
III) Principe de FEAL-4

FEAL-4 est un chiffrement de Feistel en 4 tours avec des blocs de 64 bits. Cela signifie donc que l'algorithme crypte et décrypte des données dans des blocs de 64 bits. La structure de Feistel signifie que les blocs sont en réalité divisés en deux moitiés égales. Ces moitiés sont ensuite mélangées entre elles avec des opérations XOR tout au long du chiffrage. Il y a au cours de cet algorithme l'utilisation d'un composant non linéaire : la fonction ronde.

Cette fonction à sens unique va prendre un bloc de 32 bits en entré et retourner un bloc de 32 bits en sortie. Elle sera utilisée 4 fois dans notre algorithme (une fois pour chaque tour, FEAL-4).

La clé de 64 bits est étendue en six sous-clés de 32 bits. Dans le cas de FEAL, le processus est à sens unique ainsi, il devrait être impossible d'utiliser une des sous clés pour retrouver la "vraie" clé. Autrement dit, pour l'explication, nous allons considérer que FEAL-4 a une clé de 192 bits.

Nous allons maintenant passer à l'explication de l'algorithme :



Le commencement : d'abord il y a un XOR de la moitié gauche et de la moitié droite du texte clair avec deux sous clefs (K4 et K5). Puis il y a un XOR entre les deux nouvelles moitiés. La moitié de gauche est en attente et la partie de droite va être utilisée directement. Ensuite, nous allons expliquer les tours (4).

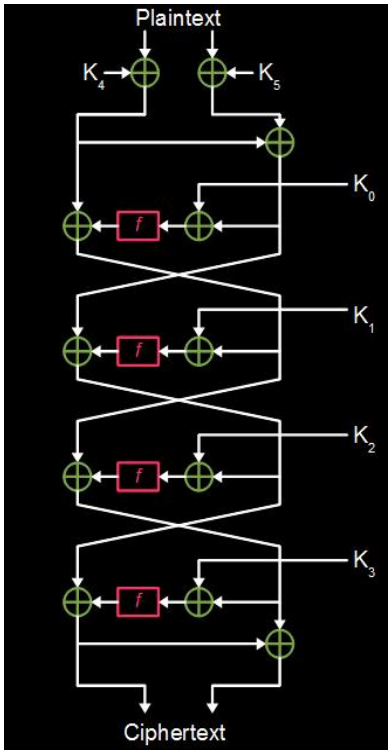
Les tours : La moitié de droite originelle est XOR avec une sous clé (K0 pour le premier tour, jusqu'à K3 pour le tour 4). Le produit de ce XOR servira d'apport à la fonction ronde. La sortie de la fonction ronde sera XOR avec la moitié de gauche qui était en état d'attente depuis lors. Pour les tours suivants, la sortie sera utilisée en entrée et la nouvelle sortie après traitement sera XOR avec l'entrée du tour précédent. Il y a donc un échange entrée/sortie au fil des tours.

Dernier tour : Le dernier tour est un peu différent puisqu'il n'y a pas d'échange entrée/sortie. La sortie du dernier tour sera la moitié gauche du texte chiffré final et la partie droite sera le XOR entre l'entrée et la sortie du dernier tour.

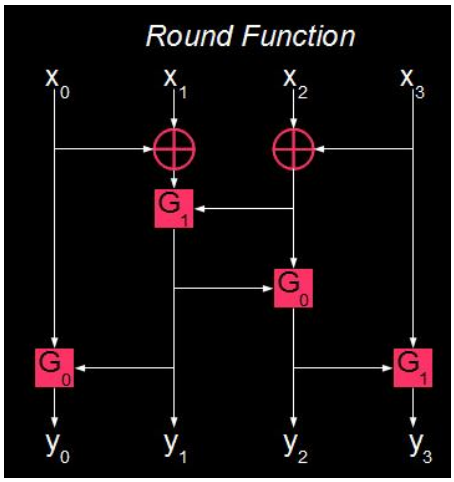
La fonction ronde qui a en entrée et en sortie 32 bits est la seule partie non linéaire du processus. A l'intérieure de celle-ci, les 32 bits sont divisés en 4 morceaux de 8 bits qui vont se mixer entre eux via des opérateurs. C'est donc cette fonction qui avait pour but de résister à la cryptanalyse différentielle.

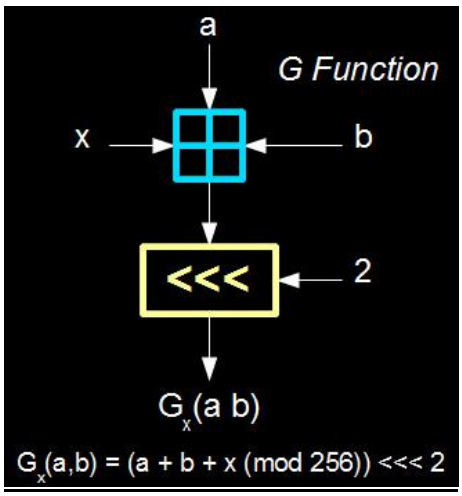
IV) Attaque sur le FEAL-4

Pour parvenir à comprendre comment fonctionne le code, on va devoir utiliser une attaque différentielle à plusieurs niveaux.



Reprenons le schéma de fonctionnement de l'algorithme FEAL :



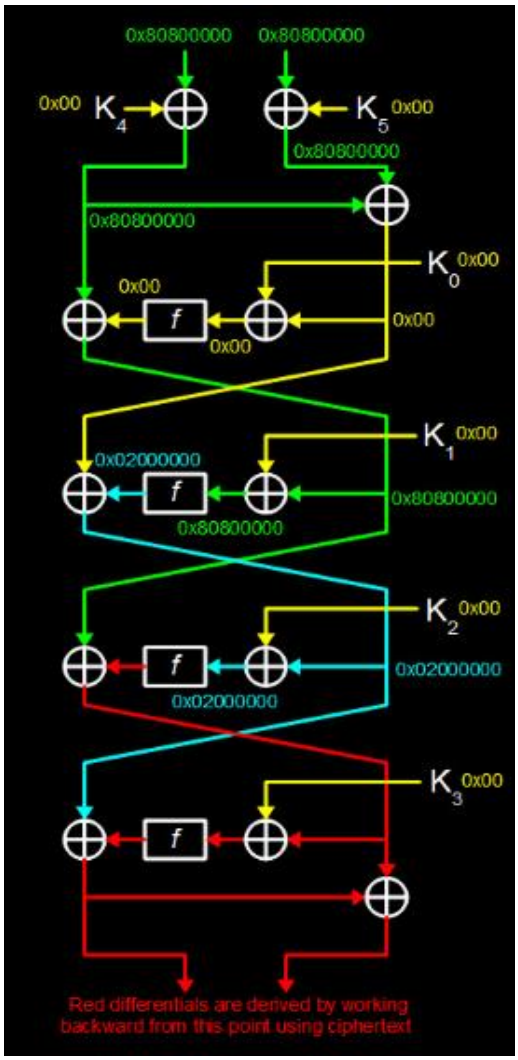


Dans un premier temps, on a besoin de savoir comment vont varier nos différentielles lors de leur chiffrement à travers l'algorithme FEAL. Il semble très difficile de déchiffrer la fonction « round » avec une attaque par force brute. (2^64 tests). Cependant, il a été possible de trouver 2 cas spéciaux :

- Lorsque le texte clair est 0x80800000 le texte chiffré est 0x02000000.
- Lorsque le texte clair est 0x00000000 le texte chiffré est 0x00000000.

Procédons étape par étape pour comprendre ce résultat :

- Quand 2 différentielles effectuent un XOR entre eux on obtient le résultat différentiel. Si des différentielles identiques effectuent un XOR entre eux le résultat obtenu sera 0x000000000.
- On effectue donc le calcul de différentielle avec la fonction G et c'est ainsi que l'on obtient les résultats suivants :
  0x00 + 0x00 --> 0x00
  0x80 + 0x00 --> 0x02
  On se rend compte qu'au final l'addition du x présente une différentielle de 0. En effet les deux nombres en entrée présente une différentielle de 0.



- On fait ensuite passer nos différentielles par la fonction f et l'on sera capable de déterminer la sortie
- Reste ensuite à intégrer le tout au chemin complet et l'on pourra présenter le résultat de la manière suivante :

Il ne restera plus qu'à déchiffrer la dernière clé puisqu'à présent on connait chacune des entrées.

V) Conclusion

Nous venons donc de voir que la cryptanalyse différentielle avait permis de casser des algorithmes tels que DES (partiellement) ou encore FEAL-4. Ce type d'attaque a cassé un grand nombre d'algorithme datant de la même époque que le DES.

Seulement, le DES a eu une bien meilleure résistance à cette attaque que le FEAL-4. On peut donc en conclure que les cryptanalystes qui ont inventé avec le FEAL-4 (en 1987) aurait dû être prévoyant face à de nouvelles attaques innovantes. C'est donc là tout le but de la protection de l'information, d'avoir une longueur d'avance face aux autres algorithmes de chiffrement et face aux nouvelles attaques possibles.

D'autres techniques de cryptanalyse différentielle ont vu le jour suite à celle d'origine à savoir : la cryptanalyse différentielle tronquée, la cryptanalyse différentielle d'ordre supérieur et la cryptanalyse différentielle impossible qui étaient orientées contre les algorithmes de

chiffrement par flots et contre les fonctions de hachage. Néanmoins ce type d'attaques est peu efficace contre les algorithmes de cryptage moderne (complexité trop grande) vu qu'ils sont fait pour résister à ce type d'attaque pour la plupart.

Sources :

- http://theamazingking.com/crypto-feal.php
- https://books.google.fr/books?id=disHBgAAQBAJ&pg=PA83&lpg=PA83&dq=algorithme%20feal&source=bl&ots=qe2tGCAY7Q&sig=6uhaqW-Yvq1gT5ivGeVAXK1J8Ho&hl=fr&sa=X&ved=0ahUKEwjYiNmVtrvTAhXJShQKHae5DfsQ6AEIUTAI#v=onepage&q&f=false
- http://www.i3s.unice.fr/~bmartin/1-CS.pdf
- https://fr.wikipedia.org/wiki/Cryptanalyse_diff%C3%A9rentielle