Sécurité des systèmes d'information

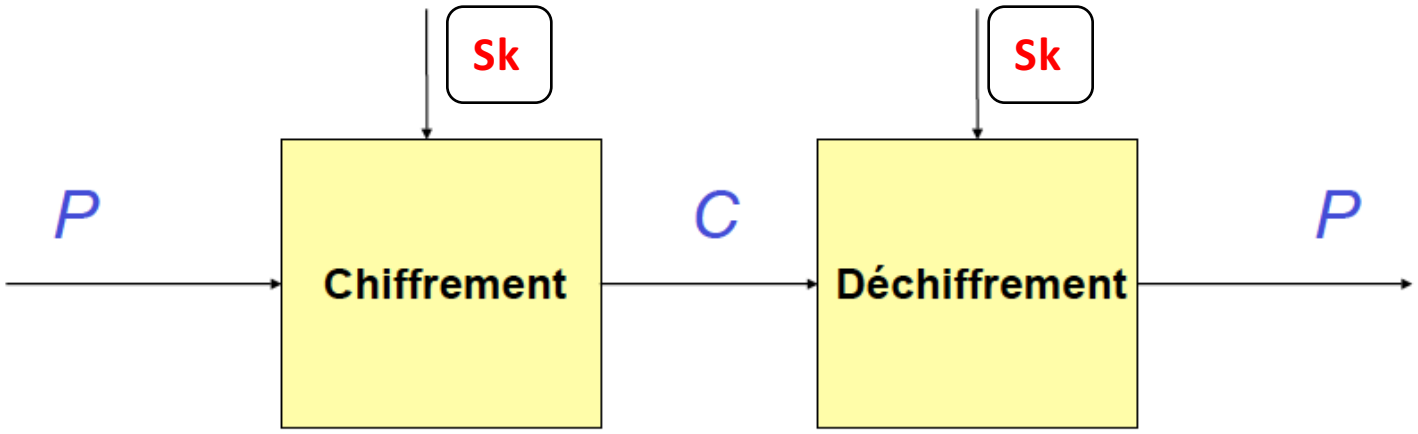
(initiation à la cryptographie)
Partie 2: cryptographie symétrique

université d'Alger 1 -
Benyoucef Benkhedda

Principe

- Chiffrer un message claire m en utilisant une **clé secrète Sk**.

- Seule la clé **Sk** peut être utilisée pour retrouver le message claire m



Types des algorithmes

Chiffrement par blocs

- Le texte claire m est divisé en blocs de taille identique n avant d'être chiffré

  Ex: DES (64bits), AES (128, 256 bits)…etc.

- Le dernier bloc est complété par d'autres caractères en cas où sa taille est inférieure à n

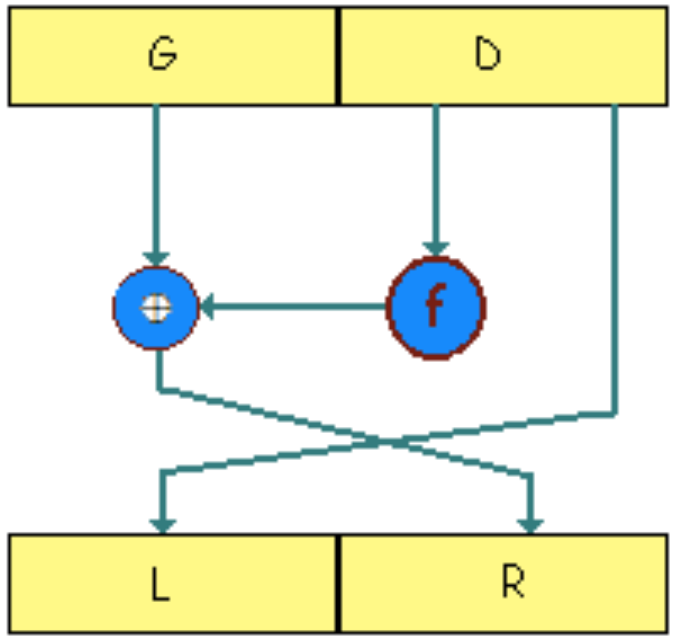
Chiffrement par flots

- Le texte claire m est chiffré directement en le traitant bit par bit
  Ex: RC4, Bluetooth E0/1, GSM A5/1…etc.
- Il n'y a pas de besoin pour compléter le texte claire m

Réseau de Feistel

- C’est une sorte de transformation d’un mot binaire d’une taille n découpé en deux mots de taille identique $\frac{n}{2}$ appelés respectivement (L_i, R_i)
- Cette transformation ce fait comme suit:

$$ENC_{K_i}(m) = \begin{cases} L_{i+1} = R_i \\ R_{i+1} = L_i \oplus f(R_i, K_i) \end{cases}$$



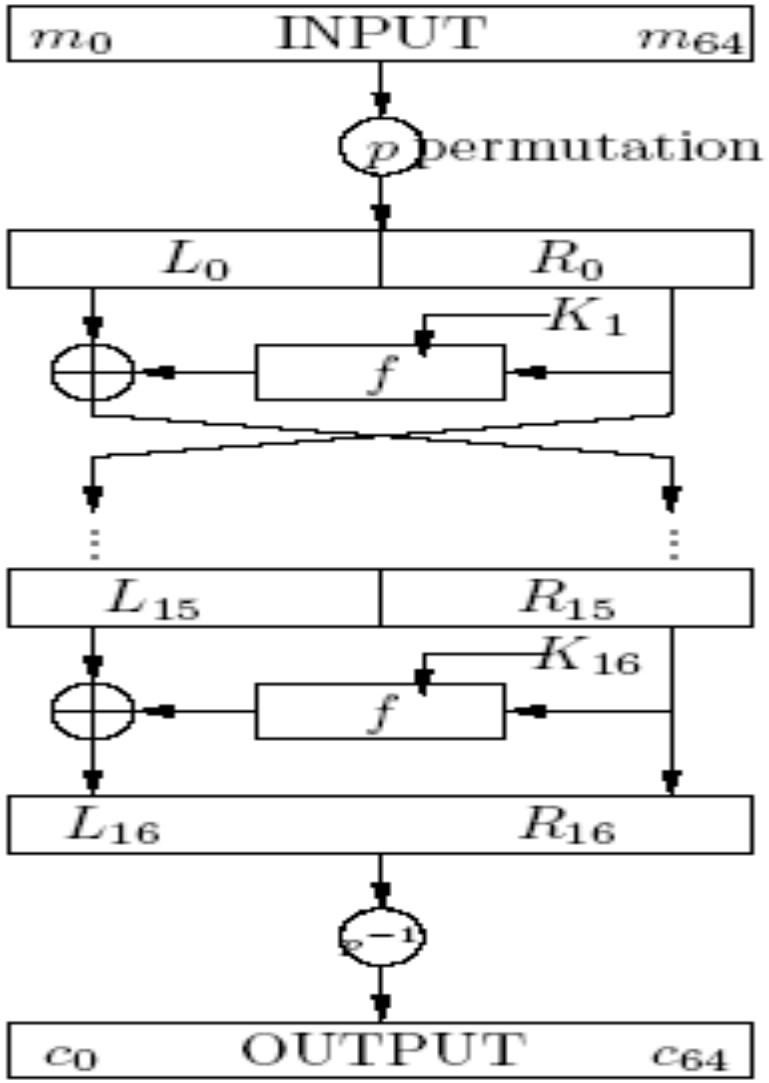
Quelques algorithmes asymétriques

L'algorithme DES

- Utilise le réseau de Feistel sur des blocs de taille 64 bits et une clé de taille 56 bits
- Représenté par un ensemble complexe de permutations et substitution entre les bits du même blocs
- Chaque blocs est chiffré en 16 tours

$$f(R_{i-1}, K_i) = P(S(E(R_{i-1}) \oplus K_i))$$

  - P est une permutation
  - S est une boite permettant la génération d'un mot binaire de taille 4 bits à partir d'un mot binaire de taille 6 bits
  - E est fonction d'expansion



Quelques algorithmes asymétriques

L'algorithme DES

- DES présente quelques faiblesses réside principalement dans sa taille de clé (56bits) qui donne 2^{56} clés possible ce qui est relativement vulnérable contre les attaques par recherche exhaustive
- Des solutions on été adoptées à ce problème en améliorant DES:

L'algorithme triple-DES

- Une version améliorée de DES en utilisant deux chiffrements du même bloc en utilisant la même clé Sk1 séparés par un déchiffrement en utilisant une autre clé Sk2

$$triple - DES_{Sk1,Sk2} = DES_{Sk1} \circ DES_{Sk2}^{-1} \circ DES_{Sk1}$$

- La clé est donc composée de deux clés de 56 bits = 112 bits qui est largement hors de portée des attaques par recherche exhaustive

Quelques algorithmes asymétriques

L'algorithme AES

- Proposé comme une solution au problème de temps d'exécution du DES et triple-DES en utilisant 4 types d'opération en 4 tours sur les blocs
- Il utilise des clés de longue taille: 128, 192 et 256 bits

L'algorithme blowfish

- Qui manipule des blocs de 64 bits et une clé de taille variante entre 32 et 448 bits
- Proposé aussi en différents variantes: blowfish, twofish…etc.

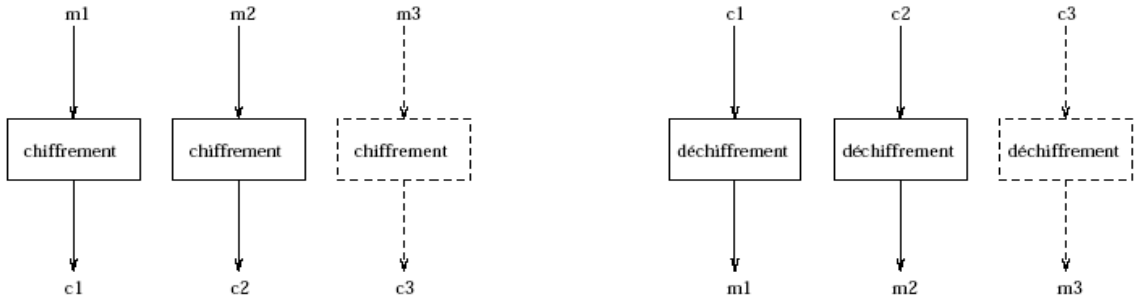
L'algorithme IDEA

- Qui manipule des blocs de 64 bits et une clé de taille 128 bits

Modes de chiffrement par bloc

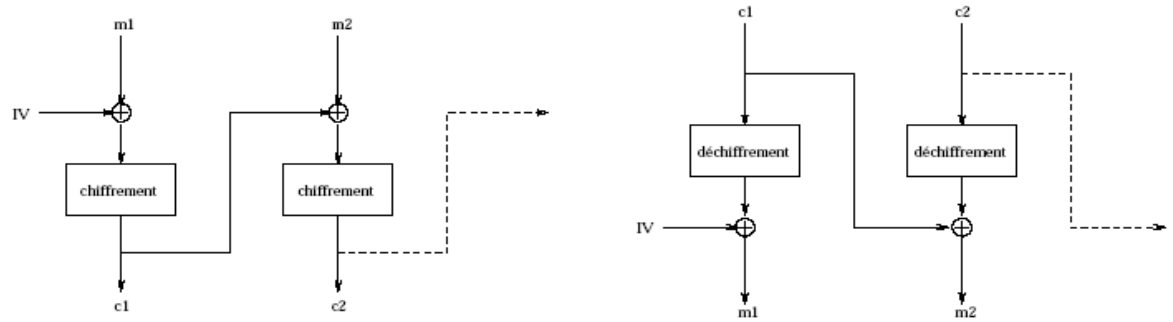
Electronic Code-Book (ECB)

- Chiffrer/déchiffrer chaque bloc indépendamment des autres



Cipher Bloc Chaining (ECB)

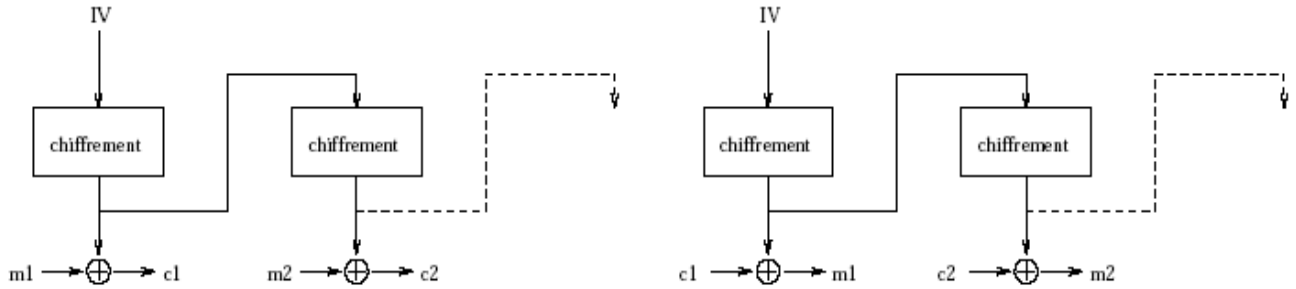
- Masquer chaque bloc par une opération de XOR avec le chiffré du bloc précédent avant de la chiffrer. Le 1er bloc est masqué par un vecteur initiale (IV)



Modes de chiffrement par bloc

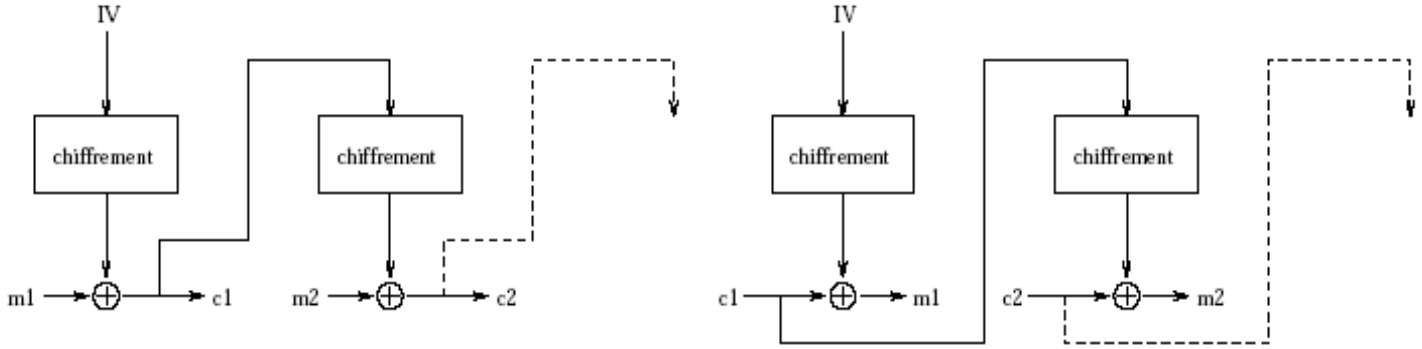
Output FeedBack (OFB)

- Consiste à chiffrer un vecteur initial (IV) puis l'utiliser pour masquer le chacun des blocs sans besoin de les chiffrer. Le IV est chiffré itérativement pour chaque bloc



Cipher FeedBack (CFB)

- Utilise le même principe de OFB sauf que chaque bloc est masqué par le chiffrement du résultat du masquage du bloc précédent



Chiffrement symétrique en pratique

Openssl

- Open source
- Préinstallé dans toute les distributions de Linux
- Simple et pratique
- Contient aussi une bibliothèque en c « openssl.h »

Avantages et inconvénients

Avantages:

- Taux de calcul réduit (on a besoin de calculer une seule clé)
- Sécurité sûre (sans la clé secrète Sk, le déchiffrement est quasi-impossible)

Inconvénients:

- Difficulté d'utilisation dans des communications (difficulté de partage de clé secrète)
- Pour les communications de plus de 2 entités, il fallait:
  - Utiliser la même clé secrète => difficulté d'identification d'émetteur de l'information
  - Utiliser une clé pour chaque 2 entités => beaucoup de ressources
- N'assure que la confidentialité des données