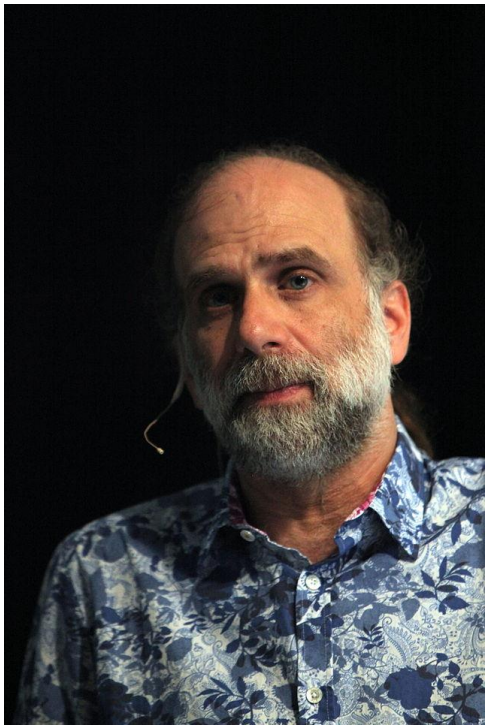
Chiffrement de Blowfish

Réalisé par : REGUIG Hadjer
RADJAI Khadidja

Introduction

- Blowfish est un algorithme de chiffrement symétrique par blocs conçu par Bruce Schneier en 1993.
- Blowfish utilise une taille de bloc de 64 bits et la clé de longueur variable peut aller de 32 à 448 bits.
- Elle est basée sur l'idée qu'une bonne sécurité contre les attaques de cryptanalyse peut être obtenue en utilisant de très grandes clés pseudo-aléatoires.

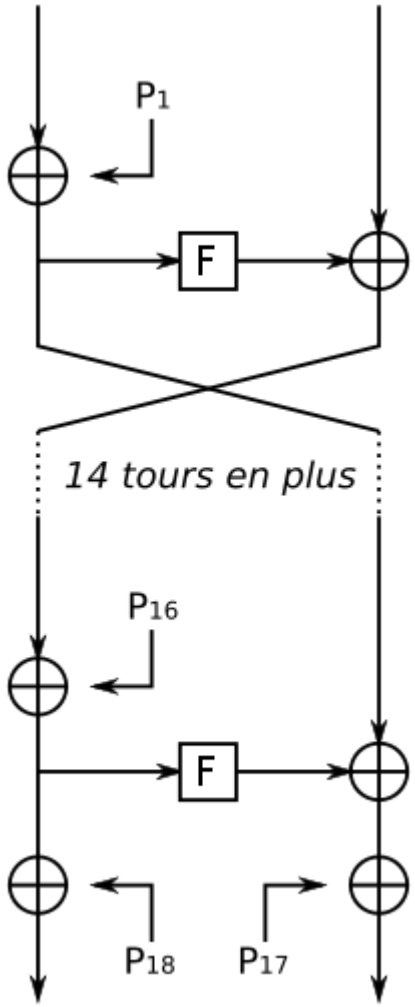


Bruce Schneier

- Blowfish présente une bonne rapidité d'exécution excepté lors d'un changement de clé, il est environ 5 fois plus rapide que Triple DES et deux fois plus rapide que IDEA .
- Malgré son âge, il demeure encore solide du point de vue cryptographique avec relativement peu d'attaques efficaces sur les versions avec moins de tours.
- La version complète avec 16 tours est à ce jour entièrement fiable et la recherche exhaustive reste le seul moyen pour l'attaquer.

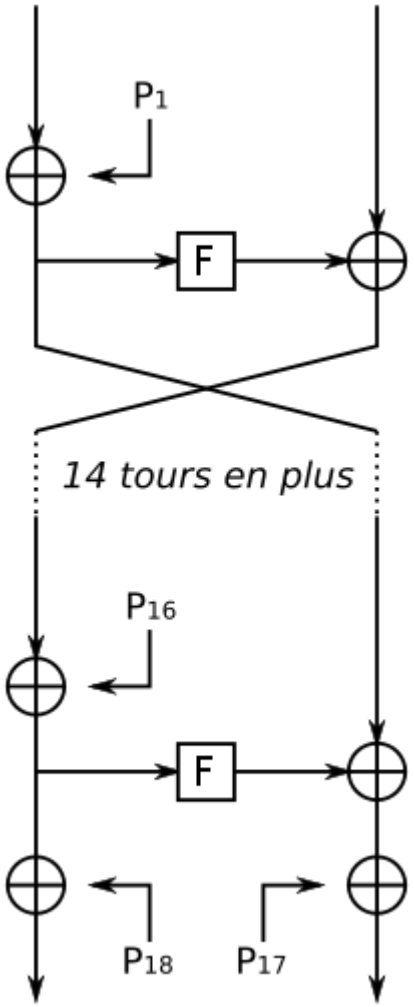
Fonctionnement général du chiffrement Blowfish

- ✓ Blowfish est basé sur un schéma de Feistel avec 16 tours et utilise des S-Boxes de grande taille qui dépendent de la clé
- ✓ Le schéma montre la structure principale de Blowfish. Chaque ligne représente 32 bits.
- ✓ L'algorithme gère deux ensembles de clés : les 18 entrées du tableau *P* et les quatre S-Boxes de 256 éléments chacune.
- ✓ Les S-Boxes acceptent un mot de 8 bits en entrée et produisent une sortie de 32 bits.



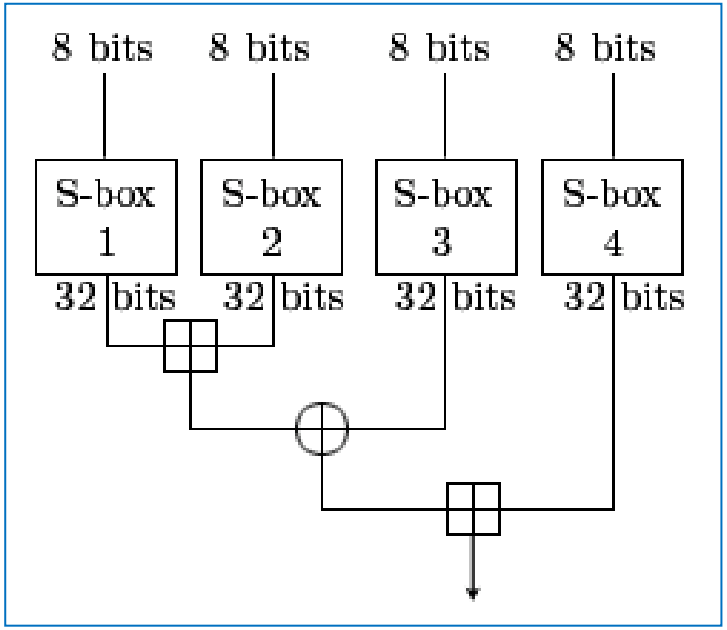
Fonctionnement général du chiffrement Blowfish

- ✓ Une entrée du tableau *P* est utilisée à chaque tour. Arrivé au tour final, la moitié du bloc de donnée subit un XOR avec un des deux éléments restants dans le tableau *P*.



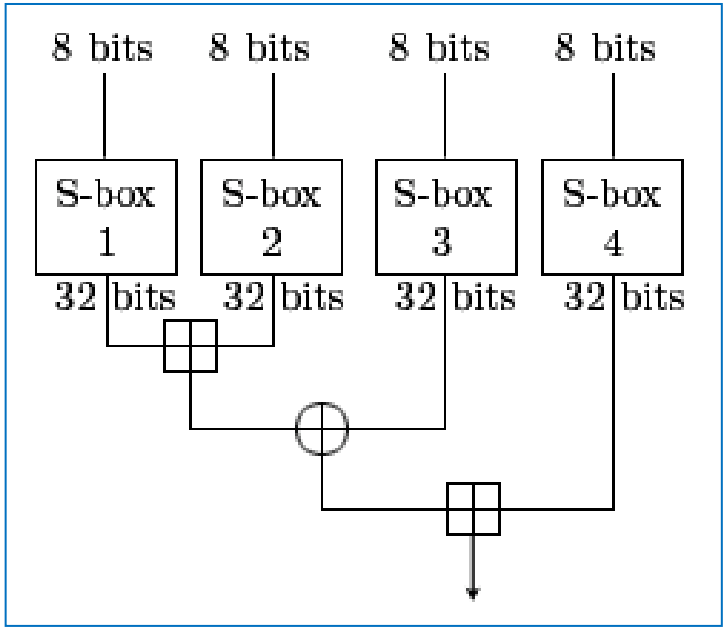
La fonction F de Blowfish

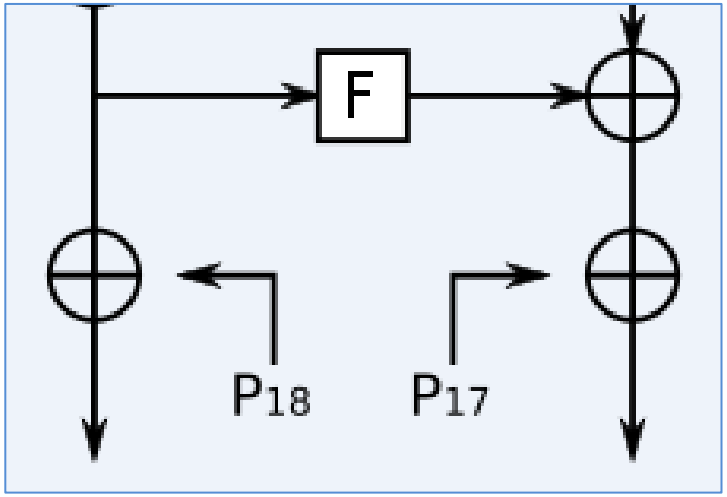
- ✓ Elle sépare une entrée de 32 bits en quatre morceaux de 8 bits et les utilise comme entrées pour accéder aux S-Boxes.
- ✓ Les sorties sont additionnées avec une somme modulo 2^{32} et l'algorithme effectue un XOR entre les deux sous-totaux pour produire la sortie finale de 32 bits.



La fonction F de Blowfish

- ✓ En tant que schéma de Feistel, Blowfish peut être inversé simplement en appliquant un XOR des éléments 17 et 18 du tableau *P* sur le bloc chiffré. Il faut ensuite utiliser les entrées du tableau *P* dans l'ordre inverse.





Génération des sous-clés

- La première étape dans l'algorithme est de séparer la clé originale en un ensemble de sous-clés
- Il y a aussi l'initialisation d'un tableau P et de quatre S-Box de 32 bits chacune.
- Le tableau P contient 18 sous-clés de 32 bits, alors que chaque *S-Box* contient 256 entrées.

Les étapes suivantes sont utilisées pour calculer les sous-clés :

1. Initialisation du tableau P et des *S-Box* avec une chaîne de caractères fixe (chiffres composant la constante PI exprimé en hexadécimal.).
2. Opération XOR entre le tableau P (et ses 18 entrées) et les bits de la clé :

   P[1] XOR (1er 32 bits de la clé),

   P[2] XOR (2e 32 bits de la clé),

   …

   P[18] XOR (Ne 32 bits de la clé)

   Lorsque les bits de la clé sont épuisés, on revient au premier 32 bits.

3. Utilisation de l'algorithme blowfish pour chiffrer Un bloc de 64 bits, tous à zéro en utilisant les sous-clés.
4. La sortie est maintenant P[1] et P[2].
5. Chiffrement des nouveaux P[1] et P[2] avec les sous-clés modifiées.
6. La sortie est maintenant P[3] et P[4].
7. Répéter 521 fois les deux dernières étapes afin de calculer les nouvelles sous-clés pour le tableau P et pour les quatre *S-Box*.

Algorithme de chiffrement

- l'entrée de 64 bits de texte clair est notée "x" et le tableau P est noté Pi/P[i], où "i" est l'itération.

Début chiffrement

Divisé x en 2 : xL et xR

Pour i allant de 1 à 16 faire
    xL = xL XOR P[i]
    xR = F(xL) XOR xR
    Permuter xL et xR
Fin Pour

Permuter xL et xR

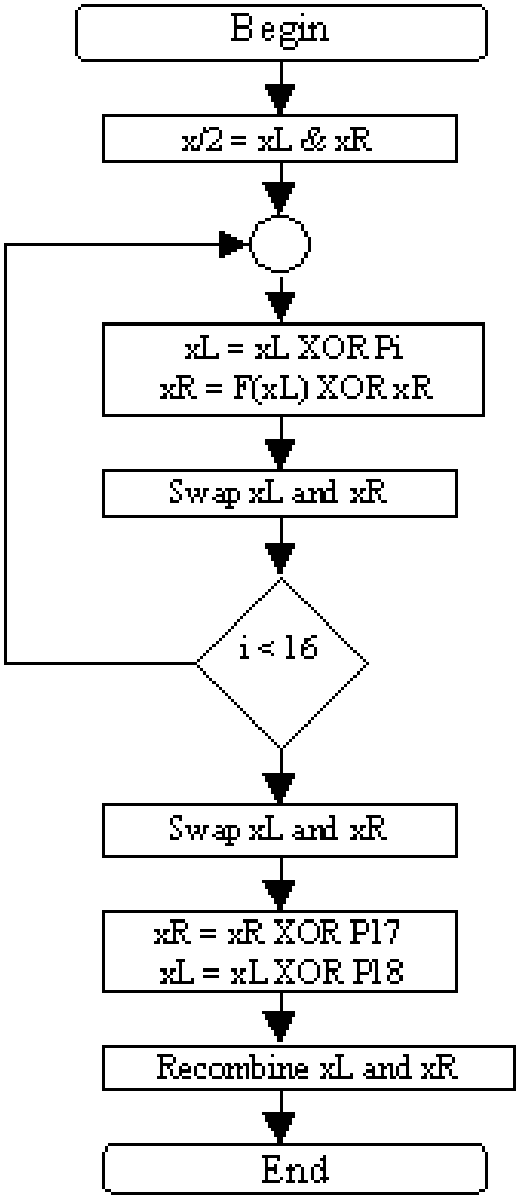
xR = xR XOR P[17]

xL = xL XOR P[18]

x = xL + xR

Retourner x

Fin chiffrement



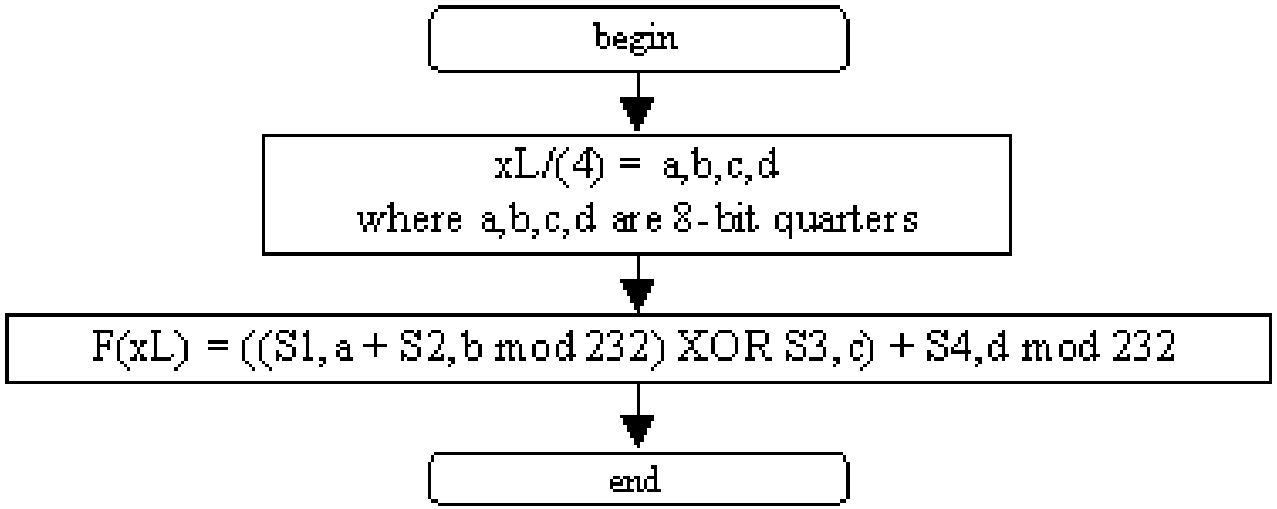
Algorithme de chiffrement

La fonction F( )

Début fonction F(Entrée : xL : 32 bits de données)

Divisé xL en 4 : a, b, c, d
Retourner ((S1,a + S2,b MOD 232) XOR S3,c) + S4,d MOD 2^32

Fin fonction F



conclusion

Mettre en application l'algorithme de Blowfish semble une option pratique pour le chiffrement de donnée étant donné qu'il est destiné à être rapide, compact, simple et relativement sécuritaire. Néanmoins, il convient mieux aux applications logicielles plutôt qu'aux applications matérielles.

Merci a tous