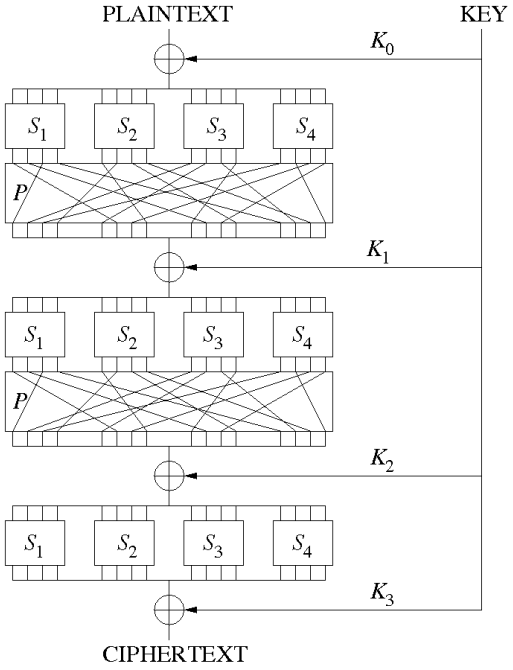
INFO910 : cryptologie
TD 3 : cryptographie symétrique, chiffrements par blocs

Pierre Hyvernat
Laboratoire de mathématiques de l'université de Savoie
bâtiment Chablais, bureau 17, poste : 94 22
email : Pierre.Hyvernat@univ-smb.fr
www : http://www.lama.univ-smb.fr/~hyvernat/

Exercice 1 : chiffrement par blocs

Question 1. AES utilise des blocs de taille 128 bits. Quelle serait la taille d'une table représentant une "vraie" substitution aléatoire sur les suites de 128 bits ?

Question 2. Un réseau substitutions / permutations sur 16 bits à 3 tours est représenté ci dessous.



Les boites $S_1, \dots, S_4$ sont des substitutions (dans ce cas sur les suites de 4 bits), et les P sont des permutations des 16 bits.

Rappel :
- diffusion : *la modification d'un bit du message clair doit modifier, en moyenne, la moitié des bits de sortie*,
- confusion : *chaque bit du résultat doit dépendre de plusieurs bits de la clé.*

Que pensez vous d'un système de chiffrement qui n'utiliserait que des substitutions, sans permutations ?

Que pensez vous d'un système de chiffrement qui n'utiliserait que des permutations, sans substitutions ?

Question 3. Dans un réseau de Feistel, chaque bloc est découpé en 2 parties de même taille L et R. Le tour i est calculé avec

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

où K_i est la sous-clé générée pour le tour i.

- Donnez la formule de déchiffrement d'un seul tour. Quelles propriétés doit satisfaire la fonction F ?
- Montrez qu'on peut déchiffrer avec le même procédé, en prenant les sous-clé dans l'ordre inverse.

Exercice 2 : mode CBC, "padding attack"

Rappel : en mode CBC, le chiffrement de $M_1 M_2$ se fait avec

- $C_0 = IV$
- $C_i = E_k(M_i \oplus C_{i-1})$

Question 1. Donnez la méthode de déchiffrement correspondante.

Lorsqu'on utilise un chiffrement par blocs, le message initial doit être complété pour avoir une taille multiple de la taille des blocs. S'il manque b octets, une manière de faire ceci est de compléter le dernier bloc avec des octets `0xbb` où `0xbb` est l'octet représentant b. (Attention, si le dernier bloc est complet, on ajoute un nouveau bloc complet d'octets `0xbb`!)

On suppose qu'un attaquant peut savoir si un texte chiffré est "correct" : par exemple, un serveur peut renvoyer une erreur `Incorrect data, aborting transaction`.

L'attaquant peut alors monter l'attaque suivante (S. Vaudeney, 2002) pour décrypter un message chiffré de 3 blocs $C_0 \,||\, C_1 \,||\, C_2$ qu'il a intercepté.

Question 2. Que pouvez-vous dire sur la taille du message initial (clair) correspondant à $C_0 \,||\, C_1 \,||\, C_2$?

Question 3. Comment est déchiffré le dernier bloc du message $C_0 \,||\, (C_1 \oplus \Delta) \,||\, C_2$ par le serveur ? (Vous pouvez supposer que Δ ne contient qu'un seul bit à `1`.)

Question 4. Que se passe t'il si l'attaquant envoie le message $C_0 \,||\, (C_1 \oplus \texttt{0xff0000...00}) \,||\, C_2$? (Il a uniquement modifié le premier octet du bloc C_1.)

Indice : dans quels cas est-ce que le serveur renvoie une erreur de déchiffrement due à un mauvais remplissage ?

Question 5. Si le serveur répond `OK`, l'attaquant envoie $C_0 \,||\, (C_1 \oplus \texttt{0x00ff0...0}) \,||\, C_2$, puis $C_0 \,||\, (C_1 \oplus \texttt{0x0000ff0...0}) \,||\, C_2$, etc.

Que peut-il en déduire ?

L'attaquant souhaite maintenant trouver la valeur du dernier octet o du bloc M_2 du message clair (avant remplissage).

Question 6. Comment l'attaquant peut-il manipuler $C_0 \,||\, C_1 \,||\, C_2$ pour modifier la valeur des octets de remplissage en $b+1$?

Question 7. Envoyer un message avec des octets de remplissage à $b{+}1$ force le serveur à regarder le dernier octet o du dernier bloc clair pour vérifier la validité du (nouveau) remplissage. Que se passe t'il si l'attaquant utilise $C_1 \oplus \Delta \oplus \delta_i$, où

- Δ est choisi pour passer les octets de remplissage à $b+1$ (question précédente)
- et δ_i permet de faire un xor entre le dernier octet (correspondant à o) et $0 \leq i < 256$.

Indice : considérez le cas où le dernier octet déchiffré (correspondant au i) vaut exactement $b{+}1$.

Question 8. Décrivez comment l'attaquant peut découvrir ainsi tous les octets du message clair.