Introduction à la cryptographie (cours 3): Chiffrement par bloc (DES)

Université Paris 13 Villetaneuse

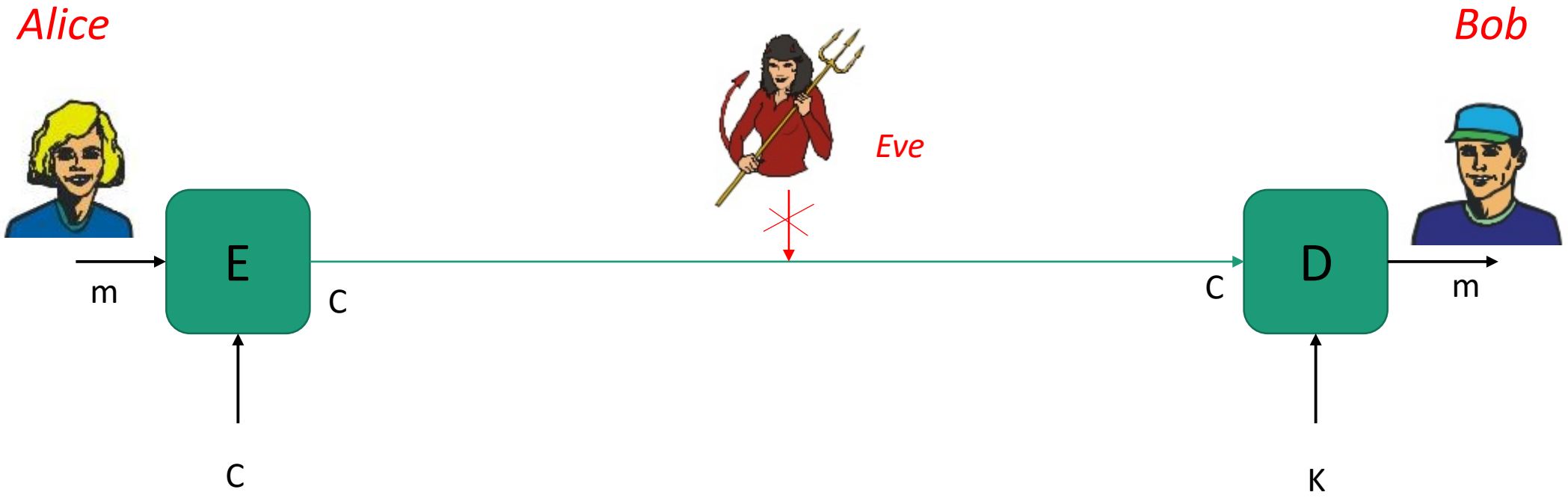
01/02/2016

Houda FERRADI

Plan

- Rappel du chiffrement symétrique
- Introduction chiffrement par blocs
- Chiffrement par blocs vs chiffrement par flots
- Construction du DES
- Attaques sur le DES
- Alternatives du DES

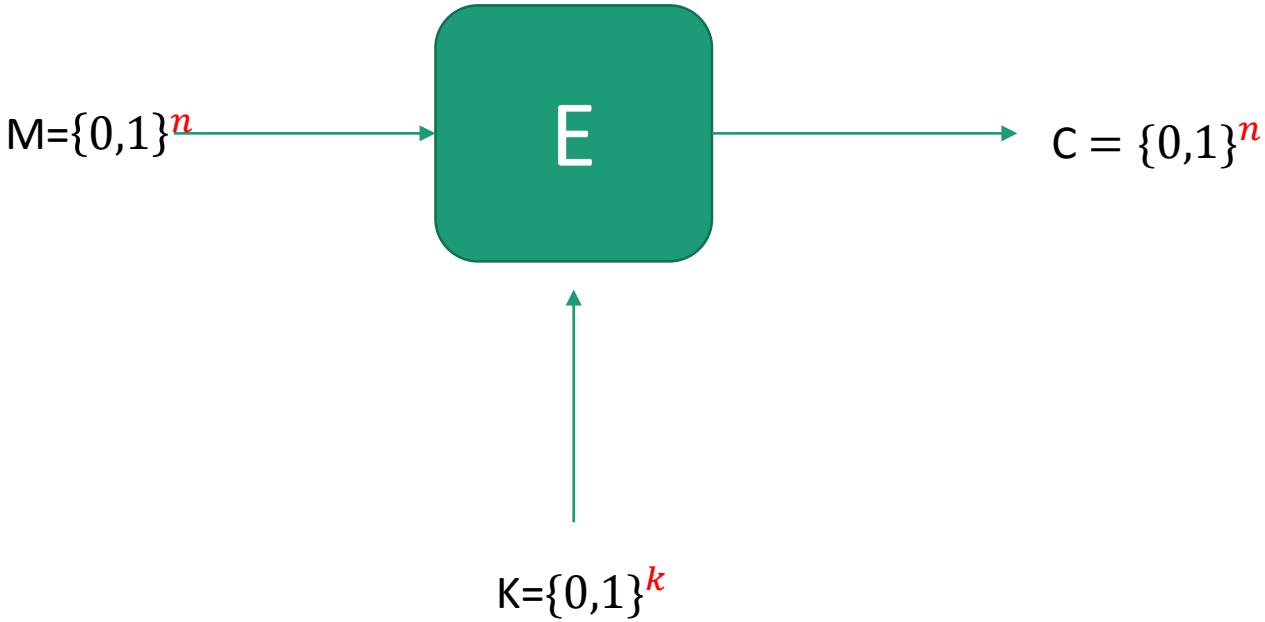
Rappel : chiffrement symétrique ou à clé secrète



E (Fonction de chiffrement) et D (Fonction de déchiffrement): Fonctions inversibles et efficaces
K: Clé secrète ou symétrique
C: Le message chiffré
m, k, et c sont de taille déterminée!

Rappel : chiffrement symétrique

Définition : Un algorithme de chiffrement symétrique transforme *un message en clair M* avec *une clé secrète K*. Le résultat est *un message chiffré C*



Deux grandes catégories

Chiffrement par blocs

- M est traité par blocs de données (ex: 64 bits ou 128 bits)

Exemple d'algorithmes : DES, AES, IDEA, RC6, BLOWFISH, …

Chiffrement par flots

- M est traité bit par bit (cours précédent)

Exemple d'algorithmes: RC4, Bluetooth E0/1, GSM A5/1,

Introduction: Chiffrement par bloc

- Les exemples historiques de chiffrement (par transposition et par substitution) vus en début de cours sont des *chiffrements par blocs*.
- La *substitution* ajoute de la *confusion* au procédé de chiffrement et la *transposition* ajoute de la *diffusion* en éparpillant l'influence moyenne (selon les différentes clés) de chaque bit du clair, sur les bits du chiffré.
- Mais aucun de ces deux procédés ne produit à la fois de la *confusion* et de la *diffusion* ➔ pas une réelle sécurité.

Les systèmes modernes, pour assurer une véritable sécurité, doivent produire à la fois de la confusion et de la diffusion, faute de quoi ils ne résistent pas aux attaques que nous décrirons plus loin.

Introduction: Chiffrement par bloc

- Une des primitives (« briques ») *les plus largement utilisées* en cryptographie symétrique mais aussi dans les fonctions de Hachage, générateur pseudo aléatoire etc…
- Dans un système par blocs, chaque texte clair est découpé en blocs de même longueur et chiffré bloc par bloc.
- La taille du texte clair est fixe => Plus de maitrise sur le propriétés du chiffrement
- Le système de chiffrement par blocs le plus utilisé jusqu'à l'an 2000 est le DES

=> Ce cours va être focalisé sur le fonctionnement de DES (le prochain sur AES)

Introduction: Chiffrement par bloc

- Dans un système par blocs, chaque texte clair est découpé en blocs de même longueur et chiffré bloc par bloc.
- La taille de bloc (n = 64 ou 128 bits) Les modes opératoires permettent généralement des attaques quand plus de $2^{n/2}$blocs sont chiffrés avec une même clé.
- La clé soit être suffisamment grande (k>128): Pour un bon algorithme, la meilleure attaque doit coûter 2^{k}opérations (la technique utilisée est l'attaque exhaustive).

Exemple:

- AES: n= 128 bits , k=128, 192, 256 bits
- 3 DES: n= 64 k= , 168 bits

Construction: Fonction aléatoire

Nouvelle définition dans le chiffrement par blocs ➔ Permutation aléatoire

Tel que: pour une fonction de chiffrement E(k,m) :

- Il existe une façon *efficace* d'évaluer cette permutation
- Il existe un algorithme d'*inversion* efficace D(k,c)
- E(k,m) doit être une fonction *bijective*

Chiffrement par permutation général

- Même principe que la permutation classique.

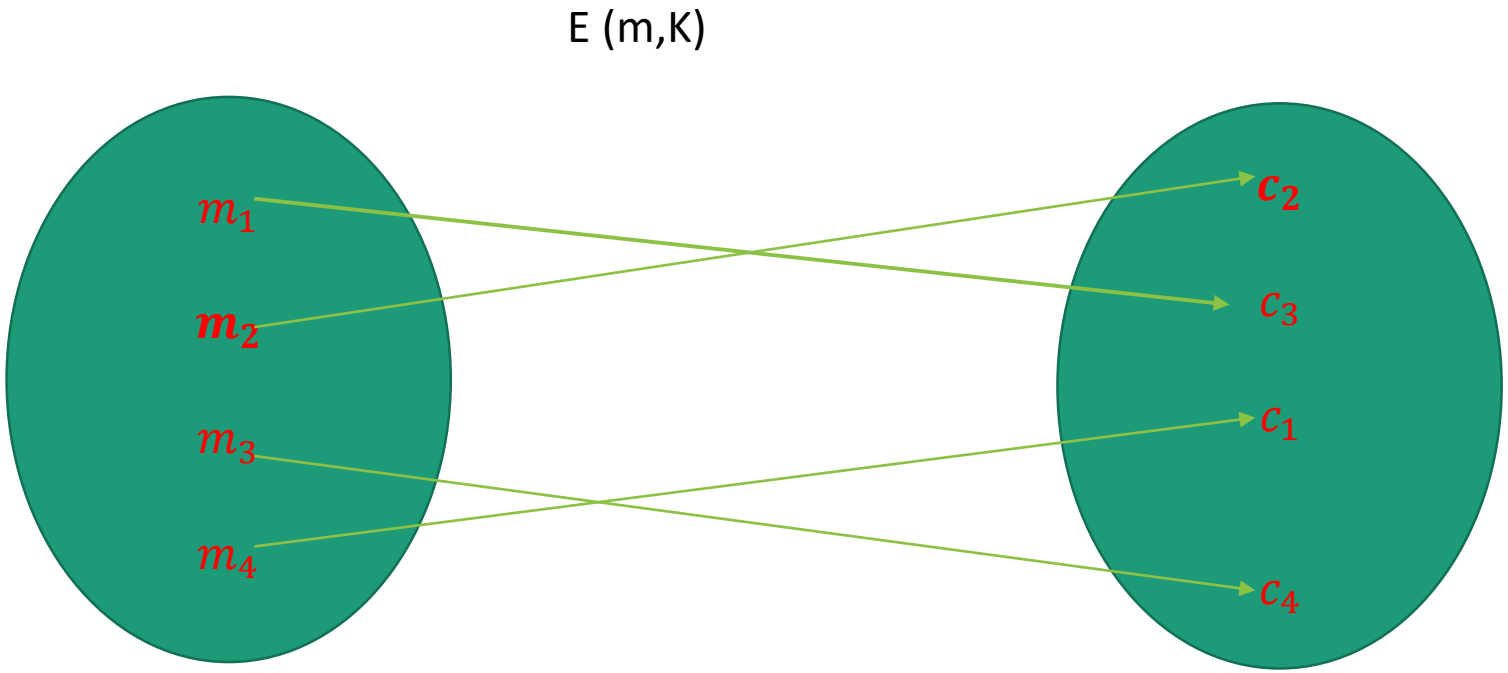
Pour un message M dont les lettres sont numéroté de I à n $M = m_1\ m_2\ m_3 \cdots m_n$. Mais ici les m_i sont des bits.

- *Et une permutation (bijection):* P : {I,...,k}⟶{I,...,k} (Slide suivant)

- *On découpe M en bloc de k bits:* M = $[m_1, m_2, m_3 \ldots m_n]$ où $M_i \in \{0,1\}^k$

Exemple de fonction bijective

- E (m, k) est fonction de chiffrement bijective



2 Principes fondamentaux pour le chiffrement par blocs

- La confusion vise à cacher n'importe quelle *structure algébrique* dans le système pour la rendre intelligible (table de substitution S-box).
- La diffusion doit permettre à chaque bit de *texte clair* d'avoir une influence sur une grande partie du texte chiffré. Ce qui signifie que la modification d'un bit du bloc d'entrée doit entraîner la modification de nombreux bits du bloc de sortie correspondant.
- Les 2 notions ont été introduites par Shannon.

La confusion est assuré par une substitution non-linéaire (S-Box)

La diffusion est assuré par une permutation linéaire

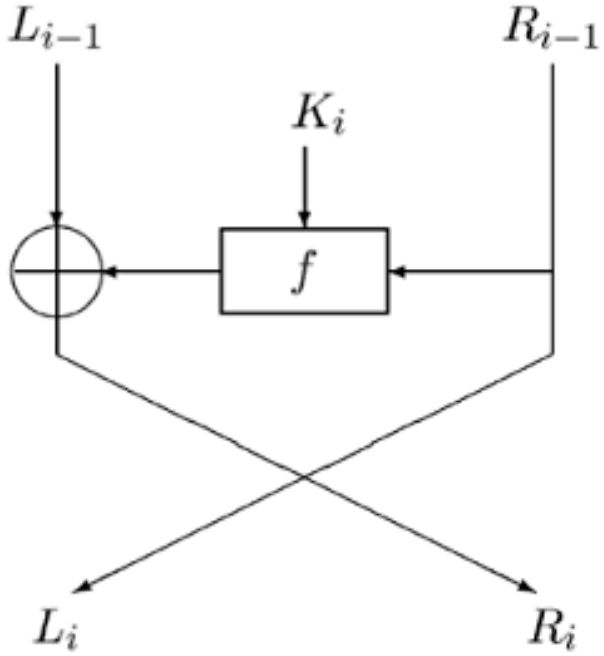
Réseaux Feistel

Méthode:

- *Chiffrement:* un bloc de texte en clair est *découpé en deux* ; la transformation de ronde est appliquée à une des deux moitiés, et le résultat est combiné avec l'autre moitié par *ou exclusif*. **Les deux moitiés sont alors inversées pour l'application de la ronde suivante.**

- *Déchiffrement:* est structurellement identique au chiffrement.

Remarque: La fonction de tour n'a pas besoin d'être inversible

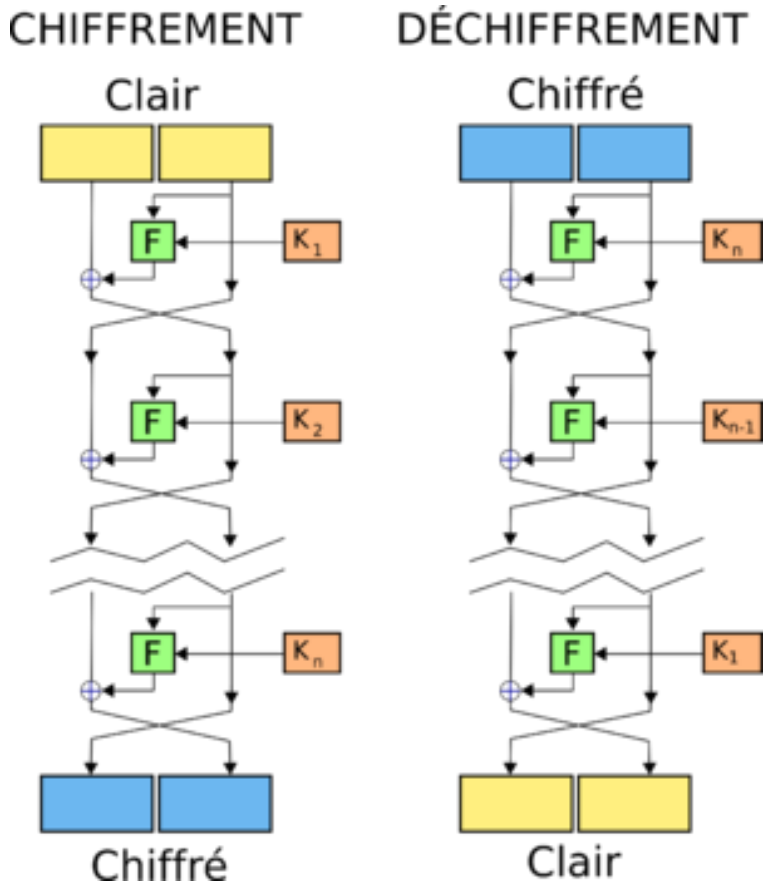


$$\begin{cases} L_i = R_{i-1} \\ R_{i-1} = L_{i-1} \oplus f(R_{i-1}, K_i) \\ avec\ K_i \text{ est dérivée de K} \end{cases}$$

Théorème de Sécurité des Réseaux Feistel:

Si une fonction aléatoire **sûre** est utilisée pour trois tours de Feistel avec trois clés **indépendantes**, on obtient alors une fonction pseudo aléatoire avec des permutations pseudo aléatoire (Luby-Rackoff, 1985).

Exemple: DES utilise 16 rondes (ou cycles) du réseau Feistel



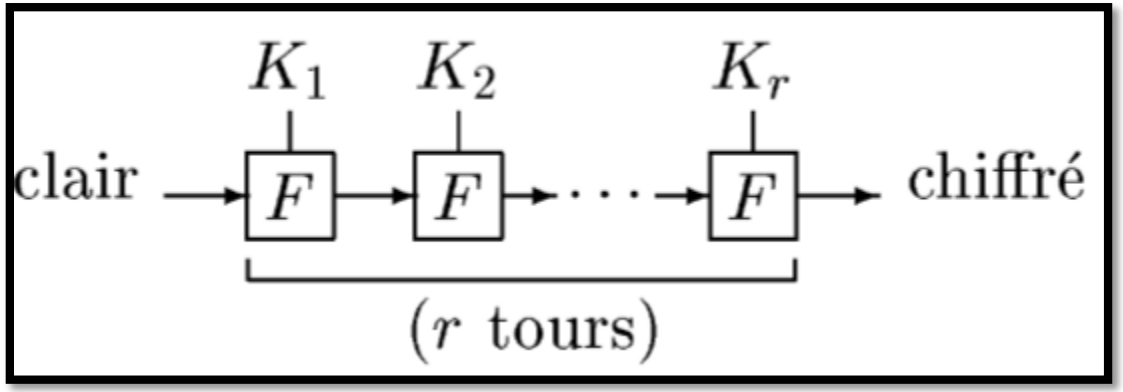
Construction du DES

Introduction : DES

- 1970: Chiffrement par bloc « Lucifer » développé par IBM : k=128 et m=128
- 1973: *DES (Data Encryption Standard)* Adopté comme standard US par le Bureau National des Standards Américains NBS (FIPS 46-2) pour le chiffrement par blocs
- 1976: Le DES, un variant de Lucifer, est adopté comme ce standard : Taille de bloc = 64 bits, k = 56 bits, m = 64
- 1997: DES cassé par la recherche exhaustive (AES en 2000.)

Construction : DES

- On utilise une combinaison de substitutions et de permutations pour le calcul générique de $F(k_i, m)$ sur 32 bits:
- Le texte clair et le texte chiffré sont des suites de bits de longueur l
- La substitution (appelée aussi "S-boîte") est notée S (non linéarité)
- La permutation P (fonction aléatoire)



Fonctionnement de F du DES

Pour le calcul générique de $F(k_i, m)$ sur 32 bits, 4 étapes:

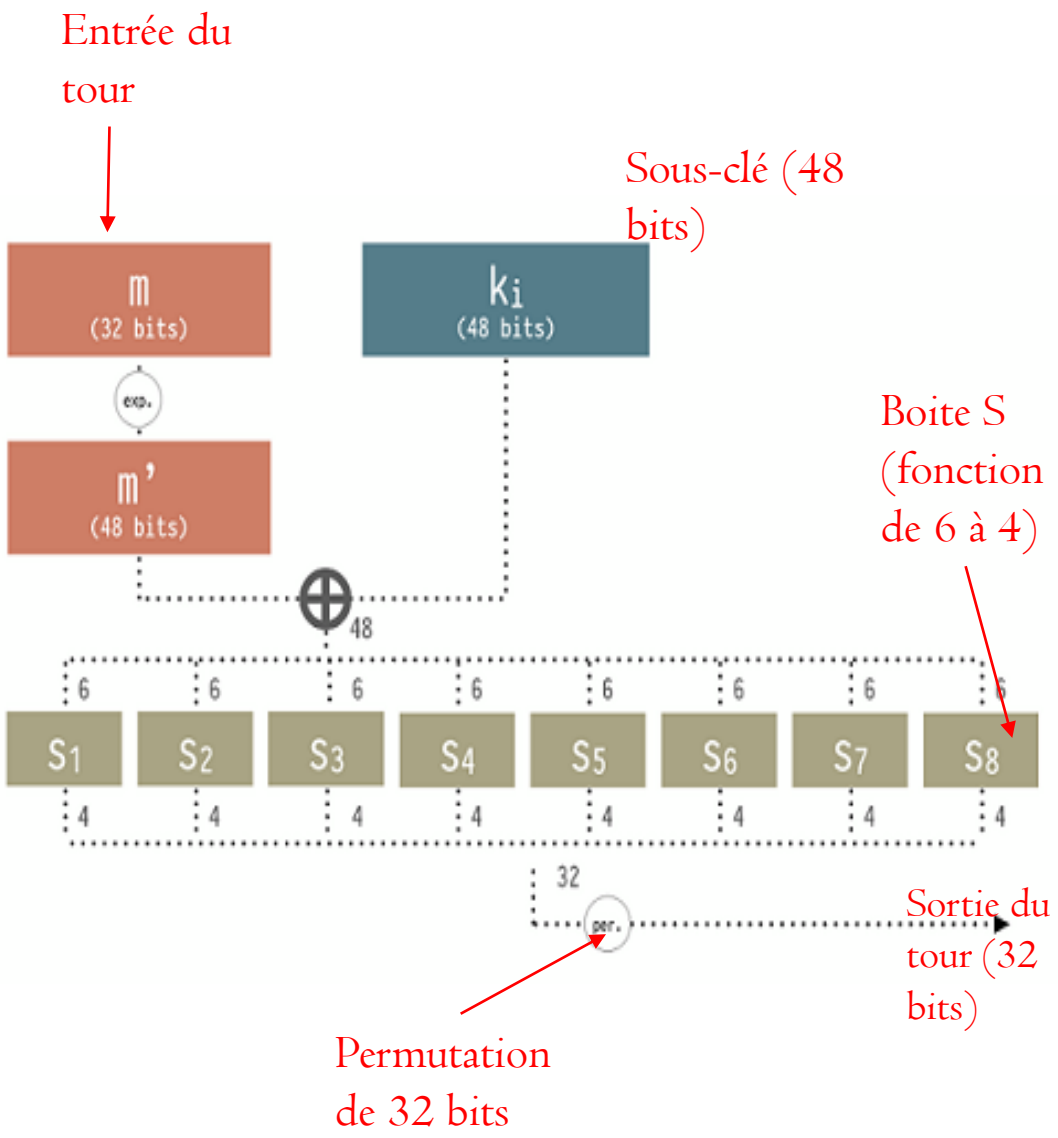
1. m est expansé en m' sur 48 Bits (pour obtenir la même taille que la clé k_i)

2. Application du XOR: S=E(R_{i-1}) $\oplus$ k_i= S_1 $S_2S_3\ldots$ S_8 sur 48 bits donc chaque S_i sur 6 bits

3. La substitution $C_i = \boldsymbol{S}(S_i)$ sur 4 bits.

Pour tout I$\leq i \leq 8$ $avec$ $S_i$est la I ème "S_box"

4. La permutation (diffusion) f(R_{i_1}, k_i) = $\boldsymbol{P}(C_1C_2 \ldots C_8)$



Algorithme DES : Table de substitution (S-box)

La sortie de 4 bits est obtenue à partir de l'entrée de 6 bits.

Méthode:

On divise ces 6 bits en deux parties : les deux bits aux extrémités et les quatre bits restants (au centre). Les deux bits indiquent la ligne et les bits centraux donnent la colonne correspondante.

Exemple: avec 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

Pourquoi « S-box » ?

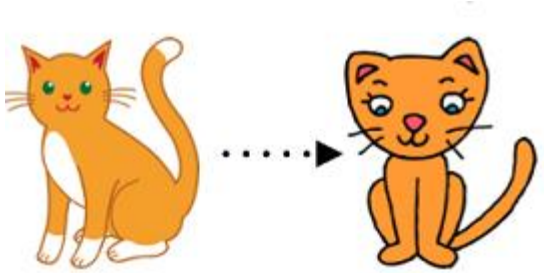
- Un DES sans S-Box est simplement une fonction XOR + permutations linéaires ➔ fonction linéaires.

- Les *fonctions linéaires* peuvent être *prédites, reversées et transformées* d'une façon algorithmique.

➔ Les S-Boxes permettent donc de casser la linéarité de la structure de chiffrement

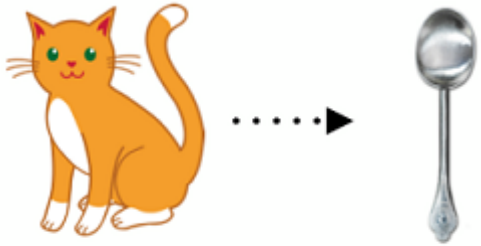
Critère d'un bon S-Box

Aucune transformation doit être "proche" à une transformation linéaire ou affine: pas de relation statistique entre le texte et son cryptage



Mauvais S-Box

Transformation avec une relation Linéaire



Bon S-Box

Transformation sans relation linéaire

Historique des Attaques contre DES

- En 1975: DES a été choisi comme norme au Etats-Unis est devenu le système cryptographique le plus utilisé dans le monde.

- 1997: Le DES commence à être critiqué à cause de la taille trop faible de sa taille de clés $k = 56$

- En 1998, le défi « DES Challenge » a été lancé pour casser DES: la machine "Deep Crack" (spécialement conçu pour attaquer DES) a réussi en quelques jours à retrouver la clé par une *attaque exhaustive*.

Attaque exhaustive: ce que l'ordinateur sait faire de mieux en traitant très rapidement une quantité gigantesque de données (notions introduites lors de premier cours)

Problèmes du DES

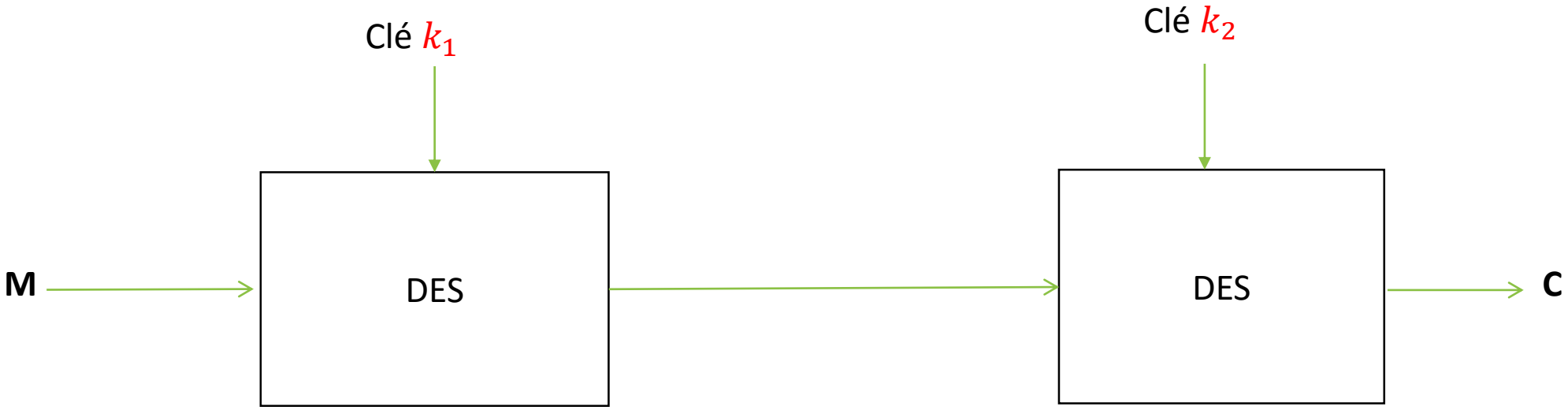
- Etant taille de clé est très petite car une recherche exhaustive en 2^{56} est très « faisable » par un ordinateur → Utilisation du Triple-DES

- Taille du bloc (attaques avec 2^{32} messages)

- Cryptanalyse linéaire et différentielle

Chiffrement multiple

- Pourquoi pas un chiffrement double (2 DES) ? $E(x) = E(k_1, E(k_2, x))$



Attaque par le milieu (chiffrement double)

- Attaque naïve: Essayer toutes les clés du chiffrement double e.g. 2^{112} clés possibles

- Attaque par le milieu: Attaque plus judicieuse qui consiste à trouver un compromis entre temps-mémoire, e.g:

➢ Operations à effectuer: 2^{56}

➢ stockage couples (clair, chiffré) en mémoire: 2^{56} couples

Attaque par le milieu (chiffrement double)

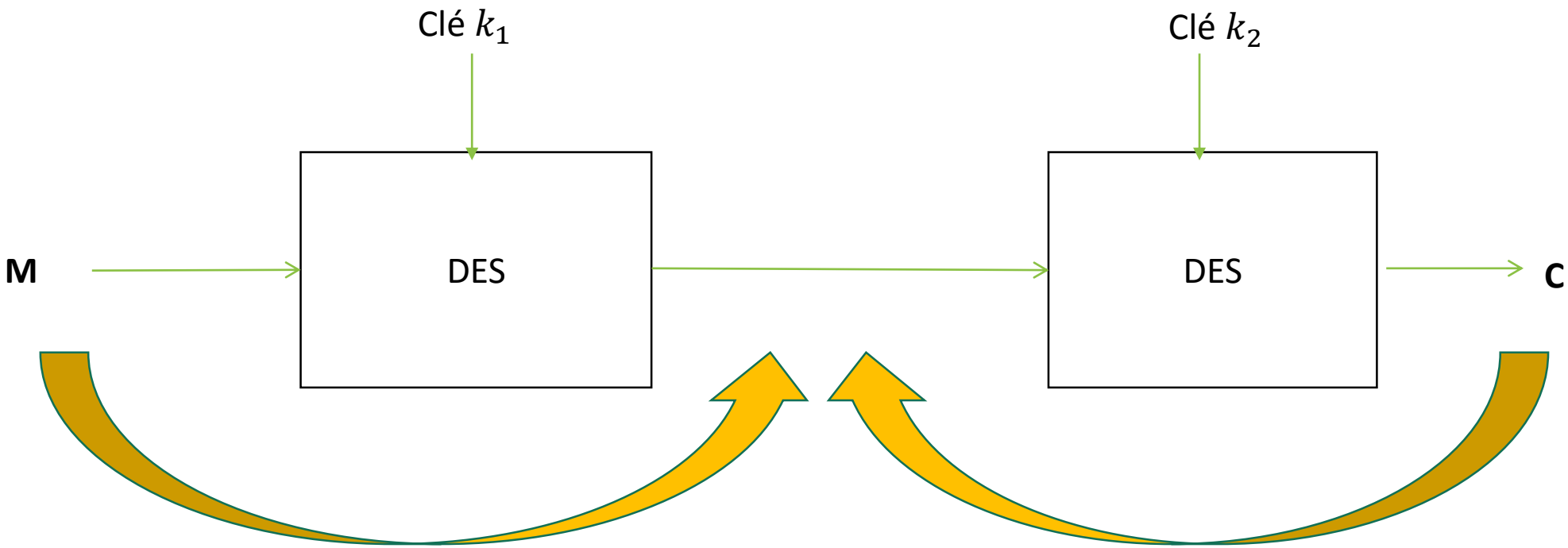
- Étant donné un couple clair-chiffré (M,C):

– calculer $N_i = DES_i(M)$ pour $0 \leq i < 256$ (i.e. pour chacune des 256 valeurs possibles de k_1)

– calculer $P_j = DES^{-1}(j,C)$ pour $0 \leq j < 256$ (i.e. pour chacune des 256 valeurs possibles de k_2)

– On cherche les indices (i,j) tels que $P_j = N_i$

Attaque par le milieu (chiffrement double)



2^{56} *calculs*
$N_i = DES_i(M)$

Pour chaque $P_j = DES^{-1}$(j,C)
Trouver $P_j = N_i$

Problème du 2 DES

Attaque par le milieu :

- Attaque en 2^{56} en temps et 2^{56} couples (chiffré, clé) en mémoire

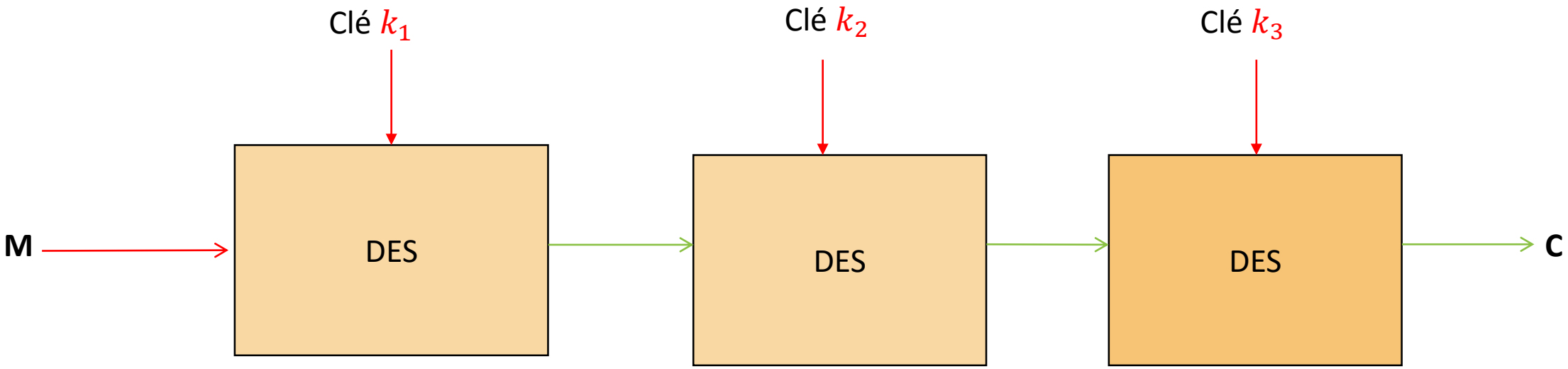
Taille de clés très courte:

- **Par conséquent :** la sécurité du double DES n'atteint pas 2^{112} (la sécurité minimal pour se protéger contre les attaques exhaustives) mais seulement 2^{56}, comme le DES

➔ Triple DES

Chiffrement triple

- Chiffrement triple (3 DES) avec 3 clés indépendantes : $E(x) = E(k_1, D(k_2, E(k_3, x)))$



Problème du 3DES

Avantage:

La construction 3DES évite *l'attaque par le milieu* du 2DES et résout le problème de *la taille de clé* dans DES et 2DES

Inconvénient:

- Le problème de la taille du bloc subsiste
- Le Triple-DES est lent à calculer

➔ Migration vers AES