Chiffrement par Bloc: Cryptanalyse Linéaire/Différentielle

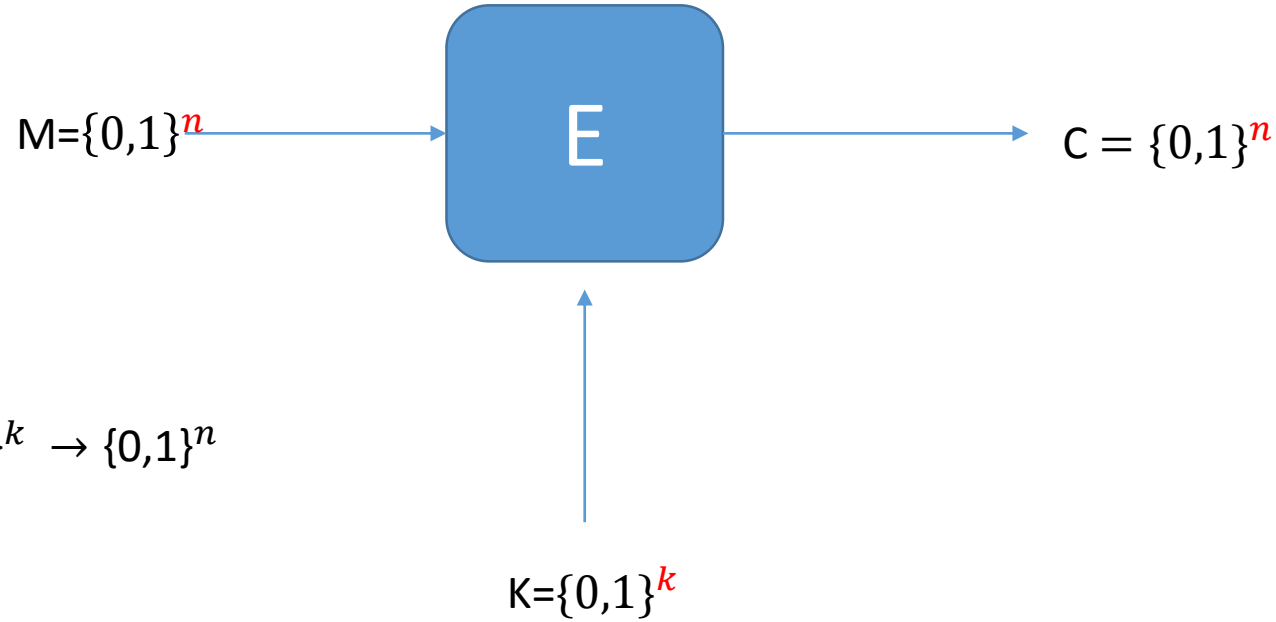
Cours 5: 14/03/2016

Plan du cours

1) Principes généraux:
    - Rappel : block ciphers
    - Attaques génériques
2) Cryptanalyse contre le DES
    - Cryptanalyse différentielle
    - Cryptanalyse linéaire
3) Critères de résistance pour l'AES
4) Autres techniques de cryptanalyse

Rappel : Block Cipher

Définition : Un algorithme de chiffrement symétrique transforme ***un message en clair M*** avec ***une clé secrète K.*** Le résultat est ***un chiffré C***



Notation:
E: {0,1}^{n} × {0,1}^{k} → {0,1}^{n}

Cryptanalyse linéaire

- C'est une attaque à **texte clair connu** contre les protocoles de cryptographie dont la **confusion est faible.**
- Texte clair connu. L'attaquant dispose de un ou plusieurs message(s) clair(s) avec le(s) message(s) crypté(s) correspondant, tous cryptés avec la même clé. L'attaquant cherche à retrouver (de l'information sur) la clé.

- Une idée. Trouver des relations linéaires de dépendance de probabilités exceptionnelles entre les bits d'entrée et de sortie.

En effet, une relation linéaire ne peut pas être vraie pour tous les messages sinon le protocole a une faiblesse.

Sécurité

- Idéalement, C ne doit laisser fuir aucune information sur M ou sur K

Mais la sécurité n'est jamais «parfaite».

Un attaquant connaissant C et M peut « tester toutes les clés » $\rightarrow 2^K$

- Modèle de la boîte noire

la recherche exhaustive

Fonction à sens unique:

f: E → *F*

x → f(x): Facile

Etant y=f(x), trouver x: Difficile

la recherche exhaustive: consiste à calculer les images par f de tous les éléments x de *E* jusqu'à en trouver un qui donne y.

Cette technique est très coûteuse en temps de calcul et doit être répétée pour chaque nouvelle valeur de y ;

Utiliser un pré calcul ?

La recherche par dictionnaire

Phase de précalcul (une fois pour toutes fait indépendamment de y)

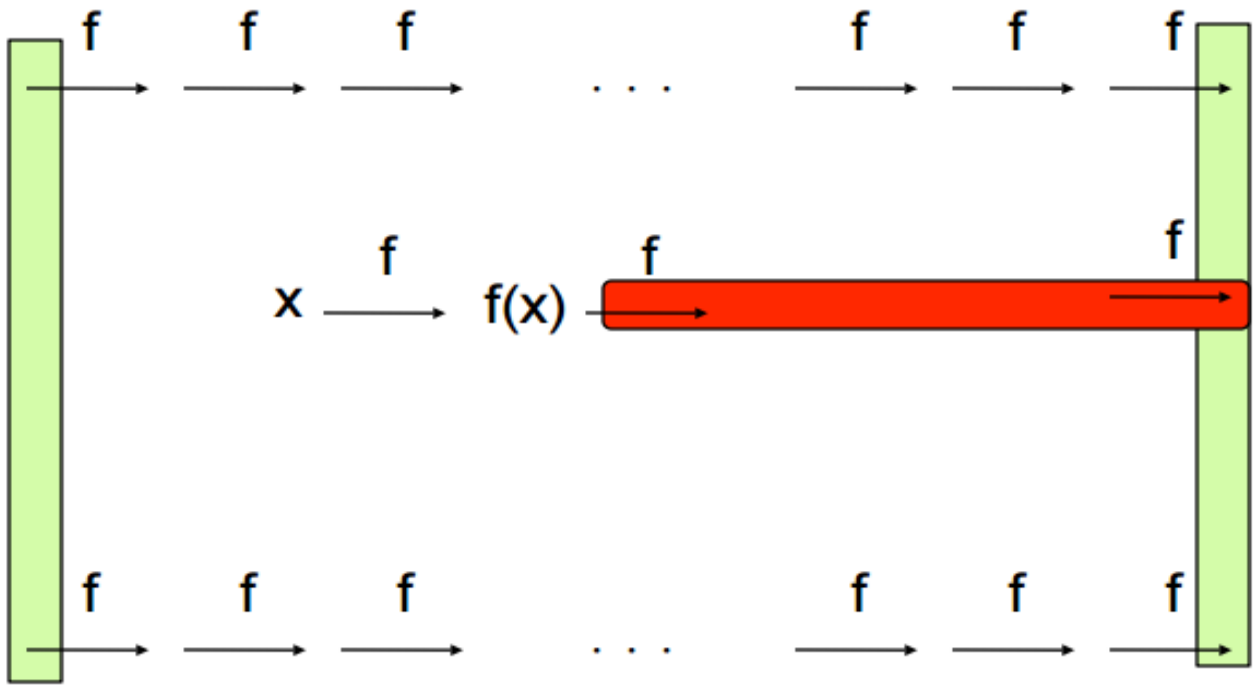
1. Calculer à l'avance les f(x)
2. Stocker en mémoire tous les couples (x, f(x)) en les triant suivant la valeur de f(x)

Phase du calcul:

Trouver x à partir de y est extrêmement rapide (avantageux par rapport à la méthode précédente).

L'inconvénient: la taille de la mémoire utilisée, puisqu'il faut stocker tous les couples (x, f(x)) ➔ Compromis temps/mémoire de Hellman

Compromis temps/mémoire de Hellman



Pour inverser f(x) :

mémoire nécessaire = #lignes temps nécessaire = #colonnes

(#lignes)x (# colonnes) = espace des clés

Recherche exhaustive (ou attaque par Force Brute)

Complexité théorique de l'attaque

2^{31}	**Cycles / seconde (2GHz)**
2^{56}	Recherche exhaustive DES (RC5 - 1997 – distributed.net)
2^{64}	« Record » de recherche exhaustive (RC5 – 2002- distributed.net)
2^{72}	Tentative en cours (RC5 – distributed.net)
2^{128}	Sécurité de AES

Compromis temps/mémoire de Hellman

Clé de k bits

Temps 2^2k/3

Mémoire 2^2k/3 | Précalcul 2^k

Exemple DES : 56 bits

- Précalcul 2^56
- Temps 2^39
- Mémoire 2^39

Introduction: Cryptanalyse

- Les deux principales méthodes connues de cryptanalyse des chiffrements par blocs symétriques sont la cryptanalyse différentielle et la cryptanalyse linéaire.
- Elles exploitent toutes deux des *comportements statistiques non uniformes* dans le processus de chiffrement. La cryptanalyse différentielle date de 1990 et est due à *Biham* et *Shamir*. La cryptanalyse linéaire date de 1992 et est due à Matsui.

Appelons x le texte clair, y son chiffré.

Fonctions linéaires/non linéaires

- Soit L une fonction linéaire

$\{0,1\}^n \rightarrow \{0,1\}^n$

$L(x \oplus y) = L(x) \oplus L(y)$

Alors la différentielle de L est très simple, en tout point x :

$L^*(\Delta) = L(x \oplus \Delta) \oplus L(x) = L(\Delta)$: Donc $L^* = L$ en tout point

- Soit L une fonction affine

Ajout d'une sous-clé K: $L_k(x) = x \oplus K$

$L^*(\Delta) = L_k(x \oplus \Delta) \oplus L_k(x)$

$= (x \oplus \Delta \oplus K) \oplus (x \oplus K)$

$= \Delta$

La différentielle $L_k{}^*$ en tout point est indépendant de K !

Fonctions non-linéaires

- La différentielle F* dépend du point x concerné

Exemple F : $\{0,1\}^2 \rightarrow \{0,1\}^2$

$$F(x,y) = (x.y, x)$$

Fonctions non-linéaires

Différentielle en (0,0)

F*(0,0) = (0,0)

F*(0,1) = (0,0)

F*(1,0) = (0,1)

F*(1,1) = (1,1)

Différentielle en (I,I)

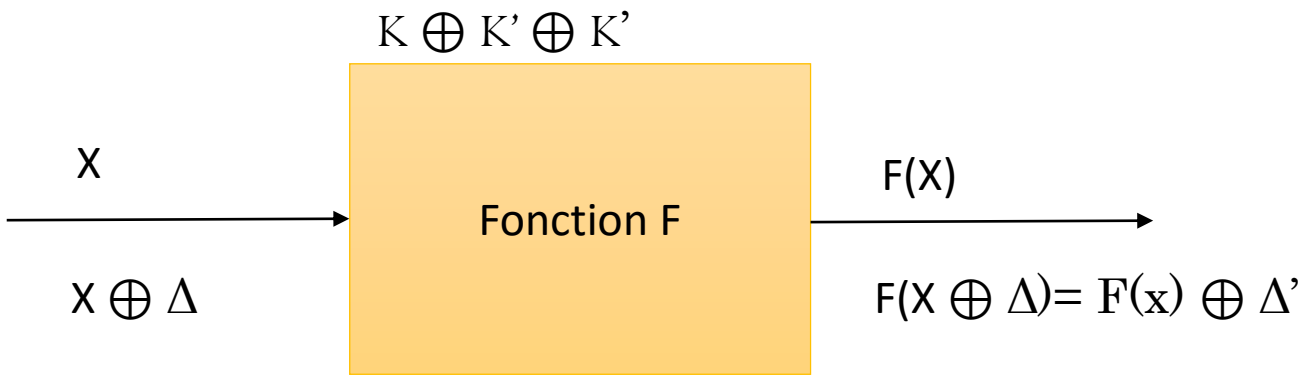
F*(0,0) = (0,0)

F*(0,1) = (1,0)

F*(1,0) = (1,1)

F*(1,1) = (1,1)

Cas des block ciphers



Pour résumer

- Fonctions linéaires :
  - Différentielle prévisible de façon exacte
- Fonctions affines (XOR de sous-clé)
  - Différentielle indépendante de la clé
- Fonctions non-linéaires
  - On ne peut rien dire de façon générale
  - Donc on ne peut pas calculer directement la différentielle pour tout le block cipher
- On adopte une approche statistique

Cryptanalyse différentielle

- Il s'agit d'une attaque à clairs choisis.

- La cryptanalyse différentielle s'intéresse à l'évolution les différences $x_i + x'_i$ pour deux clairs x, x' . On détermine que, si $x_i + x'_i = \alpha$, alors $x_{r-1} + x'_{r-1} = \beta$ avec une probabilité non négligeable.

- On utilise cela pour déterminer la clé inconnue k_r à partir de plusieurs messages x et de leurs chiffrés x_r obtenus par E_k .

Cryptanalyse 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 une loi de groupe, en général le xor bit à bit.

Cette attaque utilise la faiblesse potentielle de la fonction itérée f dans une dérivation à l'ordre 1.

Exemple: cryptanalyse différentielle

- La cryptanalyse différentielle utilise la comparaison du XOR de deux entrée avec le XOR des deux sorties correspondantes. On considère x'= (x'_1, x'_2 . . . x'_n) et x"= (x"_1, x"_2 . . . x"_n) deux entrées et y'= (y'_1, y'_2 . . . y'_n) et y"= (y"_1, y"_2 . . . y"_n) les sorties correspondantes.

- On note $\Delta x = x' \oplus x"$

$$\Delta y = y' \oplus y"$$

Exemple: cryptanalyse différentielle

- 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.

La cryptanalyse différentielle exploite le fait qu'il peut arriver qu'un Δy particulier arrive avec une très grande probabilité ,

p >> 1/2^n, d'un Δx particulier.

Le couple (Δx, Δy) est appelée une différentielle.

Méthodologie

- On suppose que le cryptanalyste dispose d'un grand nombre de quadruplets (x', x'', y', y'') où la valeur de Δx est fixée et que tous les textes sont chiffrés avec la même clef inconnue K.

- Pour chacun des quadruplets on commence par déchiffrer y' et y'' en utilisant toutes les sous clefs candidates pour le dernier étage.

- On commence par regarder les caractéristiques différentielles des S-boites. On remarque que dans ce cas $\Delta y = S(x') \oplus S(x'')$.

Approche statistique

- Caractéristique différentielle
  - Différence Δ en entrée de F
  - Différence Δ' en sortie de F
  - Probabilité p associée (moyennée sur tous les x possibles)

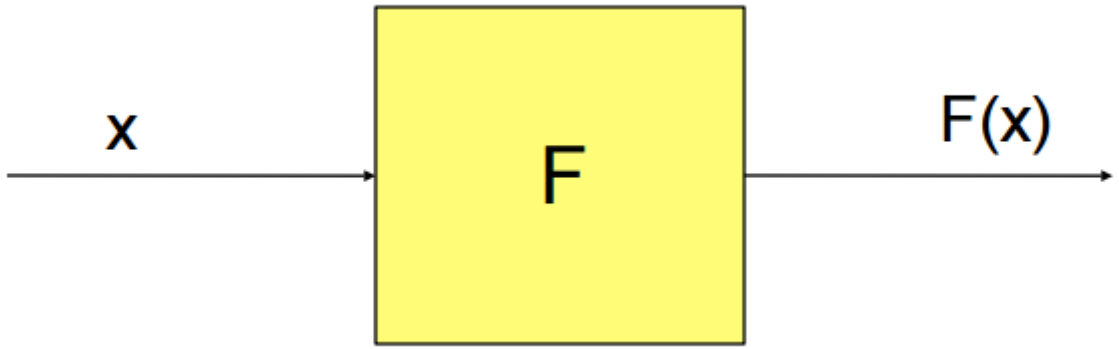
- Notation : $\Delta \rightarrow \Delta'$ [proba = p]

Cryptanalyse linéaire

- Idée générale proche de la cryptanalyse différentielle (attaque à clairs choisis)
- On utilise des approximations linéaires des algorithmes de chiffrement par bloc
- La cryptanalyse linéaire consiste à simplifier l'algorithme de chiffrement en faisant une approximation linéaire.
- En augmentant le nombre de couples disponibles, on améliore la précision de l'approximation et on peut en extraire la clé.
- La cryptanalyse linéaire s'intéresse aux relations linéaires entre les bits au cours de l'algorithme.

Forme linéaire

- Soit F : $\{0,1\}^n \rightarrow \{0,1\}^n$
- Une forme linéaire $\lambda : \{0,1\}^n \rightarrow \{0,1\}^n$ est définie par un masque $a = (a_1 \dots a_n)$ • $\lambda\,(x_1 \dots x_n) = a_1.\,x_1 \oplus \dots \oplus a_n.\,x_n$



Caractéristique linéaire

- Une caractéristique linéaire de F est un couple de formes linéaires (λ_1, λ_2) ayant pour masques associés a_1 et a_2 telles que

$\lambda_1 (x) = \lambda_2 F(x)$

- avec probabilité p (prise sur tous les x possibles)

Méthodologie

- On suppose qu'il existe une $\mathbf{F_2}$-combinaison linéaire des bits d'entrée et de sortie qui ait lieu avec une probabilité nettement supérieure ou nettement inférieure à $1/2$. Autrement dit qu'il existe $i_1, i_2, \ldots, i_U$ et $j_1, j_2, \ldots, j_v$ tels que si $x = (x_1 \ldots x_n$ sont les bits d'entrée (du message en clair) considérés comme des variables aléatoires définies sur $\{0, 1\}$ et $y_1 \ldots y_n$ sont les bits de la sortie (du message chiffré) considérés comme des variables aléatoires définies sur $\{0, 1\}$ on ait $\Pr\{ X_{i_1} \oplus X_{i_2} \oplus \cdots \oplus X_{i_n} \oplus Y_{i_1} \oplus Y_{i_2} \oplus \cdots \oplus Y_{i_m} = 0\} >> 1/2$ Le principe est alors d'approximer une partie de l'algorithme de chiffrement par cette combinaison linéaire sur $\mathbf{F_2}$.

Table des app. linéaires

- En général, deux formes linéaires aléatoires sont égales avec probabilité 0,5

- On s'intéresse donc à l'écart avec p = 0,5 aussi appelé biais ε

$a1 \rightarrow a2\ [proba = 0.5 * (1+ \varepsilon)]$

$a1 \rightarrow a2\ [biais = \varepsilon]$

- Plus $|\varepsilon|$ est grand, mieux c'est pour l'attaquant

Méthodologie

- Par analogie avec la cryptanalyse différentielle, on note souvent $a\_1 \rightarrow a\_2$ [proba = p]

- On utilise une table des caractéristiques linéaires (↔ table des différences)

Exemple: $(1,0) \rightarrow (0,1)$ [proba = 1]