Sécurité Informatique

TD-1: Cryptographie et Cryptanalyse

Proposé par : Mr. Houcemeddine HERMASSI

(Corrigé du TD)

Partie-1 : Cryptologie classique

Exercice-1 : Attaques sur le cryptosystème de Cesar

On suppose l'association suivante :

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

q	r	s	t	u	v	w	x	y	z
16	17	18	19	20	21	22	23	24	25

Scénario KPA:
Ciphrtext = gozih

on connait la paire (plaintext,ciphertext)= (P,D)
Quel est la clé ?

Scénario CPA:
Tu peux choisir le plaintext que tu veux et obtenir le ciphertext correspondant.
Donner un exemple.
Quel est ton choix ? Et quel est donc la clé

Scénario CCA:
Tu peux choisir le ciphertext que tu veux et obtenir le plaintext correspondant.
Donner un exemple.
Quel est ton choix ? Et quel est donc la clé ?

KPA : k= (D – P) mod 26 = 3 – 15 mod 26= -12 mod 26 = 14
CPA : on choisit le plaintext « A » comme entrée à la machine de cryptage, on obtient comme ciphertext « D » -> donc la clé est (D-A) mod 26= D=3
CCA : on choisit comme ciphertext « A », comme entrée à la machine de décryptage, on obtient S. donc k=(A-S) mod 26 = -S mod 26 = 8

Exercice-2: Playfair

The Playfair Cipher

A	Z	I	WX	D
E	U	T	G	Y
O	N	K	Q	M
H	F	J	L	S
V	R	P	B	C

Utiliser la lettre spéciale Z pour séparer deux lettres égales. Et ajouter Z s'il reste une lettre individuelle à la fin.

On vous demande de Déchiffrer le ciphertext :

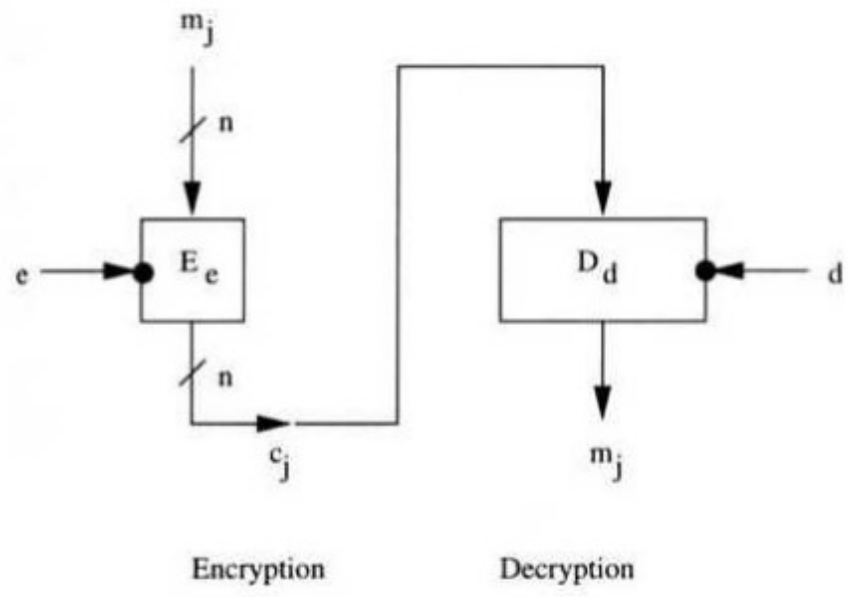
EJ DJ DJ EJ GA VO IE JY NK YV TI VO ZU

TH IS IS TH E W HE AT ST ON E C IP HE R

Partie-2 : Cryptologie moderne

Exercice-1 : ECB

On considère un cryptosystème de bloc qui applique une permutation à des vecteurs binaires de taille 4 en mode ECB.



La fonction de permutation $\boldsymbol{\pi}$ est définie comme suit :

$$b_4\, b_3\, b_2\, b_1 \rightarrow b_{\pi(1)}\, b_{\pi(2)}\, b_{\pi(3)}\, b_{\pi(4)}$$

On donne l'opération de permutation :

$$\boldsymbol{\pi} = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 3 & 4 & 1 \end{pmatrix}$$

Soit le plaintext m :

m = 101100010100101

1- Décrire mathématiquement ou avec un pseudo-code le fonctionnement de ECB

1) $c_i = E(m_i)$

2- Décomposer le plaintext en bloc de taille approprié. Faire du bourrage avec des zéros pour avoir des bloc de même taille

2) bourrage avec un seul 0 pour avoir des blocs de taille n=4. m = 1011 0001 0100 1010 m_1 = 1011 ; m_2= 0001 ; m_3 = 0100 ; m_4 = 1010

3- Appliquer le mode ECB lors du chiffrement des blocs du plaintext

3) Les blocks sont chiffrés séparément. On obtient $c_1 = E(m_1)$=0111 ; $c_2 = E(m_2)$=0010 ; $c_3 = E(m_3)$= 1000 ; $c_4 = E(m_4)$=0101

4- Donner le ciphertext final

4) D'où le ciphertext final est : C= 0111001010000101

5- Appliquer le déchiffrement et vérifier avec le message original

5) D(0111 0010 1000 0101) = 1011 0001 0100 1010.

6- Considérer un plaintext formé par les mêmes blocs 1010, cette redondance est-elle propagé dans le ciphertext ?

6) Oui la redondance sera propagé ds le ciphertext : E(1010 1010 1010 1010) = 0101 0101 0101 0101

7- Si l'ordre des blocs des ciphertexts est modifié ? le décryptage de chaque bloc est-il possible ?

7) l'ordre des blocs dans ECB n'affecte pas le chiffrement/déchiffrement des blocs

8- Que pensez-vous de la sécurité de ECB et dans quel application est-il approprié ?

8) ECB n'est pas sécurisé. Il est adéquat seulement pour le chiffrement des messages très courts comme les IV ou les clés dans d'autres modes.

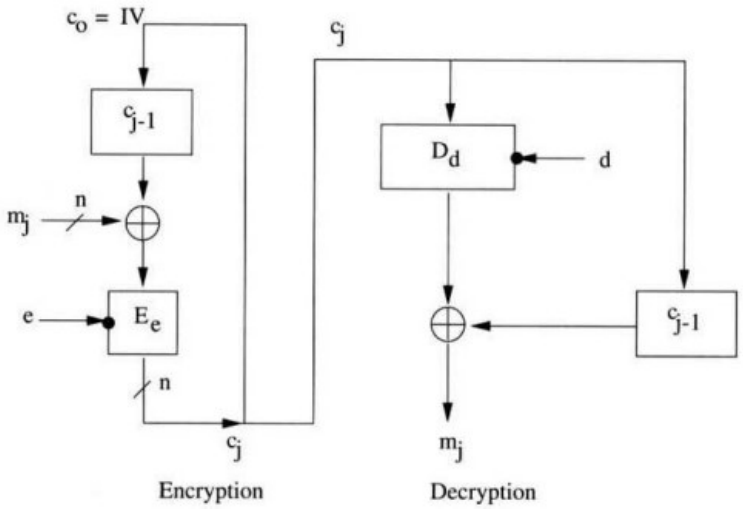
9- Le mode ECB est adéquat à la transmission des messages courts.

10- Openssl>enc –e –aes-128-ecb –in message.txt –out message.enc

11- Openssl>enc –d –aes-128-ecb –in message.enc –out message.dec

Exercice-2 : CBC

On utilise ma même clé, le même plaintext, la même opération mais en mode CBC. On donne IV=1010



1- Décrire mathématiquement ou avec un pseudo-code le fonctionnement de CBC.

1) Cryptage :

C_0= IV ; $c_j=E(c_{j-1} \oplus m_j)$ pour $1 \leq j \leq t$

Décryptage :

C_0= IV ; $m_j= c_{j-1} \oplus D(c_j)$ pour $1 \leq j \leq t$

2- Décomposer le plaintext en bloc de taille approprié. Faire du bourrage avec des zéros pour avoir des blocs de même taille

2) bourrage avec un seul 0 pour avoir des blocs de taille n=4. m = 1011 0001 0100 1010

m_1 = 1011 ; m_2= 0001 ; m_3 = 0100 ; m_4 = 1010

3- Appliquer le mode ECB lors du chiffrement des blocs du plaintext

3) c_0=1010 ;

$c_1=E(c_0 \oplus m_1)=E(0001)$ = 0010

$c_2=E(c_1 \oplus m_2)=E(0011)$ = 0110

$c_3=E(c_2 \oplus m_3)=E(0010)$ = 0100

$c_4=E(c_3 \oplus m_4)=E(1110)$ = 1101

4- Donner le ciphertext final

4) ciphertext final est :

c=0010 0110 0100 1101

5- Appliquer le déchiffrement et vérifier avec le message original

5) Décryptage :

$m_1=c_0 \oplus D(c_1)$ =1010 $\oplus$ D(0010)= 1010 $\oplus$ 0001 = 1011

$m_2=c_1 \oplus D(c_2)$ =0010 $\oplus$ D(0110)= 0010 $\oplus$ 0011 = 0001

$m_3=c_2 \oplus D(c_3)$ =0110 $\oplus$ D(0100)= 0110 $\oplus$ 0010 = 0100

$m_4=c_3 \oplus D(c_4)$ =0100 $\oplus$ D(1101)= 0100 $\oplus$ 1110 = 1010

m = 1011 0001 0100 1010

6- Considérer un plaintext formé par les mêmes blocs 1011, cette redondance est-elle propagé dans le ciphertext ?

6) La redondance ne sera pas propagée :

E(1011 1011 1011 1011) = 0010 0011 0001 0101

7- Si l'ordre des blocs des ciphertexts est modifié ? Le décryptage de chaque bloc est-il possible ?

7)si l'odre de blocs de ciphertext change ou les blocs de ciphertext sont remplacés par d'autres, alors le décryptage devient impossible. Ceci est un avantage de CBC par rapport ECB

8- Que pensez-vous de la sécurité de CBC et dans quelle application est-il approprié ?

8) sécurité améliorée parce que plus de confusion !

12- Si une erreur se passe dans le premier bloc du ciphertext. Etudier la propagation d'erreur sur le décryptage (dire quels sont les blocs affectés et les blocs intacts du plaintext).

9) la propagation d'erreur :

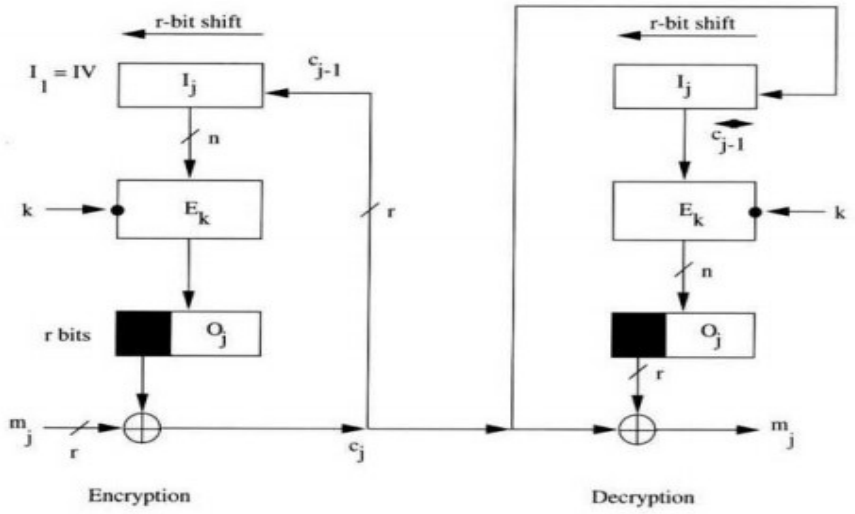
m_j estcalculé par c_j et c_{j-1}. pour cela si c_j a été reçu erroné, alors les plaintext m_j et m_{j+1} peuvent être erronés. Mais les blocs m_{j+2} et m_{j+3}, … ne seront pas influsencés, ils seront corrects.

10- Dire quelle application CBC est approprié

10) CBC peut être approprié pour le chiffrement de long messages.

Exercice-3 : CFB

Dans CFB, on a besoin d'un IV et aussi un entier r avec $1 \leq r \leq n$. le plaintext sera décomposé en blocs de r. Et Initialisation d'$\mathbf{I_1 = IV}$



Refaire l'exercice avec la même opération de permutation E, même plaintext et même **IV**, on donne aussi **r =3**.

1)
Pour $1 \leq j \leq u$, alice fait le suivant
- $O_j = E(I_j)$;
- Extraire t_j qui est les r premiers bits de O_j ;
- $c_j = m_j \oplus t_j$
- $I_{j+1} = (2^r I_j + c_j) \bmod 2^n$. donc I_{j+1} est généré en supprimant les premiers r bits de I_j et en ajoutant c_j.
- Le ciphertext sera donc $c = c_1 c_2 c_3 \ldots c_u$

Le décryptage est similaire :

Pour $1 \leq j \leq u$, bob fait le suivant
- $O_j = E(I_j)$;
- Extraire t_j qui est les r premiers bits de O_j ;
- $m_j = c_j \oplus t_j$
- $I_{j+1} = (2^r I_j + c_j) \bmod 2^n$. donc I_{j+1} est généré en supprimant les premiers r bits de I_j et en ajoutant c_j.
- Le plaintext sera donc $m = m_1 m_2 m_3 \ldots m_u$

2) $m_1 = 101$; $m_2 = 100$; $m_3 = 010$; $m_4 = 100$; $m_5 = 101$

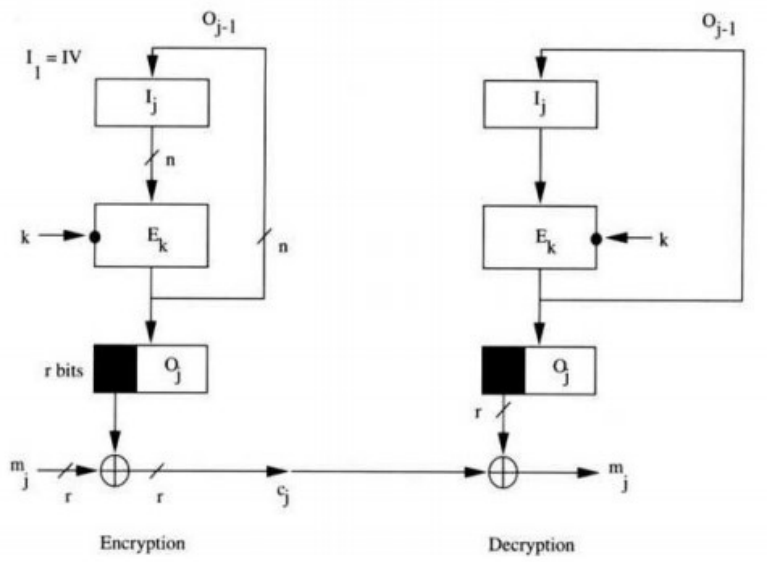
j	I_j	O_j	t_j	m_j	c_j
1	1010	0101	010	101	111
2	0111	1110	111	100	011
3	1011	0111	011	010	001
4	1001	0011	001	100	101
5	1101	1011	101	101	000

9) il ya propagation d'erreur aussi dans CFB puisque la partie erroné du ciphertext se trouve dans Ij. Il faut noter aussi que CFB ne peut être fonctionnel dans les cryptosystèmes à clé publique comme RSA puisque le récepteur utilise la clé publique aussi dans l'opération E comme l'émetteur.

10) Alice et Bob peuvent calculer t_{j+1} dès qu'ils connaissent le bloc c_j. Pour cela le bloc t_1 peut être calculé par Alice et Bob simultanément. Alice genere le ciphertext $c_1 = m_1 \oplus t_1$ et l'envoie à Bob. Le calcul de c1 sera rapide puisque il est fait par un simple XOR. Alors Alice et Bob peuvent simultanément calculer le bloc t_2, etc. CFB est donc approprié pour les longs messages mais il est plus rapide que CBC.

Exercice-4 : OFB

OFB est très similaire à CFB



Refaire l'exercice avec les mêmes Plaintext, clé, IV, r que CFB. Analyser la propagation d'erreur, la sécurité de OFB et sa rapidité.

1)
Pour $1 \leq j \leq u$, Alice fait le suivant
- $O_j=E(I_j)$;
- Extraire t_j qui est les r premiers bits de O_j ;
- $c_j=m_j \oplus t_j$
- $I_{j+1} = O_j$.
- Le ciphertext sera donc $c= c_1c_2c_3\ldots c_u$

Le décryptage est le même, seulement la troisième étape est remplacé par $m_j=c_j \oplus t_j$

2) Si un bit du ciphertext est reçu erroné alors le plaintext sera erroné exactement dans la même position. Donc il n ya pas de propagation d'erreur.

3) le bloc tj dépend seulement du vecteur d'initialisation I_1 et de la clé k. il peut donc être calculé par Alice et bob simultanément. Ceci est plus rapide que CFB. Mais le problème est que le chiffrement d'un bloc de plaintext dans OFB ne depend pas des blocs précédents mais seulement par sa position (ordre). Pour cela la manipulation du ciphertext par un intrus est beaucoup plus facile en OFB qu'en CFB.

m_1= 101 ; m_2=100 ; m_3= 010 ; m_4= 100 ; m_5=101

j	I_j	O_j	t_j	m_j	c_j
1	1010	0101	010	101	111
2	0101	1010	101	100	001
3	1010	0101	010	010	000
4	0101	1010	101	100	001
5	1010	0101	010	101	111

- Si la même clé est réutilisée pour le chiffrement de deux plaintexts, alors l'IV doit changer. Sinon, la même séquence tj est régénérée et à partir de deux ciphertexts $c_j=m_j \oplus t_j$ et $c'_j=m'_j \oplus t_j$, l'intrus peut constater que $c_j \oplus c'_j = m_j \oplus m'_j$. Et donc il peut déterminer m'_j s'il connaît m_j.

Exercice-5 : une ronde DES

Donner le résultat après une seule ronde de cryptage de DES sur le plaintext p :
P=0123456789ABCDEF
Avec la clé K :
K= 133457799BBCDFF1

Le codage binaire de P=0123456789ABCDEF

0	0	0	0	0	0	0	1
0	0	1	0	0	0	1	1
0	1	0	0	0	1	0	1
0	1	1	0	0	1	1	1
1	0	0	0	1	0	0	1
1	0	1	0	1	0	1	1
1	1	0	0	1	1	0	1
1	1	1	0	1	1	1	1

Apres IP on obtient

1	1	0	0	1	1	0	0
0	0	0	0	0	0	0	0
1	1	0	0	1	1	0	0
1	1	1	1	1	1	1	1
1	1	1	1	0	0	0	0
1	0	1	0	1	0	1	0
1	1	1	1	0	0	0	0
1	0	1	0	1	0	1	0

Donc

$$L_0 = 11001100000000001100110011111111,$$

$$R_0 = 11110000101010101111000010101010.$$

L'ecriture binaire de la clé K= 133457799BBCDFF1

0	0	0	1	0	0	1	1
0	0	1	1	0	1	0	0
0	1	0	1	0	1	1	1
0	1	1	1	1	0	0	1
1	0	0	1	1	0	1	1
1	0	1	1	1	1	0	0
1	1	0	1	1	1	1	1
1	1	1	1	0	0	0	1

On calcule la première clé intermediaire :

$$C_0 = 11110000110011001010101011111$$

$$D_0 = 01010101011001100111100001111$$

$$C_1 = 11100001100110010101010111111$$

$$D_1 = 10101010110011001111100011110$$

Et donc

$$K_1 = 000110110000001011101111111111000111000001110010$$

En utilisant cette clé on a

$$E(R_0) \oplus K_1 = 011000010001011110111010100001100110010100100111$$

$$f_{K_1}(R_0) = 00100011010010101010100110111011$$

Et finalement

$$R_1 = 11101111010010100110010101000100$$