CRYPTOGRAPHIE À CLÉ SECRÈTE

OLFA BESBES

olfa.besbes@isitc.u-sousse.tn

3ÈME LICENCE A.U. 2022-2023

1

Plan

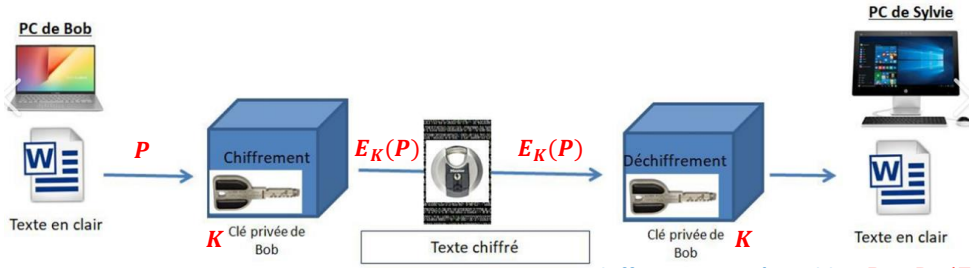
- Principe du chiffrement à clé secrète
- Cryptographie classique
- Chiffrement par bloc
- Système DES
- Système AES
- Cryptanalyse des systèmes à clé secrète
- Chiffrement par flot

2

Principe du chiffrement à clé secrète

❑ Le chiffrement et le déchiffrement se basent sur **une clé secrète partagée** entre les parties impliquées dans la communication.

❑ L'**expéditeur échange sa clé privée** avec le destinataire.



Le chiffrement est réversible : $P = D_K(E_K(P))$

3

Caractéristiques du chiffrement symétrique

⊕ Crypto-systèmes rapides en implantation matérielle.

⊕ Clés relativement courtes : 128 bits - 256 bits.

⊖ Gestion des clés difficiles (plusieurs clés)

⊖ Partage délicat à gérer de la clé secrète **!** ==> Confidentialité des messages vs. Confidentialités des clés secrètes.

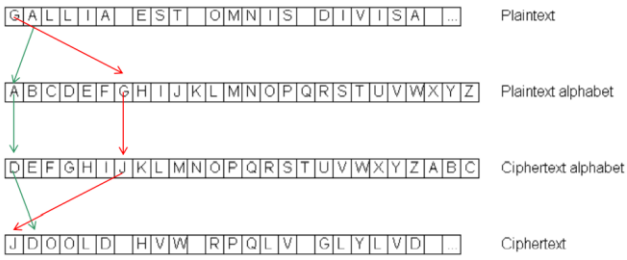
4

Cryptographie classique (1/3)

❑ **Transposition** : Changer l'ordre des mots selon un système sur lequel les parties impliquées se sont préalablement entendues.

- **Chiffrement de César** : K_e définie le décalage à droite des lettres de l'alphabet.
- 26 clés possibles => Très facile à casser !

$$K_e = 3$$



5

Cryptographie classique (2/3)

❑ **Substitution** : La clé secrète définit la permutation à appliquer pour chiffrer le message.

- **Chiffrement de Vigenère** : $chiffré = clair + clé$

clé ->	s	e	c	r	e	t	s	e	c	r	e	t
clair ->	d	a	s	i	s	t	g	e	h	e	i	m
chiffré ->	v	e	u	z	w	m	y	i	j	v	m	f

f = (t + m) mod 26
5 = (19 +12) mod 26

Matrice de chiffrement

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
l	f	g	i	w	y	d	x	e	b	s	n	t	c	q	u	h	z	v	r	m	k	j	a	o	p

Nombre de clés possibles =26! - 1

Clair

Clé secrète

Index	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
Plaintext	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
Shift 0	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
1	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a
2	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b
3	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c
4	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d
5	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e
6	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f
7	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g
8	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h
9	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i
10	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j
11	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k
12	m	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l
13	n	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m
14	o	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n
15	p	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
16	q	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p
17	r	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q
18	s	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r
19	t	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s
20	u	v	w	x	y	z	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t

6

Cryptographie classique (3/3)

- Chiffrement par transposition **===>** Nombre de clés possibles est faible (César).
- Chiffrement par substitution **===>** Analyse des fréquences des lettres dans le cryptogramme.

  ===> Cryptanalyse basée sur les statistiques (Vigenère).
- Faiblesse de ces systèmes classiques : **Taille** et **réutilisation** de la clé secrète.
- **Chiffrement de Verman** ou **Masque Jetable**: Chiffrement de Vigenère mais la taille de la clé est de même taille que le texte en clair. Ses caractères sont choisis d'une façon aléatoire. Elle ne doit pas être réutilisée.
- **Conclusion** : César et Vigenère sont ***peu sûrs*** ; Verman est *théoriquement incassable* mais ***peu pratique.***

==> Comment construire des crypto-systèmes à la fois sûrs et pratiques ?

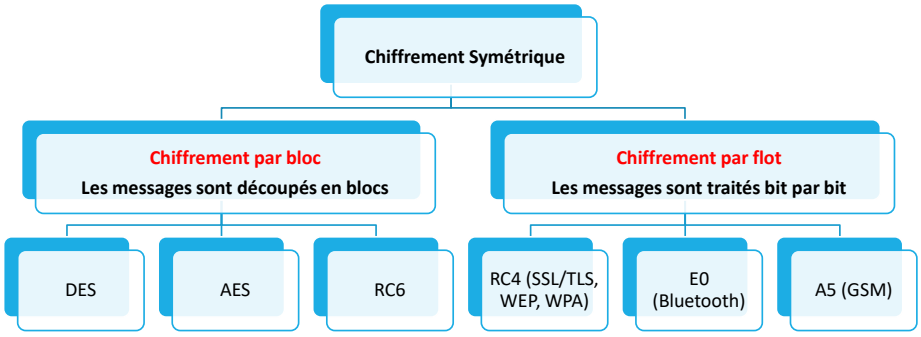
7

Principe de construction

- **Théorie de Shannon** : La combinaison de **confusion** et **diffusion** permet d'obtenir une sécurité convenable. Un bon algorithme de chiffrement doit satisfaire les deux propriétés :
  - **Confusion** : Masquer la relation entre message en clair et massage chiffré (pour éviter les attaques par analyses statistiques) ==> **Substitution**
  - **Diffusion** : Éparpiller la redondance du message (ex. deux lettres redondantes ne doivent pas être proche) ==> **Transposition**

8

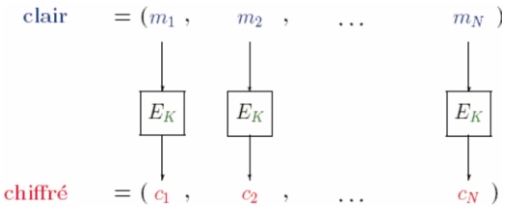
Chiffrement symétrique moderne



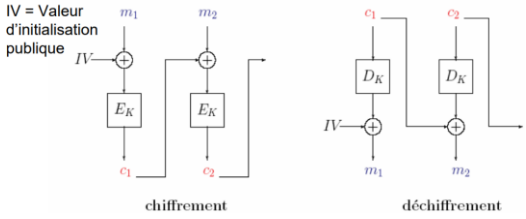
9

Modes de chiffrement par bloc

❑ **Mode ECB (Electronic Code Book) :**



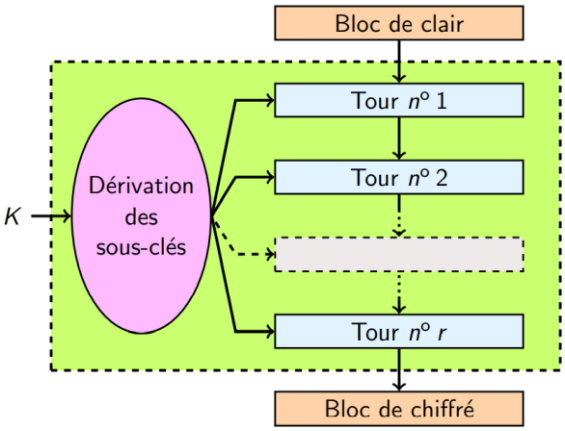
❑ **Mode CBC (Cipher-Block Chaining) :**



10

Construction d'un chiffrement par bloc

- **Principe** : Chaque texte clair est découpé en blocs de **même longueur** et chiffré **bloc par bloc**.
- Construction des **tours** dont chacun utilise une **sous-clé**.
- **Construction itérative** : Application itérée d'une transformation à chaque tour.
- Combinaison de **transformations** élémentaires : Ex. substitution, transposition, opérations linéaires et arithmétiques.
- Chaque transformation de tour dépend d'une sous-clé.
- Les sous-clés sont générées à partir d'une **clé maître**.



11

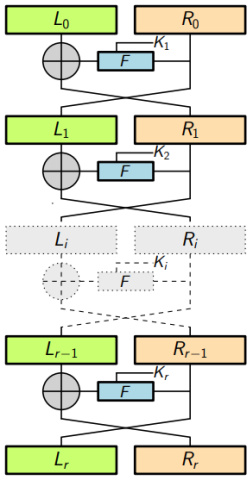
Schéma de Feistel

- Un **chiffrement itératif par blocs** transformant un message $P = (L_0, R_0)$ en un message chiffré $C = (L_r, R_r)$ par un procédé de $r > 1$ tours.
- Chaque tour transforme $(L_{i-1}, R_{i-1}) \rightarrow (L_i, R_i)$ en utilisant une sous-clé K_i et une **fonction de confusion** F:

$$\boldsymbol{L_i = R_{i-1}} \quad \text{et} \quad \boldsymbol{R_i = L_{i-1} \oplus F(R_{i-1}, K_i)}$$

- Le procédé est **réversible** :

$$\boldsymbol{R_{i-1} = L_i} \quad \textbf{et} \quad \boldsymbol{L_{i-1} = R_i \oplus F(L_i, K_i)}$$



12

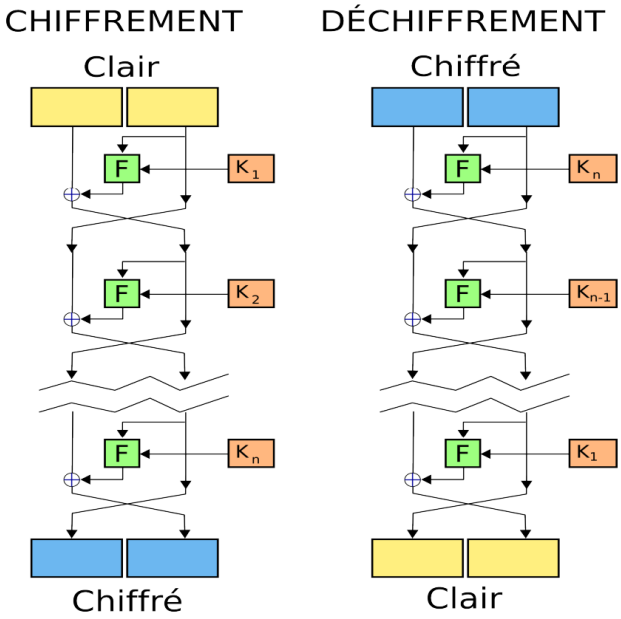


Schéma de Feistel

13

Exercice

Soit ce chiffrement de Fiestel à 3 tours.

Q1- Exprimer L_i et R_i en fonction de L_{i-1}, R_{i-1} et f_{ski} pour i=1,2,3.

Q2- Soit le message en clair $M = 1101111010$

$sk1 = 10101, sk2 = 11001$ et $sk3 = 10111$.

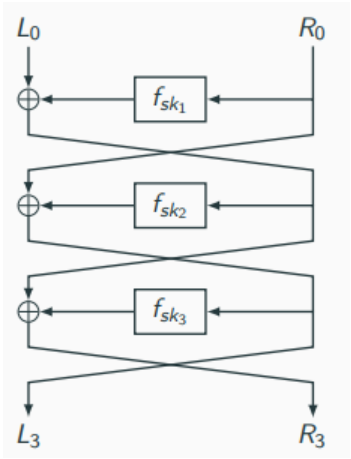
$f_{sk}(x) = \bar{x} \oplus sk$

Déterminer le chiffré de M.

Q3- Représenter le schéma de son déchiffrement.

Q4- Exprimer pour le déchiffrement L_{i-1} et R_{i-1} en fonction de L_i, R_i et f_{ski} pour i=1,2,3.

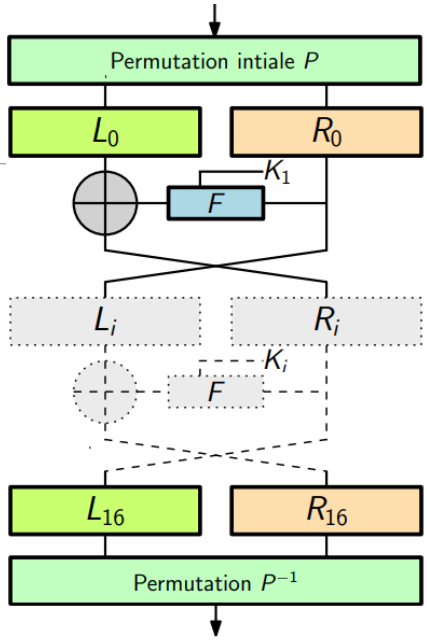
Q5- Déchiffré le message chiffré $C = 1100110011$



14

DES : Data Encryption Standard

- **Bloc - 64 bits** et **clé - 56 bits** ==> Bloc chiffré de 64 bits.
- DES est utilisé pour **chiffrement** et **authentification**.
- Basé sur le **schéma de Feistel** : 16 tours avec des sous-clés de 48 bits.
- Standardisé dès années 70, il est utilisé jusqu'à les années 2000.
- Il est à la base d'autres cryptosystèmes plus récents comme IDEA, FEAL, CAST, RC5, BLOW-FISH.



15

Permutation Initiale

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

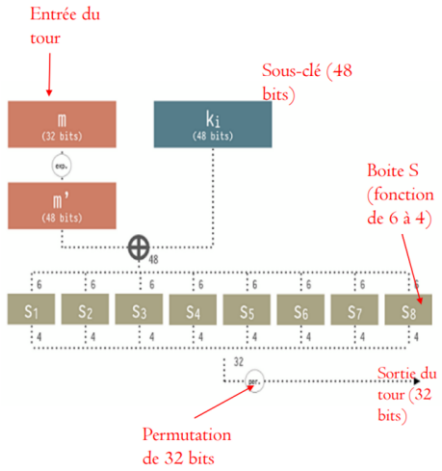
Le bit numéro 21 de la sortie
…
provient du bit numéro 30 de l'entrée

16

Fonction F du DES

- Dans DES, la fonction F: $32\ bits \rightarrow 32\ bits$ est constituée de :
  - Une **expansion** du message : $32\ bits \rightarrow 48\ bits$ (duplication de certain bits).
  - Un **XOR** de 48 bits avec la sous clé.
  - La **substitution** par concaténation de 8 sous-fonctions S_i : $6\ bits \rightarrow 4\ bits$ appelées **boîtes** pour assurer la **non-linéarité**.
  - Une **permutation** P aléatoire des 32 bits de sortie pour assurer la **diffusion**.

Input A	Input B	XOR Output
0	0	0
0	1	1
1	0	1
1	1	0



17

Expansion (32 → 48)

32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

Le bit numéro 15 de la sortie
…
provient du bit numéro 10 de l'entrée

Certains bits de l'entrée sont dupliqués (ex: bit 32)

18

S-Boîte : Table de substitution

❑ **Principe :** On divise les 6 bits en deux parties :

- Les deux bits aux extrémités indiquent la ligne.
- Les quatre bits centraux donnent la colonne correspondante.

❑ **Exemple** :

- Pour une entrée "011011", on divise en "0 1101 1". Ce qui donne pour la ligne "01" et pour la colonne "1101". La sortie de la table est alors "1001".

S_5		4 bits au centre de l'entrée															
		0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
Bits externes	00	0010	1100	0100	0001	0111	1010	1011	0110	1000	0101	0011	1111	1101	0000	1110	1001
	01	1110	1011	0010	1100	0100	0111	1101	0001	0101	0000	1111	1010	0011	1001	1000	0110
	10	0100	0010	0001	1011	1010	1101	0111	1000	1111	1001	1100	0101	0110	0011	0000	1110
	11	1011	1000	1100	0111	0001	1110	0010	1101	0110	1111	0000	1001	1010	0100	0101	0011

19

L'utilité des S-Boites

❑ DES sans les S-boites revient à une fonction XOR en plus des permutations linéaires.

❑ Ces fonctions linéaires peuvent être prédites, transformées, reversées d'une façon algorithmique.

===> Les **S-boites** permettent de **casser la linéarité** de la structure de chiffrement et **d'assurer la non-linéarité**.

20

Permutation (32 → 32)

16	7	20	21
29	12	28	17
1	15	23	26
5	18	31	10
2	8	24	14
32	27	3	9
19	13	30	6
22	11	4	25

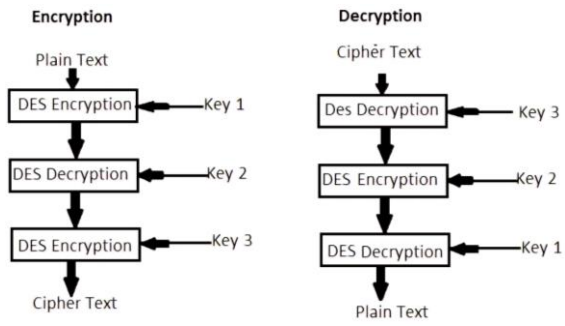
Le bit numéro 11 de la sortie
…
provient du bit numéro 23 de l'entrée

21

Triple DES

- **Limites du DES** :
  - **Taille des clés** : La recherche exhaustive des clés (2^{56}) devient possible.
  - **Taille de blocs** : 64 bits est devenu court et présente des risques d'attaques.

  ➔DES est cassé par **recherche exhaustive** (AES en 2000).
- **Solution** : Le Triple DES qui applique successivement 3 DES chacun avec une clé différente. ➔ Recherche exhaustive des $2^{56\times3}$ clés.
- **Limites du Triple DES** :
  - Taille de blocs de 64 bits.
  - 3 fois plus lent que le DES .



22