# Semestre : 6

# Unité d'enseignement :

## Cours 3

# Cryptographie

**Dr Mahmoud Hadef**

# Cryptographie

**Partie 1 :** Cryptographie symétrique

**L'Algorithme du DES**

**Partie 2:** Cryptographie asymétrique

**L'Algorithme du RSA**

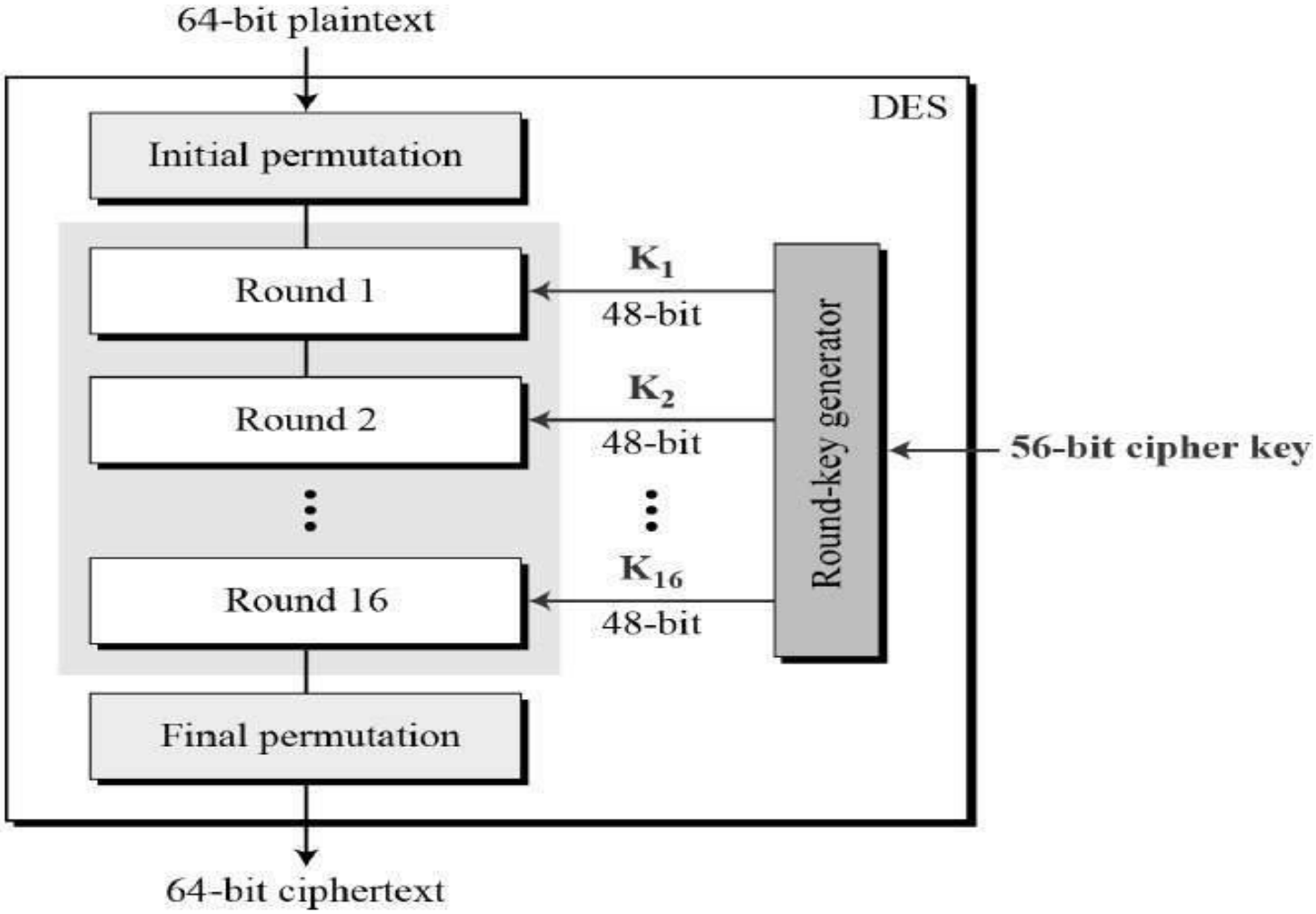
# L'Algorithme du DES
## Généralités

- DES chiffre des blocks de **64 bits** du texte normale en utilisant une clé qui est elle-même **64 bit** de longueur. On peut simplifier le DES en **trois** étapes majeurs *(nous décrivons la procédure pour 1 block):*
  1) Les 64 bits d'un block sont permutés.

  2) 16 tours d'opérations identiques sont appliqués sur la clé et sur le résultat de permutation de l' étape (1).

  3) L'inverse de la permutation est applique au résultat de l' étape (2).

# L'Algorithme du DES



# L'Algorithme du DES
# Etape 1 – Permutation

- La permutation initiale ne contribue pas a rendre le DES plus sécurisé. En réalité, cette étape était ajoutée pour convenir au hardware utilisé par IBM a l'époque pour implémenter le DES.
- La permutation suit un tableau statique qui est publiquement connu.
- Exemple:

| Input | 1 | 2 | 3 | 4 | 5 | … | 60 | 61 | 62 | 63 | 64 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Output | 40 | 8 | 48 | 16 | 56 | … | 9 | 49 | 17 | 57 | 25 |

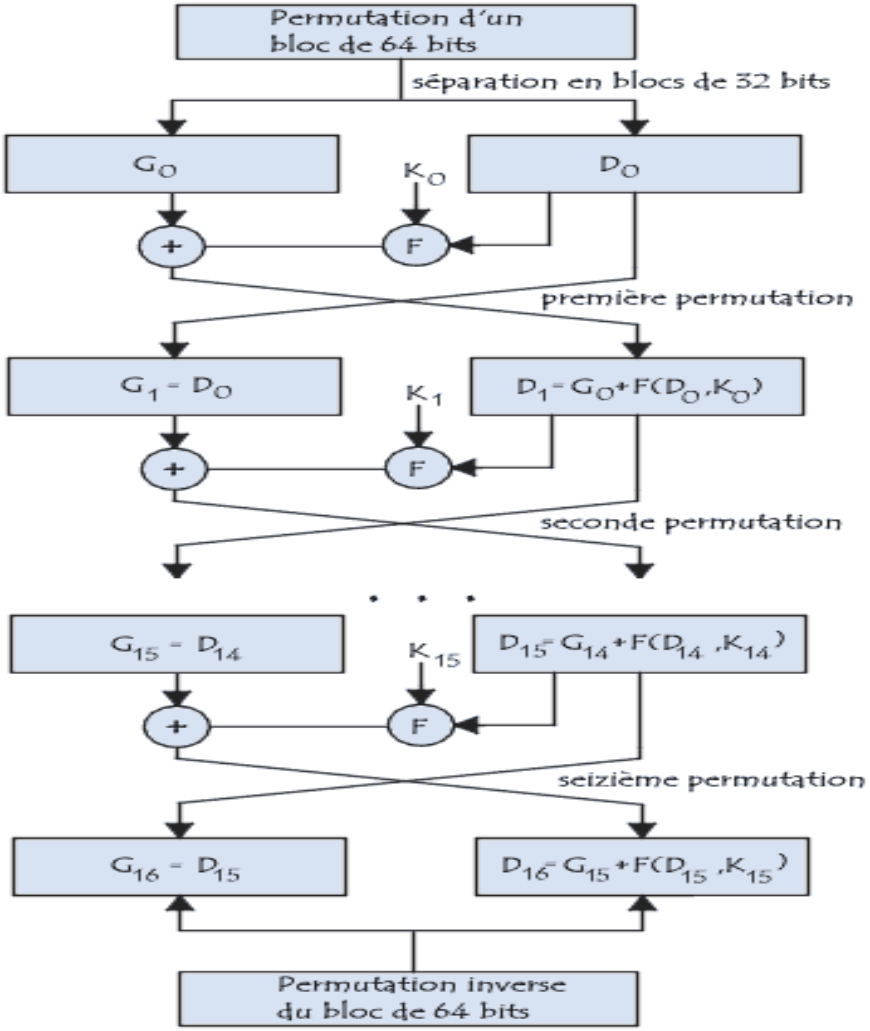
# L'Algorithme du DES
## Etape 2 – Les Operations d'un Tour

Dans chaque tour les opérations suivantes sont appliquées:

1. Le block de 64 bit (permuté) est divisé en deux blocks, chacun a 32 bits.
2. 48 bits sont sélectés a partir de la clé de 64 bits.
3. On dénote le demi-block gauche pendant le tour $i$ par $L_i$ et le demi-block droite par $R_i$ et les 48 bits selectes de la clé pendant le même tour par $K_i$

# L'Algorithme du DES

## Etape 2 – Les Operations d'un Tour



# L'Algorithme du DES

## Etape 2 – Les Operations d'un Tour

4. L'opération qui s'applique chaque tour est la suivante:

$$L_i = R_{i-1} \quad \ldots.(1)$$

$$R_i = L_{i-1} \oplus F\,(R_{i-1}, K_i) \ldots (2)$$

F: La Fonction de combinateur (combiner function)

$\oplus$ : est l'operateur du **XOR** (Exclusive OR)

| Table de vérité de XOR (OU exclusif) | | |
|---|---|---|
| **a** | **b** | **a XOR b** |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

# L'Algorithme du DES

## Etape 2A – Comment Sélecter $K_i$?

1) En faite, la clé contient **56** bits, car le dernier bit de chaque des 8 bytes est en faite un bit de parité.
2) Les 56 bits sont permutés d'une manière statique en fonction du tableau suivant:

| Input | 1 | 2 | 3 | 4 | 5 | … | 59 | 60 | 61 | 62 | 63 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Output | 8 | 16 | 24 | 56 | 52 | … | 17 | 25 | 45 | 37 | 29 |

3) Les 56 bits de la clé sont divisés en deux blocks de 28 bits. Chaque demi-block est indépendamment décalé a gauche par ou bien 1 ou 2 bits en fonction du numéro du tour.

# L'Algorithme du DES
## Etape 2A – Comment Sélecter $K_i$ ?

| Tour | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Décalage | 1 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 1 | 2 | 2 | 2 | 2 | 2 | 2 | 1 |

4) Les 56 bits qui résultent de ce décalage sont utilisés comme input a l'équation (2) et aussi comme la nouvelle clé pour le prochain tour.
5) Les 48 bits sont choisis a partir du 56 bits (résultat de décalage). Notez que les 48 bits sont choisis et permutés en même temps comme par les tableaux suivants:

# L'Algorithme du DES

## Etape 2A – Comment Sélecter $K_i$?

| Input | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Output | 5 | 24 | 7 | 16 | 6 | 10 | 20 | 18 | 12 | 3 | 15 | 23 | 1 | 9 | 19 | 2 |

| Input | 19 | 20 | 21 | 23 | 24 | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 36 | 37 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Output | 14 | 22 | 11 | 13 | 4 | 17 | 21 | 8 | 47 | 31 | 27 | 48 | 35 | 41 | 46 | 28 |

| Input | 39 | 40 | 41 | 42 | 44 | 45 | 46 | 47 | 48 | 49 | 50 | 51 | 52 | 53 | 55 | 56 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Output | 39 | 32 | 25 | 44 | 37 | 34 | 43 | 29 | 36 | 38 | 45 | 33 | 26 | 42 | 30 | 40 |

# L'Algorithme du DES
# Etape 2B – Que fait la fonction F?

**Rappel:**

$$L_i = R_{i-1} \quad \ldots.(1)$$
$$R_i = L_{i-1} \oplus F\,(R_{i-1}, K_i) \ldots (2)$$

Pour simplifier la notation, nous dénotons $R_{i-1}$ et $K_i$ par simplement $R$ et $K$.

1) La fonction **F** étend $R$ de 32 bits a 48 bits: Initialement, elle divise les 32 bits en huit mini-blocks de 4 bits chacun, elle étend chaque mini-block en un block de 6 bits et cela en copiant les deux bits adjacents des deux blocks voisins. *Exemple*: [0110] [1111] [1010]. Le block au milieu devient [0111111]

# L'Algorithme du DES
## Etape 2B – Que fait la fonction F?

2) La fonction **F** divise le 48-bits $K$ en huit blocks de 6 bits chacun, et chaque 6-bits blocks de clé est XORe avec le block de 6-bits de données.

3) Chaque block de 6-bits résultant de l'étape précédente est passe a une fonction substiturice $S\ (S\ Box)$ . Cette fonction réduit un block de 6 bits en un block de 4 bits.

4) En faite chacun des huit 6-bits blocks passe une fonction $S$ differente qui réduit chaque block de 6 bits en un block de 4 bits. Et donc il y'a huit blocks $S$ differents.

# L'Algorithme du DES
# Etape 3 – Permutation Finale

La dernière étape est le résultat final est permuté inversement au première étape.

**Quelques Notes Importantes:**

1) L'algorithme du DES est utilisé pour le chiffrement et le déchiffrement, c'est l'un des avantages de DES. La seule différence entre les processus est que les clés sont utilisées dans l'ordre inverse: $K_{16}, K_{15}, \ldots . K_1$

2) Pour chiffrer un block de donnée plus long que 64 bits, la technique du **Cipher Block Chaining (CBC)** est utilisé: le chiffrement du block j est XORe avec le texte du block $j + 1$ avant que le block$j + 1$est chiffré.

# L'Algorithme du RSA
## Généralités

- Le RSA est un algorithme de cryptographie asymétrique: il utilise une clé de chiffrement publique et une clé de déchiffrement privée.

- Un aspect essentiel de RSA est la façon avec laquelle les clés sont choisies. La clé de RSA consiste en 1024 bits, donc elle est beaucoup plus couteuse que le DES.

- Le chiffrement et le déchiffrement sont exprimés en une fonction, qui nécessite une puissance de calcule énorme.

# L'Algorithme du RSA
## Les Etapes

- Génération de la clé publique et privée: A cette fin, il faut choisir deux nombres primaires très larges $p$ *and* $q$. Puis les multiplier pour obtenir le nombre *n.*

- Les deux nombres $p$ *et* $q$ *doivent être 256 bits de longueur (approximativement).*

- Ensuite, il faut choisir la clé du chiffrement $e$ de façon a ce que $e$ et $(\mathrm{p}-1)*(\mathrm{q}-1)$ sont relativement primaires (c'est-à-dire ils n'ont aucun dénominateur commun supérieur a 1)

# L'Algorithme du RSA
## Les Etapes

- Finalement, calculer la clé du déchiffrement $d$ de façon a ce que:

$$\boldsymbol{d = e^{-1} \, mod \, ((p-1) * (q-1))}$$

- La clé publique doit être construite a partir du pair $< e, n >$ et la clé privée est donnée par la pair $< d, n >$

- Les nombres primaires originaux $p$ *et* $q$ ne seront désormais pas requis, nous pouvons en débarrasser mais nous ne devons pas les divulguer.

# L'Algorithme du RSA
## Les Etapes

- Etant donné les deux clés, le chiffrement est défini par la formule suivante:

$$c = m^e \bmod n$$

- Et le déchiffrement est défini par:

$$c = c^d \bmod n$$

- Ou $m$ est le message ( a chiffrer) et $c$ est le message chiffré résultant. Notez que $m$ doit être inferieur a $n$ , et ceci implique que le message ne doit être pas plus que 1024 bits.

- Et donc un long message doit être divise en des blocks de 1024 bits.

# L'Algorithme du RSA
## Exemple

- Nous allons démontrer le principe du RSA avec des petites valeurs de $p$ *et* $q$.
- Supposez, nous choisissons $p$ = 7 *et* $q = 11$, donc

$$\boldsymbol{n = 7 * 11 = 77}$$

$$\boldsymbol{(p-1)*(q-1) = 60}$$

- Nous devons choisir une valeur de $e$ qui est relativement primaire avec 60. On selecte $e$ =7.
- Nous calculons $d$ de façon a ce que:

$$\boldsymbol{d = 7^{-1}\, mod((7-1)*(11-1))}$$

$$\boldsymbol{7{*}d = 1\ mod\ 60}$$

$$\boldsymbol{d = 43}$$

$$\boldsymbol{7 * 43 = 301 = 1\ mod\ 60}$$

# L'Algorithme du RSA
# Exemple (suite)

- Maintenant nous avons la clé publique $< e, n >$= $< 7, 77 >$ et la clé privée $< d, n >$= $< 43, 77 >$.

- ***Note:*** que dans cet exemple, il est facile de deviner $p$ *et* $q$ une fois qu'on connaisse $n$, et donc deviner $e$ a partir de $d$. Mais si $n$ était le produit de deux numéros primaires de 256 bits chacun, ca aurait été infaisable de trouver les deux numéros primaires $p$ *et* $q$.

- Donc, la raison pour laquelle nous ne devons pas divulguer $p$ *et* $q$ devient évidente.

# L'Algorithme du RSA
## Exemple (suite)

- Considérons une opération de chiffrement simple. Supposez que nous voudrions chiffrer un message qui contient la valeur 9. En suivant la méthode de chiffrement décrite dans notre cours:

$$\mathbf{c} = m^e \, mod \, n$$

$$\mathbf{c} = 9^7 \, mod \, 77$$

$$\mathbf{c} = 37$$

- Et donc le message chiffre qu'on doit envoyer est 37. A la réception du message, le message chiffré doit être décrypté:

$$\mathbf{m} = 37^{43} \, mod \, 77 = 9$$

# L'Algorithme du RSA
## Conclusions

- Donc l'aspect de sécurité du RSA découle du faite que factoriser des numéros très larges est une opération mathématique très couteuse.

- La vitesse avec laquelle des larges numéros peuvent être factorises dépend largement du vitesse du processeur utilisé et l'algorithme utilise pour la factorisation. Il est estime que les numéros de 512 bit seront facilement factorise dans les prochaines années a venir.

# L'Algorithme du RSA
# A-t-on cassé le RSA?

- En 1977, un défi a été publiquement publié consistant a déchiffrer un message de 129 bits (encrypté avec RSA). On croyait a l'époque que le code nécessitait 40 quadrillions d'années de computations pour le déchiffrer.

- En April 1994, seulement 17 ans après, 4 chercheurs ont pu déchiffrer le message caché qui était *«The Magic words are squeamish ossifrage»*