

Module de Cryptographie:

Blowfish

Auteur: Louis CHABARDES

Cours de Cryptographie (MATH5051)

2006

Robert Erra

Préambule:

L'algorithme de cryptographie Blowfish a été inventé dans le courant de l'année 1993 par Bruce [1] afin de proposer une alternative solide à l'algorithme DES. En effet, celui-ci commence à ne plus assurer une confidentialité suffisante à cette époque. C'est un algorithme de cryptographie de type symétrique, basée sur les réseaux de Feistel. Les buts recherchés par l'auteur lors de la conception de cet algorithme étaient de fournir une alternative libre, gratuite, solide et aisément portable sur tous les types de machines disponibles à l'époque. En effet, il est possible d'implémenter Blowfish, sur des puces dédiées, des processeurs grand public, ainsi que dans des processeurs légers tel que ceux présents sur les cartes à puce.

Aujourd'hui encore, Blowfish est un algorithme qui ne connaît pas de failles révélées. Bien que l'attention soit centrée sur des algorithmes plus complexes tel que AES ou autres, il reste très utilisé dans de nombreux logiciels.

Dans ce document, nous commencerons par une description du principe de fonctionnement des réseaux de Feistel, car ils sont un préalable indispensable à la compréhension de l'algorithme en lui-même, qui sera ensuite décrit. Nous nous intéresserons ensuite à l'état de la cryptanalyse de Blowfish, ainsi qu'à l'état de l'utilisation de celui-ci dans les logiciels actuels.

1. Les réseaux de Feistel:

Les réseaux de Feistel, ou crypte de Feistel, sont une structure particulière de d'algorithme de chiffrement par blocs. Cette structure a été inventée par un l'employé d'IBM Horst Feistel [2]. L'un des principaux avantages du réseaux de Feistel est que les opérations de chiffrements et de déchiffrements sont pratiquement identiques, ce qui limite la complexité du circuit nécessaire pour faire tourner l'algorithme. L'article original décrivant pour la première fois le principe du réseau de Feistel peut être trouvé ici [3]

1.1. Le chiffrement par bloc:

Le terme de chiffrement par blocs signifie que la donnée d'entrée de l'algorithme a une longueur fixe. Ainsi, un message est encodée par partie de la longueur du bloc. Il en résulte un bloc chiffré de même longueur. Le passage dans un bloc assure une transformation unique car celle-ci est dépendante d'une seconde entrée: la clé de chiffrement. Le chiffrement par bloc est opposé au chiffrement par flux d'entrée (stream ciphers en anglais) où l'algorithme agit sur chaque bit d'entrée les uns après les autres.

Les algorithmes de chiffrement par blocs sont donc constitués de deux fonctions: une fonction pour chiffrer, ainsi qu'une fonction pour déchiffrer. Ces deux fonctions ont pour entrées un bloc (chiffré ou clair) de longueur n la longueur du bloc ainsi que la clé de longueur k .

1.2. Principe du réseau de Feistel:

Le réseaux de Feistel est un algorithme qui consiste à appliquer plusieurs fois la même opération à un bloc de texte. A chaque itération ou passe, on applique une fonction f sur la moitié du bloc, et on la combine par le biais d'une opération logique "ou exclusif" (XOR) à l'autre partie du bloc de texte. A chaque passe, la fonction est appliqué sur l'autre partie du texte avec une sous clé différente. La sous clé est tirée de la clé principale de chiffrement. Etant donné que c'est la sous clé qui contrôle le résultat de la fonction, il n'est pas nécessaire que la fonction soit une bijection. Pourtant l'algorithme dans son ensemble est une bijection. La fonction f peut donc être très complexe et n'a pas besoin d'être une bijection. De plus, le fait d'appliquer la fonction avec une sous clé différente à chaque passe sur des données provenant des tours précédent, implique qu'à chaque tour, l'influence de la fonction est plus grande. Ce phénomène également connu sous le nom d'effet d'avalanche est ce qui fait la robustesse des algorithmes de chiffrement ou le nombre de passe est important.

Des schémas explicatifs concernant les réseaux de Feistel peuvent être trouvés ici [4].

1.3. Algorithme utilisant le chiffrement basé sur des réseaux de Feistel:

La majorité des algorithmes récents de chiffrement par blocs utilise la structure des réseaux de Feistel. Comme nous l'avons vu les avantages sont nombreux, ainsi, on retrouve la structure de Feistel dans:

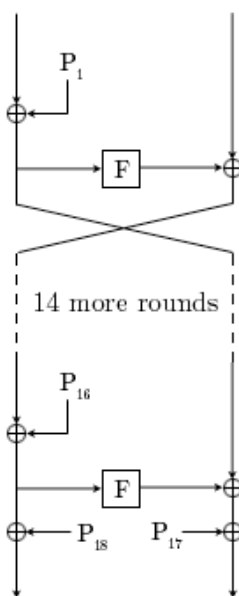
- Lucifer (projet initial d'IBM auquel a participé Horst Feistel)
- DES, 3DES
- Twofish
- Camelia
- RC5

Mais aussi bien d'autres...et bien sur: Blowfish.

2. Blowfish:

Comme nous l'avons vu plus haut, Blowfish est un algorithme de chiffrement symétriques à longueur de clé variables basé sur une structure de type "réseau de Feistel". Maintenant que nous savons ce qu'est un réseau de Feistel, intéressons nous au fonctionnement de Blowfish lui même.

2.1. Fonctionnement général du chiffrement Blowfish:



On peut voir de manière simpliste l'algorithme Blowfish comme un réseau de Feistel à 16 passes. Chaque passe consiste en une permutation dépendante de la sous clé, ainsi qu'une substitution dépendante de la sous clé et des données. Evidemment comme dans tout réseau de Feistel, la sous clé change à chaque passe. La longueur du bloc dans Blowfish est de 64 bits. Le réseau de Feistel de Blowfish est dit équilibré car il agit sur deux blocs de 32 bits chacun. Toutes les opérations de l'opération sont donc des additions ou des "OU exclusifs" sur des mots de 32 bits.

Nous nous occupons ici du chiffrement pur, nous verrons plus tard, qu'une partie importante de Blowfish réside dans l'expansion de la clé principale en 18 sous clé.

Les 16 passes sont typiques d'un réseau de Feistel dans le sens où on applique le schéma classique ci contre.

Toutefois, on peut apercevoir en bas du schéma deux "OU exclusifs" surnuméraires, chacun sur une partie du bloc chiffré. Ceux-ci sont appliqués avec les deux sous-clés supplémentaires générées par rapport au nombre de passes.

2.2. La génération des clés dans Blowfish:

Toute l'originalité de Blowfish ainsi qu'une grande partie de sa robustesse réside dans la manière originale dont sont générées les clés et les boîtes de substitution.

En effet, pour générer les 18 sous-clés nécessaires pour le réseau de Feistel de Blowfish, l'algorithme utilise des décimales de Pi en hexadécimal comme base de génération des clés. Ainsi, chaque sous-clé est passée dans un OU exclusif avec une partie de la clé principale. Ce résultat temporaire sert à chiffrer un bloc de 64 bits initialisé à zéro. Le résultat de ce chiffrement temporaire remplace l'entrée temporaire. Cette suite d'opération est répétée plusieurs fois jusqu'à ce que les 18 sous-clés soient générées, et que les S-Box soient initialisés. Ce calcul assure une bonne initialisation des clés et des boîtes de permutation. Toutefois, il a pour inconvénient d'être très lourd lorsqu'il est nécessaire de changer la clé. Environ 4 kilo-octets de données sont générées lors de cette opération, qui nécessite en tout 521 itérations.

2.3. La fonction F de Blowfish:

Afin de comprendre le fonctionnement de la fonction F de Blowfish, il est nécessaire de connaître la notion de S-Box, ou Boîte de Substitution:

2.3.1. Les boîtes de substitution:

Une boîte de substitution est en fait une table permettant d'effectuer la correspondance entre une donnée d'entrée m avec une donnée de sortie n . La donnée de sortie n n'a pas nécessairement la même taille que la donnée d'entrée. Les premiers algorithmes à chiffrement par bloc utilisaient des boîtes de substitution pré-déterminées. Il convenait donc de bien choisir ses boîtes de substitution afin que celles-ci résistent aux attaques. Par exemple, on sait que les boîtes de substitution de DES ont été conçues pour résister aux attaques [\[5\]](#).

2.3.2. Les boîtes de substitution dans Blowfish:

Comme nous l'avons vu plus haut, le grand intérêt de l'algorithme Blowfish est que les boîtes de substitution sont générées dynamiquement avec les sous-clés. Ceci élimine une faiblesse des boîtes de substitution qui lorsqu'elles sont fixes pour un algorithme.

Blowfish

Les boîtes de substitution de Blowfish ont aussi pour particularités de prendre pour entrées des octets, et d'en ressortir des blocs de 32 bits.

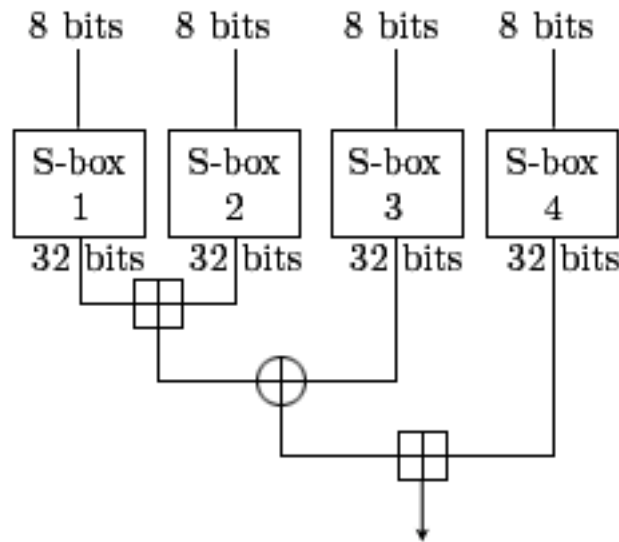
Maintenant que nous connaissons le fonctionnement des boîtes de substitution et que nous avons vu comment celles-ci étaient initialisées. Nous pouvons décrire la fonction F de Blowfish. Tel qu'il est décrit dans l'article original [6] de l'auteur de Blowfish, voici la fonction F de Blowfish:

$$F(xL) = ((S1,a + S2,b \text{ mod } 2^{32}) \text{ XOR } S3,c) + S4,d \text{ mod } 2^{32}$$

où:

- xL: correspond à la partie gauche du bloc en cours
- S1/2/3/4: correspondant à la S-Box 1/2/3/4
- a/b/c/d: correspondent à la division de xL en quatre octets.

D'une manière plus simple, on représente la fonction Blowfish de la manière suivante:



On voit bien ici que Blowfish ne consiste qu'en une suite de permutation (S-Box), d'addition et de Ou Exclusifs, gage de sa relative légèreté en calcul lorsque l'on parle uniquement du chiffage et non de la génération des clés.

3.Perspectives autour de l'utilisation de Blowfish:

Comme nous l'avons en introduction, Blowfish a été conçu pour pouvoir remplacer des algorithmes en déclin, particulièrement DES. Aujourd'hui encore, Blowfish est encore considéré dans son implémentation complète (16 passes) comme sûr. Il n'existe pas à ce jour d'attaques sur Blowfish complet.

En revanche, il a été prouvé que certaines clés dites faibles peuvent être détectées sur des textes longs, encodant plus de 2^{8r+1} blocs de texte [7].

Grâce à ses caractéristiques plutôt bonnes, Blowfish est utilisé dans de nombreux logiciels, on peut en trouver une liste non exhaustive sur le site officiel de l'auteur [8]

Bibliographie:

- [1] Bruce Schneier: <http://www.schneier.com/blowfish.html>
- [2] Horst Feistel: <http://domino.research.ibm.com/comm/pr.nsf/pages/bio.feistel.html>
- [3] Scientific American May, 1973: <http://www.prism.net/user/dcowley/docs.html>
- [4] Feistel Network: http://en.wikipedia.org/wiki/Feistel_network
- [5] Practical S-Boxes Design: [Practical S-Box design, papier de C. Adams](#)
- [6] Blowfish Paper: <http://www.schneier.com/paper-blowfish-fse.html>
- [7] On the weak Keys of Blowfish: http://lasecwww.epfl.ch/php_code/publications/search.php?ref=Vau96a
- [8] Blowfish Products: <http://www.schneier.com/blowfish-products.html>