**ESI-SBA** **Promotion :**3ième Année Ingénieur

**Fiche TD N°2 : Principes et algorithmes du Chiffrement Symétrique**

**Exercice 1**

Un schéma simplifier d’IDEA est basé sur une variante du mécanisme de Feistel, décrite ci-dessous :



1. Écrire les équations donnant l'expression du chiffré (L1, R1) en fonction du clair (L0, R0) ;
2. Montrer que ce schéma est inversible quelle que soit la fonction F et donner les formules décrivant le déchiffrement ;
3. Décrire un schéma de Feistel à trois tours (sans la permutation finale des deux blocs L3 et R3) qui lui est équivalent ;
4. Montrer comment distinguer la fonction de chiffrement (L0, R0) → (L1, R1) d'une transformation aléatoire.

**Exercice 2 (Mode CBC)**

Soit m un message composé de L block AES (L=100 par exemple). Après chiffrement de m en utilisant le mode opératoire CBC, le block N° L/2 est corrompu durant la transmission, alors que tous les autres blocks sont reçus correctement.

1. Après déchiffrement des blocks reçus, combien de block en claire sera corrompu ? Quel est la valeur de ce nombre si le mode opératoire utilisé est OFB ?
2. Montrer que si deux blocs du cryptogramme, ci et cj sont identiques, avec 1≤i < j ≤ L alors on dispose d'une information sur les blocs mi et mj (montrer que mi ⊕ mj est connu).
3. Dans le cadre d’utilisation du mode opératoire de chiffrement CBC, on suppose que les blocs d'un cryptogramme sont distribués aléatoirement dans l'ensemble {0, 1}n des blocs (n est la taille de chaque bloc). Sachant que la probabilité pour qu'il existe deux blocs de n bits égaux dans un cryptogramme de t blocs est équivalent à , quelle condition doit satisfaire la taille *n* des blocs pour que cette probabilité soit inférieure à 1/1000 sur des cryptogrammes d'un million de blocs ?
4. Un attaquant qui a le pouvoir de modifier le texte chiffré connait la valeur en clair du premier bloc M1 et veut le remplacer par un autre A1 sans affecter les autres. Comment peut-il faire ceci si le mode utilisé est OFB? peut-il le faire si le mode est CBC ? (Justifiez)

**Exercice 3.**

1. Soit le mode opératoire décrit par la formule suivante : C0=IV ; Ci=EK(Mi)⊕Ci-1

- Schématisez ce mode.

* + Donnez la formule de déchiffrement**.**
  + Expliquez pourquoi ce mode n’est par recommandé (donnez une situation dans laquelle certaines informations sur le texte en clair peuvent être déduites à partir du texte chiffré seulement).

**Exercice 4 :**

On considère un diagramme de Feistel à deux tours sur des chaînes de bits avec deux fonctions f1 et f2 correspondants à chacun des tours.

1. Donnez sous forme d’un schème la représentation de ce diagramme.
2. On considère f1(a)=a⊕1011 et f2(a)=not(a)⊕0101 pour tout chaîne a sur 4 bits :
   1. Calculer l’image de la chaîne 11010011 par ce diagramme. Déterminer une chaîne de 8bits dont l’image par le diagramme est elle-même.

**Solutions des Exercices de la fiche de TD N°2 : Principes et algorithmes du Chiffrement Symétrique**

**Exercice 1 :**

1)La valeur qui rentre dans la fonction est L0⊕R0 donc on a :

2) Le schéma est inversible si on peut calculer L0 et R0 en fonction de L1 et R1. Si on réalise une addition en Xor entre les deux équations on obtient :

L1⊕R1=L0⊕R0⊕F(L0⊕R0)⊕F(L0⊕R0) =L0⊕R0⊕0= L0⊕R0

Donc :

Donc on n’a pas besoin de F-1 pour inversé le schéma (décrypté).

3) Intuitivement, ce schéma d'IDEA comporte 3 XOR. Comme chaque tour d'un Feistel en comporte 1, on sent que 3 tours seront nécessaires.

Pour pouvoir calculer F, il faut avoir formé le XOR de L0 et R0 ; ce sera le rôle du premier tour (donc, la fonction de confusion de ce premier Feistel sera l'identité). On voit alors qu'au deuxième tour, c'est la fonction F qui doit servir de fonction de confusion. Le troisième tour fait apparaître la nécessité d'utiliser à nouveau l'identité.

****

On a donc successivement : (G1,D1) = (D0,G0⊕D0), puis G2 = D1 et D2 = G1⊕ F(D1) = D0 ⊕ F(G0⊕ D0). Et finalement :

On a donc bien la même relation entre (G3,D3) et (G0,D0) qu'entre (L1,R1) et (L0,R0).

4) On a vu que ce schéma conservait la valeur du XOR des deux blocs : autrement dit on a L0⊕R0 = L1⊕R1. Ce n'est certainement pas le cas si une fonction aléatoire est utilisée pour calculer (L1,R1) à partir de (L0,R0).

**Exercice 2**

1. En utilisant le mode CBC deux blocks claire serons corrompu après déchiffrement : le block N° l/2 et le suivant (voir le schéma du mode CBC). En utilisant le mode OFB,  seulement le block N° l/2 sera corrompu (voir le schéma du mode OFB).
2. On suppose que ci=cj. Selon le schéma de déchiffrement du mode CBC, les deux blocs mi et mj sont obtenus comme suit :

mi=D(ci)⊕ci-1 et mj=D(cj)⊕cj-1 (D est la fonction de déchiffrement utilisée)

ceci implique que mi⊕mj=D(ci) ⊕ci-1⊕D(cj) ⊕cj-1=D(ci) ⊕D(cj) ⊕ci-1⊕cj-1

ci=cj⇒D(ci)=D(cj)⇒D(ci) ⊕D(cj)=0⇒mi⊕mj=ci-1⊕cj-1

puisque ci-1 et cj-1 sont connue ; ci-1⊕cj-1 est donc connue et ainsi mi⊕mj l’est aussi.

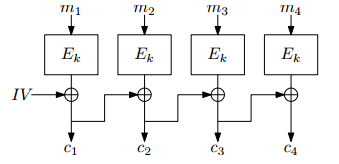
1. Si le mode utilisé est OFB, remplacer le bloc M1 par A1 est possible :

Le chiffrement du premier bloc M1 est égal à C1=M1 ⊕Ek(IV||1) ⇒ Ek(IV||1)= M1 ⊕ C1, l’attaquant ensuite remplace C1 par C1’=A1⊕Ek(IV||1)=A1 ⊕ M1 ⊕ C1. Les autres blocs ne sont pas affectés.

Si CBC est utilisé, il n’est pas possible de modifier C1 car la modification d’un bloc entraine celle des blocs suivants. Par contre l’attaquant peut modifier C0 en cherchant la valeur C0’ qui vérifie : A1=DK(C1)⊕C0’ et ensuite remplacer C0 par C0’. Il est facile de noter que C0’=A1⊕DK(C1)= A1⊕M1⊕C0. En remplaçant C0 par C0’ le bloc A1 est obtenu lors du déchiffrement au lieu de M1 sans affecter les blocs suivants. Donc c’est aussi possible avec CBC !

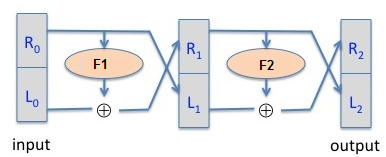
1. La probabilité que qu’il existe deux blocs identiques dans une suite de t blocs est égale à 1-(la probabilité que tout les t blocks sont différents) :

**Exercice 3**

****

1. Le déchiffrement se fait comme suit : C0=N ; Mi=Dk(Ci⊕Ci-1).
2. Si m2=m3 alors on obtient c1=c3 ce qui est détectable à partir du texte chiffré seulement. A partir de deux blocs chiffrés identiques ci et ci+2 on peut déduire que les deux blocs clairs mi+1 et mi+2 sont identiques.

**Exercice 4 :**

****

**2.** On calcule les formules donnant le mot de sorite L2.R2 en fonction du mot d’entré R0.L0 (le . dénote la concaténation des mots de bits) :

R2=f2(f1(R0)⊕L0)⊕R0=not(L2)⊕0101⊕R0

=not(R0⊕1011⊕L0)⊕R0=1011⊕L0

L2=f1(R0) ⊕L0=R0⊕1011⊕L0

Ainsi pour R0=1101 et L0=0011 on obtient L2=0101 et R2=0011 donc l’image de 11010011 (i.e., R0L0) par ce diagramme est finalement 01010010 (i.e., L2R2).

**3.** On veut que R2=R0 et L2=L0. En remplacent dans les formules ci-dessus on obtient :

R0= f2(f1(R0)⊕L0)⊕R0

L0= f1(R0) ⊕L0

Donc : après les calcule, on trouve que le mot 10110000 (i.e., R0L0) est invariant par ce diagramme.