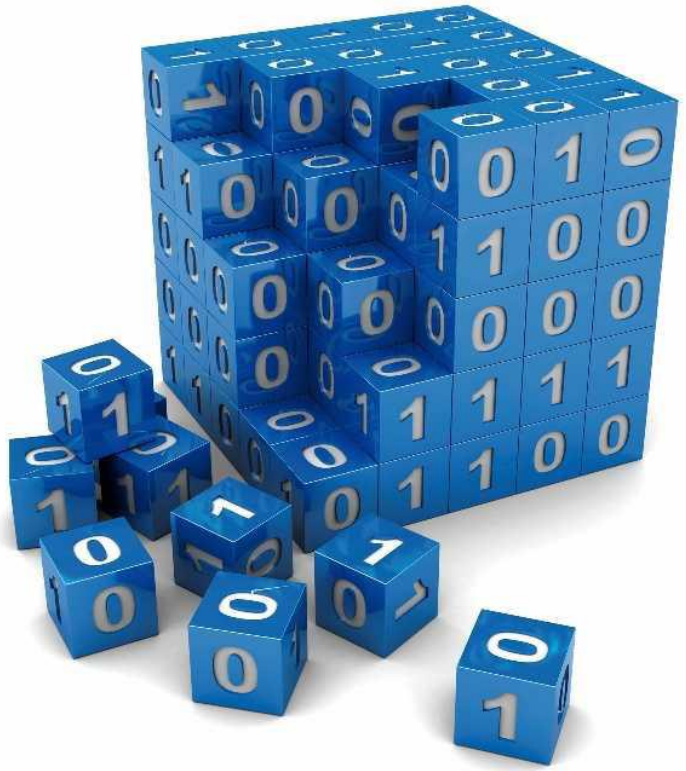
II. CHIFFREMENT MODERNE

- Notion de clé
- Modes de chiffrement
- Chiffrement à clé privée
- Chiffrement à clé publique
- Fonction de hachage et empreinte digitale
- Cryptanalyse
- Attaques réseau..



CHIFFREMENT MODERNE

- Les algorithmes actuels manipulent des bits au lieu des caractères.

- Le codage et le décodage des textes sont basés sur la notion de clés.

❖ I. Clé ?

- Une clé est une valeur représentée par des bits et qui est utilisée avec un algorithme cryptographique pour produire un texte chiffré spécifique.

- Le niveau de sécurité d'un algorithme dépend de la clé.

✓ Plus la clé est longue plus le chiffrement est sûr.

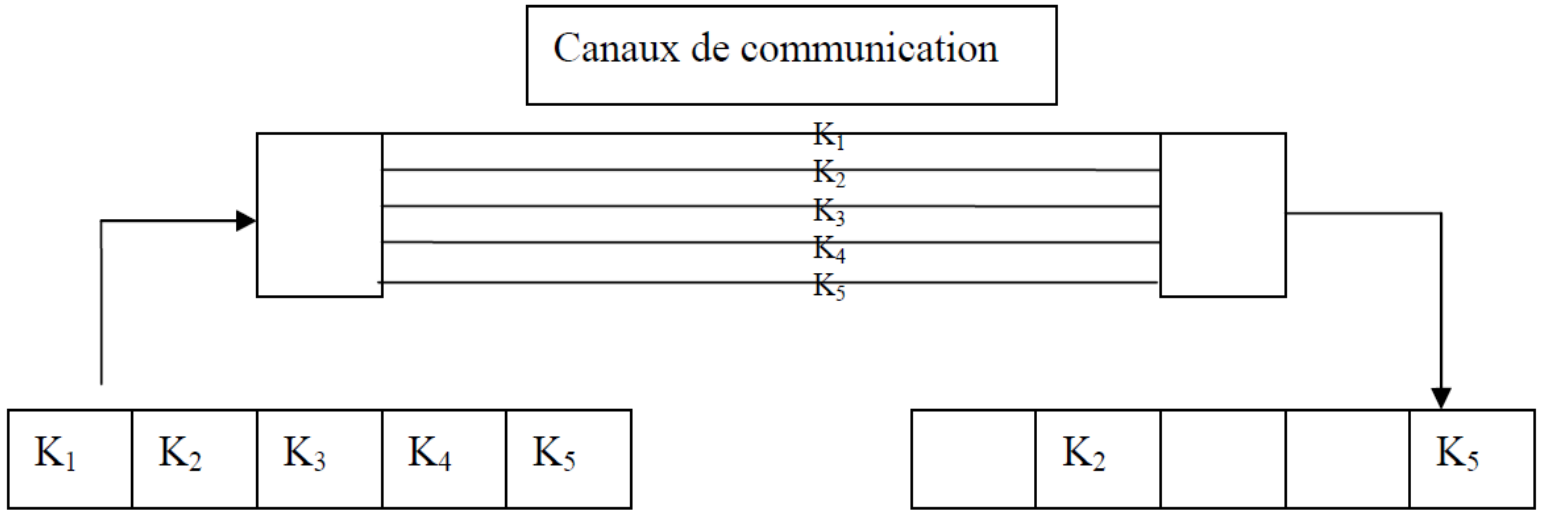
❖ II. Gestion des clés

- La gestion des clés est la procédure de stocker et de distribuer les clés cryptographiques aux destinataires autorisés de façon sûrs.

- Transfert de clés
- Utilisation des clés
- Stockage des clés
- Duplication des clés
- Longévité des clés
- Procuration des clés

• Transfert de Clés

- ➢ Utiliser un seul canal de communication
  - ○ Inconvénient : écoute du canal
- ➢ Diviser la clé en morceaux et envoyer chaque morceau par un canal différent : téléphone, émail…



  - ○ Inconvénient : si la clé change tous les jours (non commode).

✓ Solution : envoyer toutes les clés (d'un mois) en une seule fois :

- bien ranger cette liste de clés (utiliser une clé de chiffrement de clés (clé maitresse)).

- **Utilisation des Clés**

  - Utiliser un logiciel (Chiffrement par logiciel)

    - Inconvénients:
      - L'application de chiffrement ne s'exécute pas sans interruption (système d'exploitation multi-taches)

      [Interruption de chiffrement (sauvegarde de la clé et du programme de chiffrement]
      - Risque de récupérer la clé par un attaquant

• Stockage des Clés

- La clé est mémorisée par l'utilisateur :

  - Soit entrer la clé (de 64 bits par exemple)

  - Soit entrer la clé sous forme d'une chaîne de caractères et utiliser une technique de broyage de clés :
    - Transformer la chaîne de caractères en une suite de bits.
    - Découper la clé en deux parties : une partie mémorisée par l'utilisateur et l'autre partie par le système

- **Duplication des Clés**
  - ➢ Une personne responsable doit connaître et sauvegarder toutes les clés (ex : celles des employés):
  - ○ Inconvénient : Risque d'utilisation des clés pour des fins personnelles (par le responsable)
  - ✓ Solution : Partage de secret
- **Longévité des Clés**
  - ➢ Aucune clé ne doit être utilisée pour une période indéfinie

- **Procuration des Clés**
  - Différents moyens de se procurer la clé publique de quelqu'un :
    - Directement de la personne concernée.
    - D'une BD centralisée (Autorité de certification ou de confiance).
    - A partir de sa BD privée.

- **La gestion des Clés peut être :**
  - Centralisée (certificat de clés publiques) : une clé publique de quelqu'un est signée par une personne (autorité de confiance)
  - Distribuée : un ensemble de personnes (parrains) signent la clé publique de quelqu'un. Exemple de système utilisant ce procédé est le PGP.

III. Modes de chiffrement

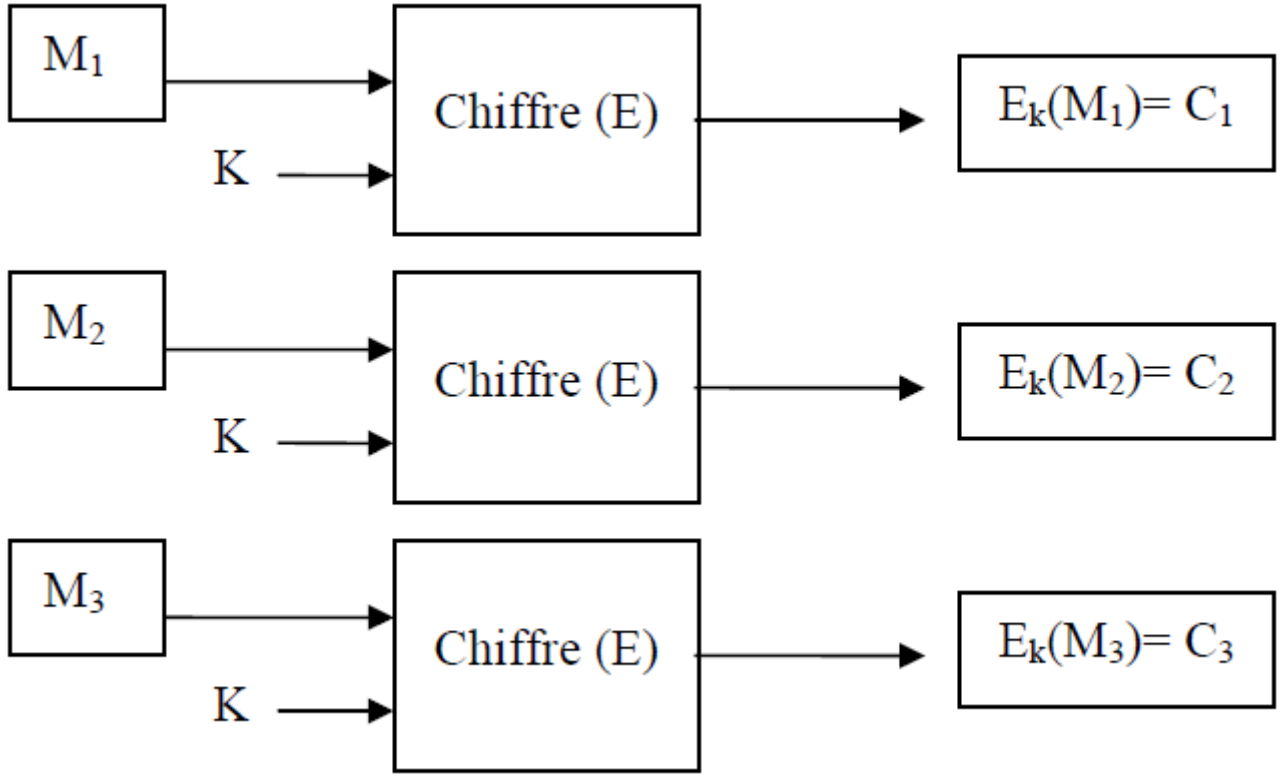
1. Chiffrement par bloc

A-Le mode ECB (Electronic Code Book : carnet de codage électronique)

- Le document à crypter est découpé en blocs de 64 bits.
- Chaque bloc est crypté indépendamment des autres.
- un même bloc en clair est toujours chiffré par le même bloc (avec la même clé)
- taille des blocs est 64 bits ===> un carnet de codage de 2^{64} entrées
- pour chaque clé différente ===> un carnet de codage différent.

ECB : Principe

T[n] = nième bloc de texte clair.
C[n] = nième bloc de texte chiffré.
E(m) = fonction de chiffrement du bloc m.
D(m) = fonction de déchiffrement du bloc m.
Chiffrement : C[n] = E(T[n])
Déchiffrement : T[n] = D(C[n])



ECB :

- **Avantages :**

chaque bloc est crypté indépendamment des autres (chiffrement non linéaire : pas d'ordre).

- **Inconvénients :**

Si un cryptanalyste dispose de textes chiffrés et de textes en clair , il peut (commencer à) construire le carnet de codage sans connaître la clé.

Exemple : un message contenant des redondances est facile à cryptanalyser (entête et fin de documents).

ECB : Exemple (Bloc rejoué)

Transfert d'argent entre banques (Les messages sont chiffrés en mode ECB.)

Un attaquant peut modifier les messages sans connaître la clé (il peut devenir riche).

Format de message :

1	2	3	4	5	6	7	8	9	10	11	12
12 Octets Banque1		12 Octets Banque2	48 octets Nom du déposant						16 octets N° cpte		8 Mon tant

Attaque :

* Enregistrer tous les messages chiffrés de banque1 vers banque2.
* Transférer une somme d'argent (1000 D.A.) de banque1 vers son compte de banque2.
* Recommencer l'opération une deuxième fois.
* Chercher une paire de messages identiques
* Finalement : envoyer à volonté le message et devenir riche…

Bloc rejoué : solution

Ajouter une datation (1 octet)

1	2	3	4	5	6	7	8	9	10	11	12	13
1 octet	12 Octets		12 Octets	48 octets						16 octets		8 octet
Datation	Banque1		Banque2	Nom du déposant						N° cpte		Montant

Attaquant :

- extraire les 8 blocs [5-12] correspondant au nom et numéro de compte (de l'attaquant)
- intercepte aléatoirement des messages
- remplacer les 8 blocs par le nom et numéro (de l'attaquant)

Remarque : ECB est très fragile aux attaques par bloc rejoué.

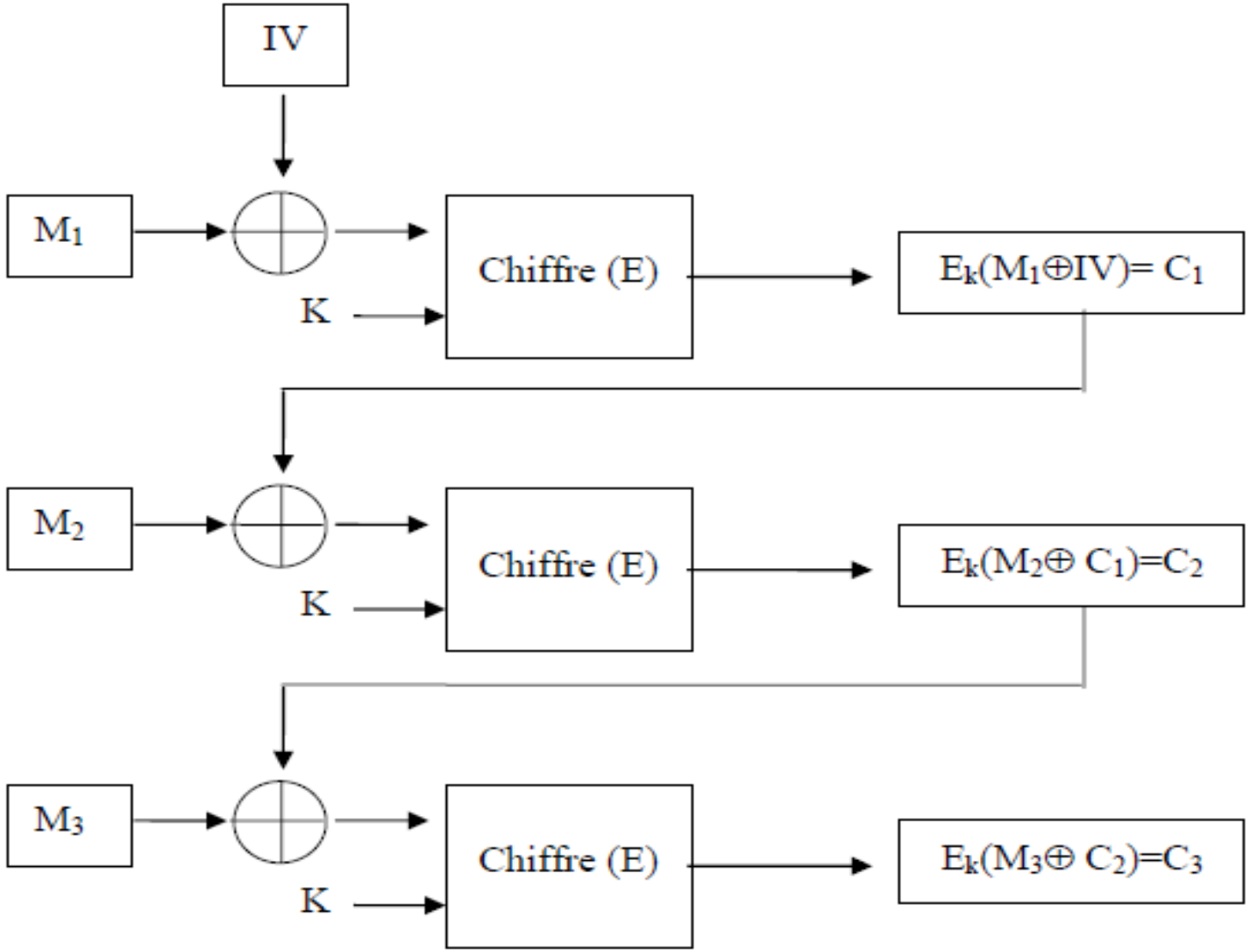
B-Le mode CBC (Cipher Block Chaining : Chiffrement par chaînage simplifié de blocs)

- Avant de crypter un bloc en clair, un XOR est effectué entre ce bloc en clair et le bloc précédemment crypté.
- Problème : Deux messages identiques donnent le même message crypté

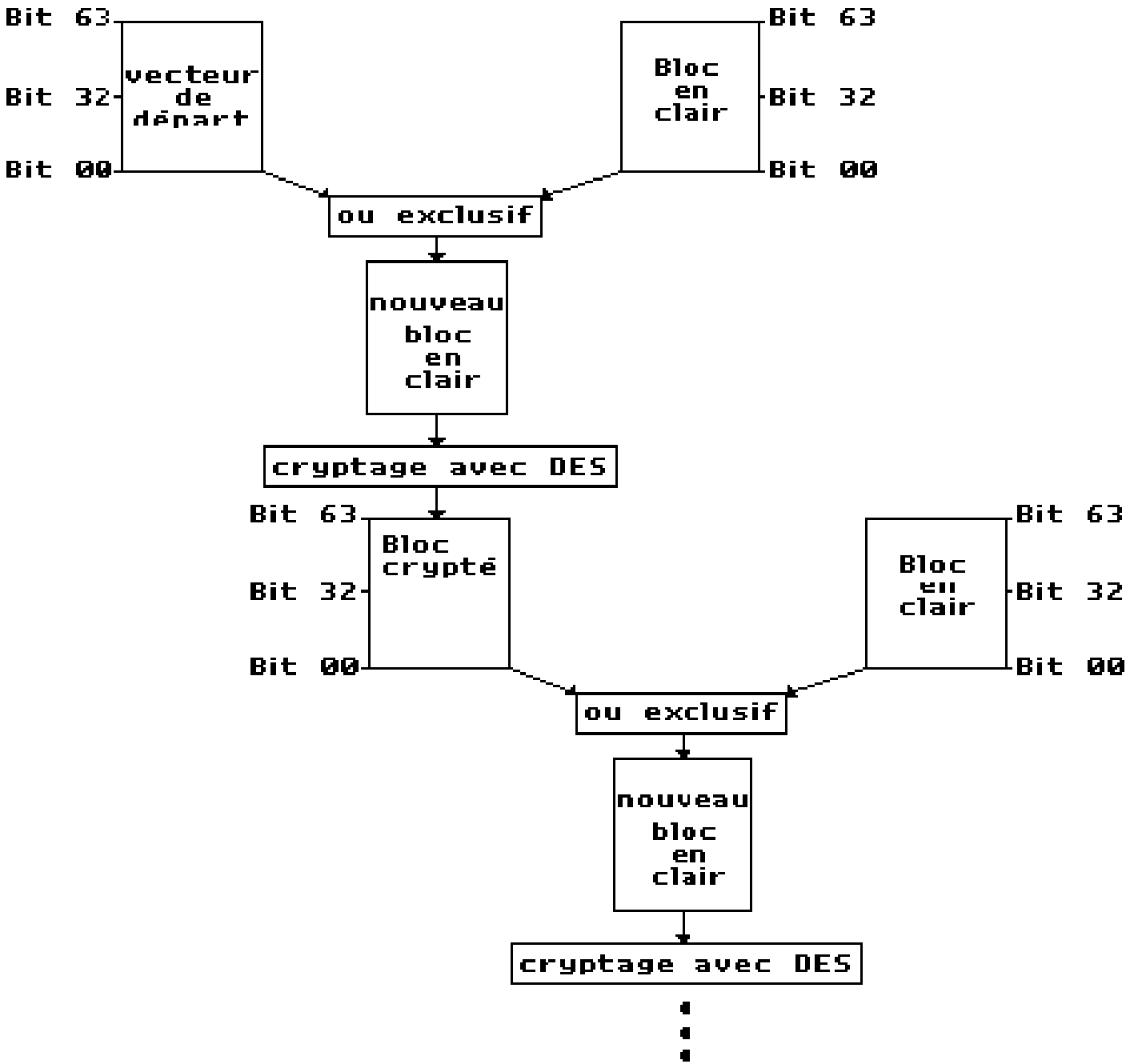
✓ Solution : utiliser un vecteur aléatoire (IV) pour rendre chaque message unique ===> deux messages identiques donnent 2 textes différents

CBC : Principe

- Chaque bloc est crypté en fonction du bloc précédemment crypté



CBC : Mode opérationnel



C-Le mode CFB (Cipher FeedBack : Chiffrement à rétroaction)

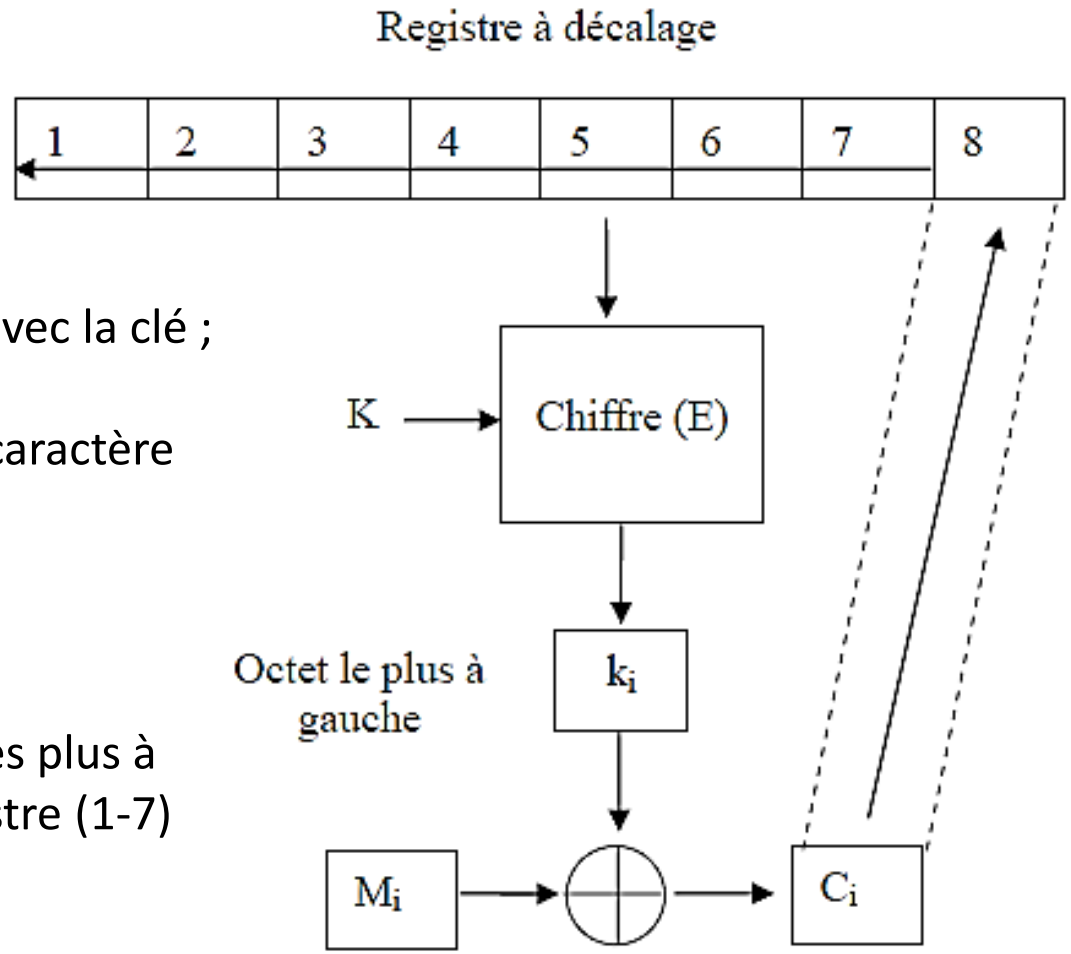
- Les données sont chiffrées par unité plus petite que la taille du bloc (< 64bits).

- Exemple :

Chiffrement caractère par caractère (CFB à 8 bits) :

  - Après chiffrement du caractère, il peut être transféré avant de terminer le chiffrement de tout le bloc (8Ø).

• CFB à 8 bits



(1) : chiffrer le vecteur aléatoire (registre) avec la clé ;

(2) : $\mathbf{C_i}$= [8 bits les plus à gauche ($\mathbf{k_i}$)] XOR [caractère du bloc courant ($\mathbf{M_i}$)] ;

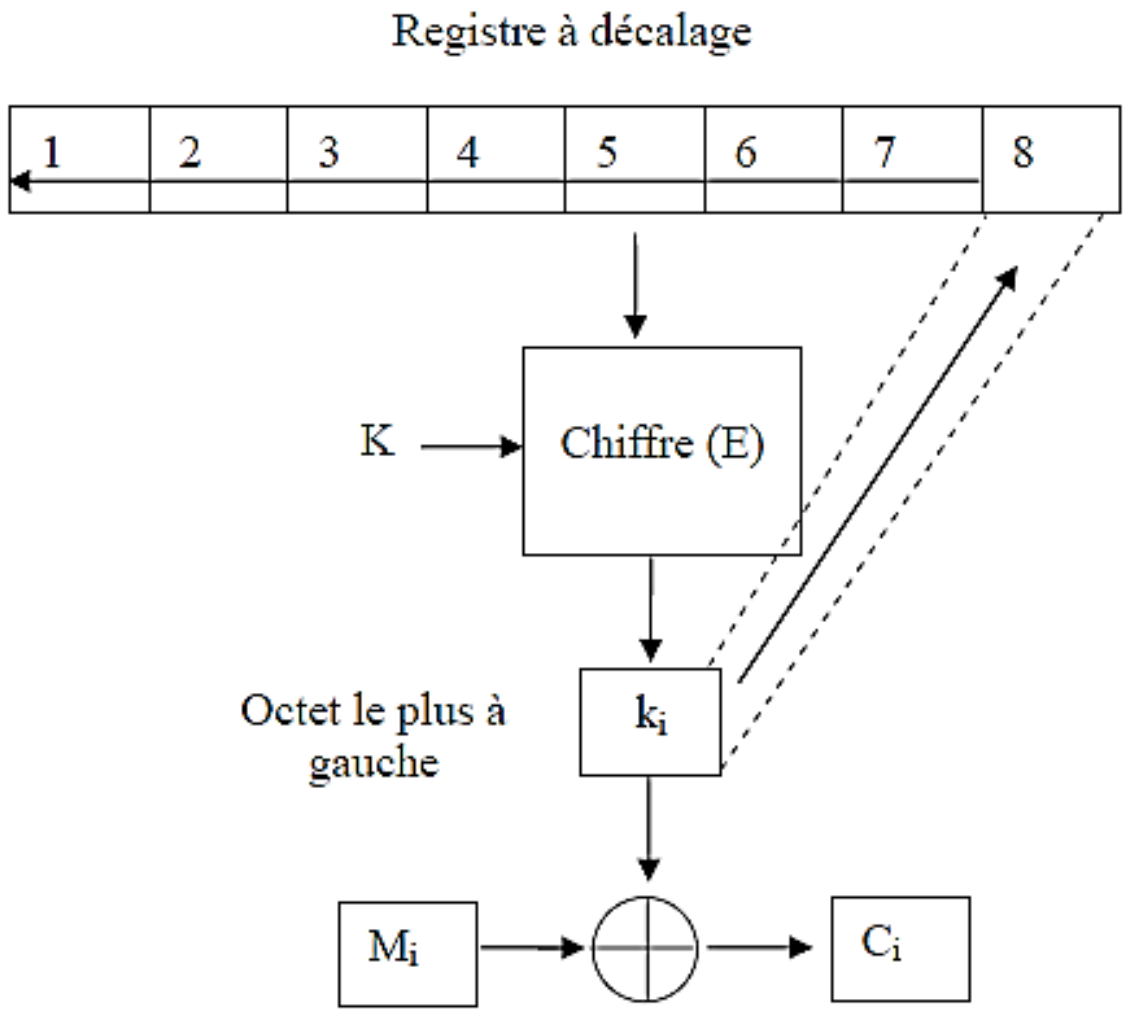
(3) : $\mathbf{C_i}$ peut être transmis ;

(4) : $\mathbf{C_i}$ est placé dans le registre **IV** (8 bits les plus à droite) et décaler à gauche le reste du registre (1-7) de 8 positions ;

(5) : aller à 2.

D-Le mode OFB (Output Feedback : Chiffrement à rétroaction de sortie)

- Similaire à CFB avec la sortie k_i au lieu de C_i



E-Le mode Compteur

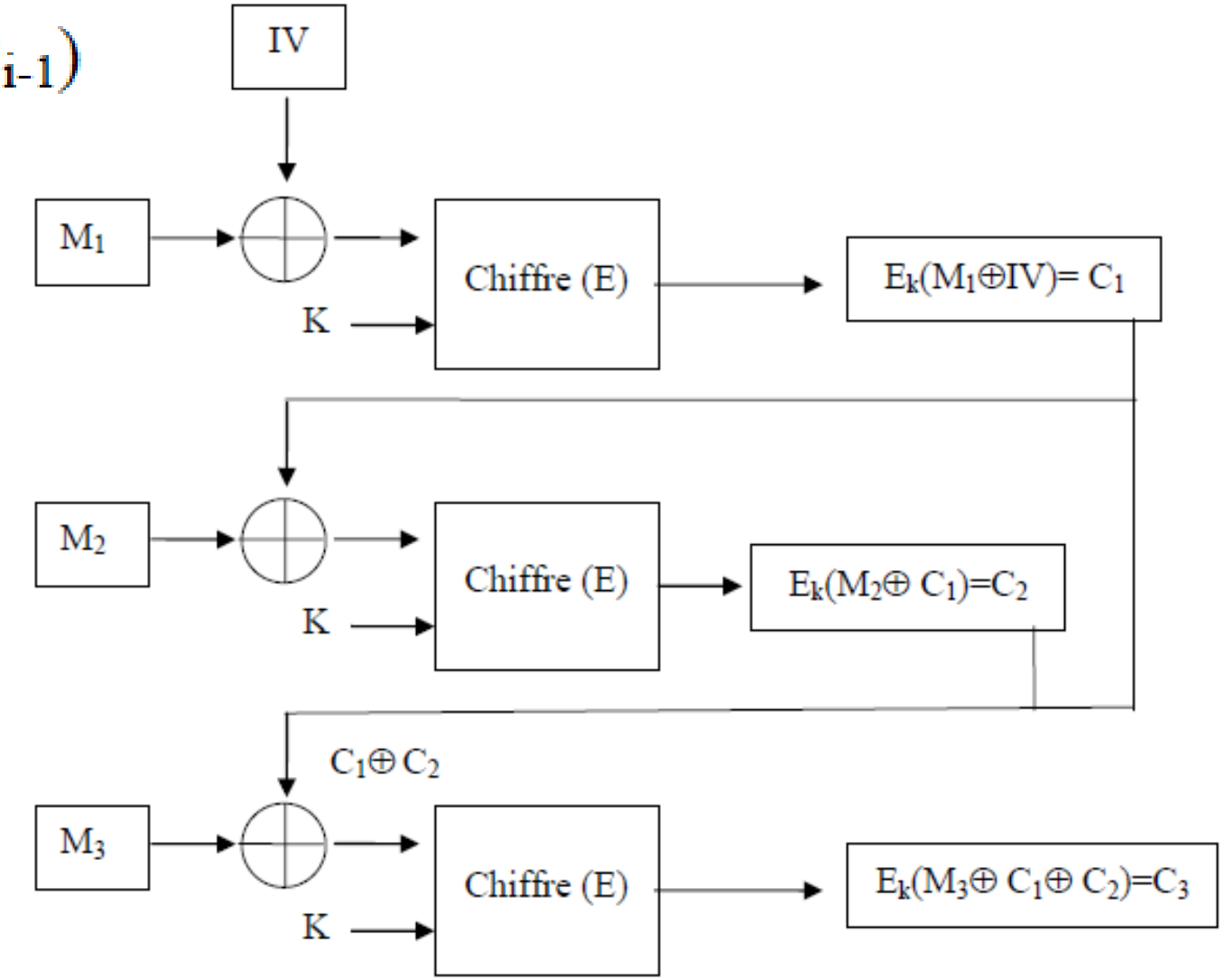
- Au lieu d'utiliser la sortie de chiffrement $\mathbf{k_i}$ pour remplir le registre à décalage, on utilise un compteur .

- après chiffrement d'un bloc, incrémenter le registre

F-Le mode BC (Block Chaining: Chaînage de blocs)

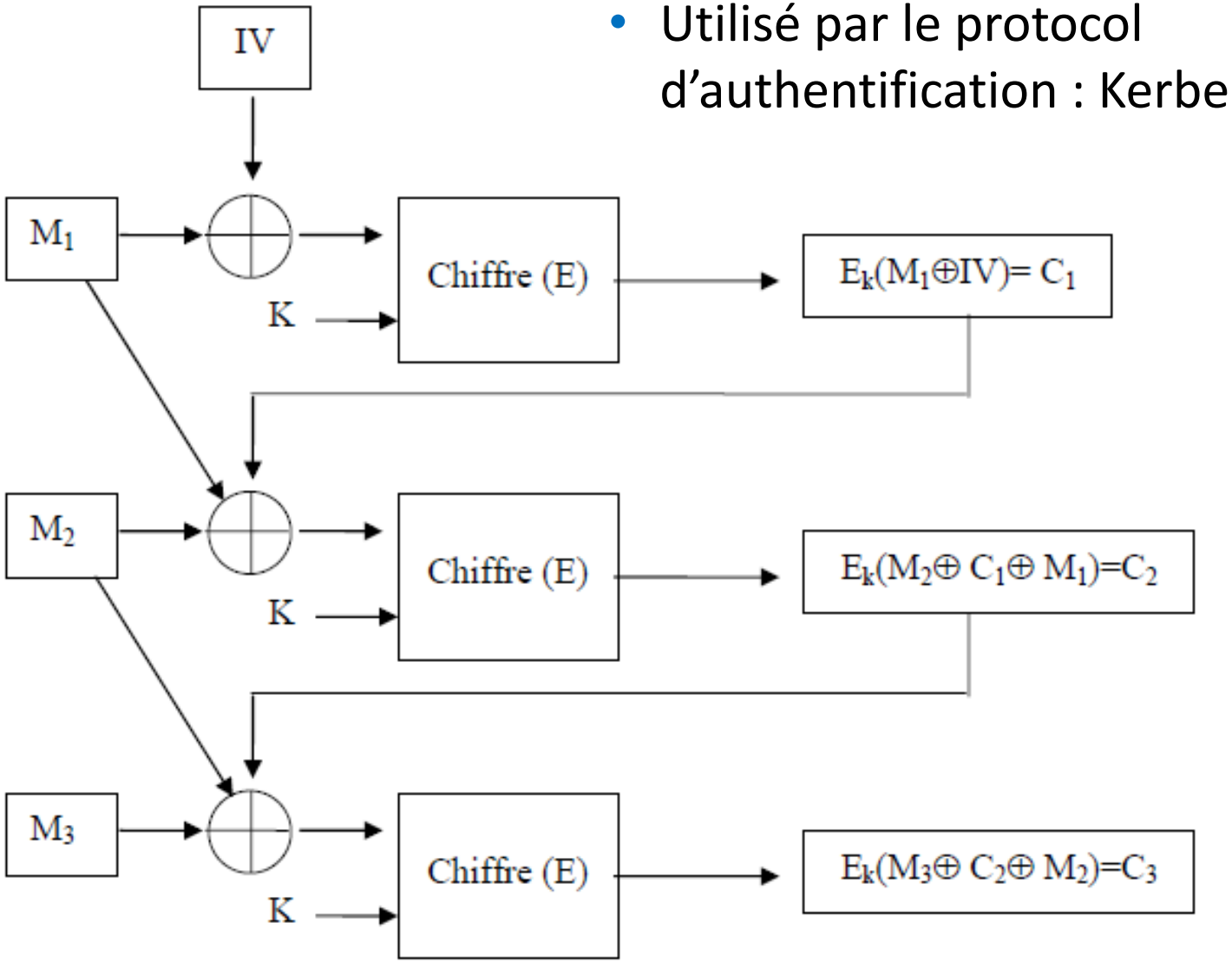
- Chaque bloc est crypté en fonction de tous les blocs précédemment cryptés.

$$C_i = E_k(M_i \oplus C_1 \oplus \ldots C_{i-1})$$



G-Le mode PCBC (Propagating Cipher Block Chaining: Chiffrement par Chaînage de blocs avec propagation)

$C_i = E_k(M_i \oplus C_{i-1} \oplus M_{i-1})$



- Utilisé par le protocol d'authentification : Kerberos

2. Le Sur-Chiffrement

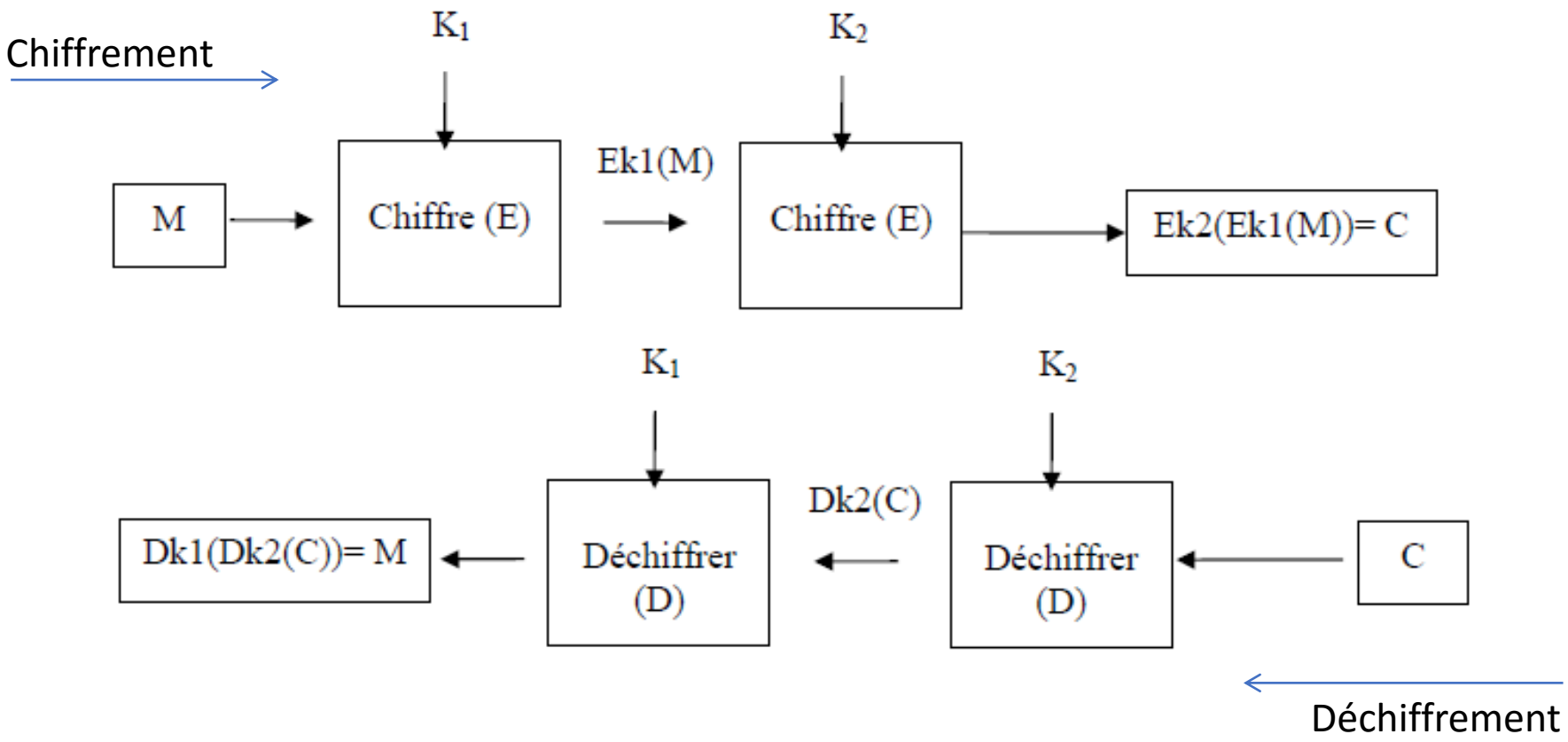
- **Principe:** Chiffrer le même bloc de texte en clair un certain nombre de fois.

- ***Exemple :***

Chiffrer un texte deux fois avec un même algorithme et une même clé .

Inc. : ne modifie pas la complexité d'une attaque exhaustive
Sol. : augmenter le niveau de sécurité en utilisant plusieurs clés différentes.

A- Sur-chiffrement double:



Une recherche exhaustive nécessite 2^{2n} tentatives (ex : bloc de 64 bits nécessite 2^{128} tentatives).

B- Sur-chiffrement triple: (EDE : Encrypt-Decrypt-Encrypt)

- Chiffrer un bloc trois fois avec deux clés différentes :

  chiffrer + déchiffrer + chiffrer

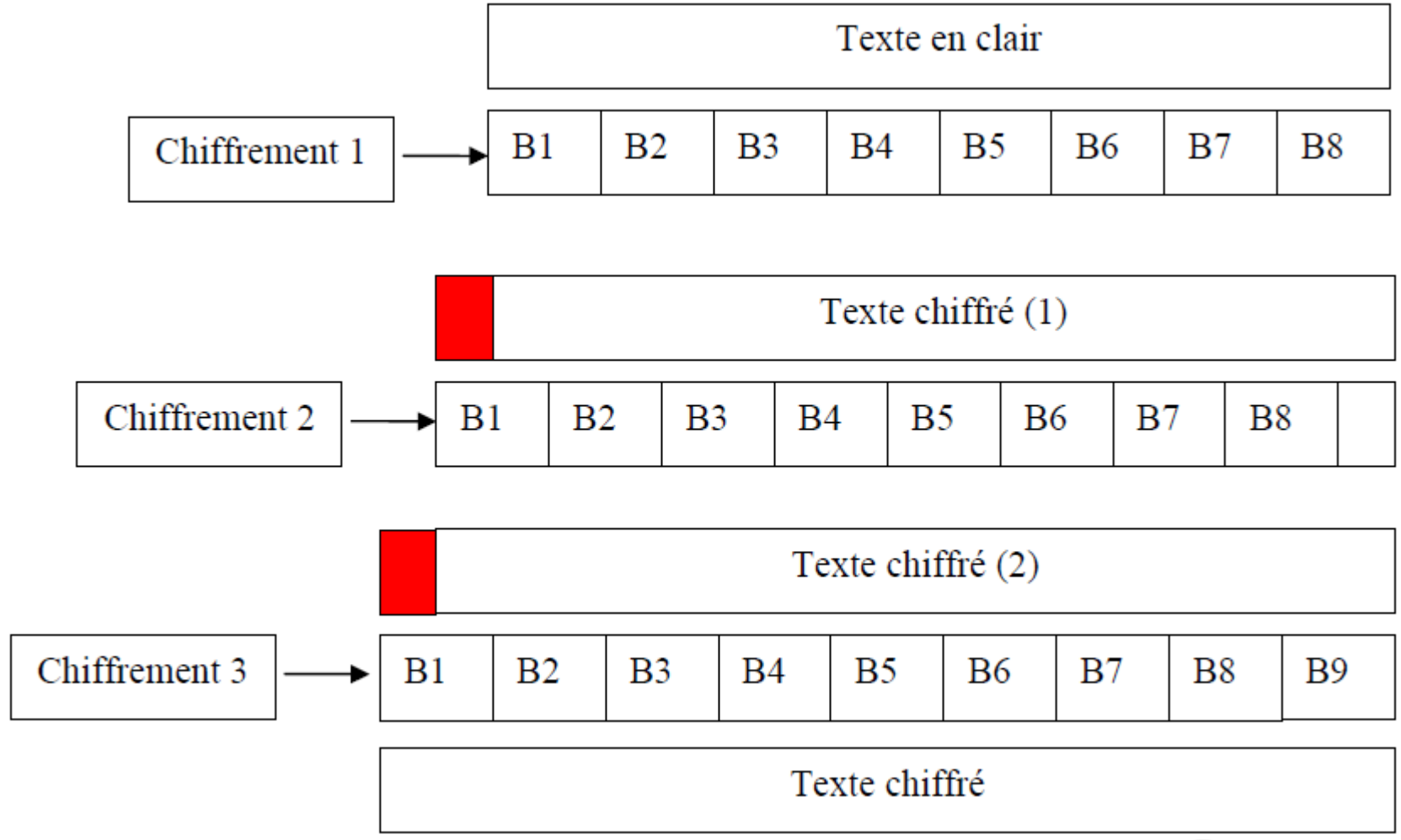
$$C=Ek1(Dk2(Ek1(M)))$$
$$M=Dk1(Ek2(Dk1(C)))$$

- Ou bien chiffrer avec trois clés différentes :

$$C=Ek3(Dk2(Ek1(M)))$$
$$M=Dk1(Ek2(Dk3(C)))$$

C- Sur-chiffrement triple avec remplissage

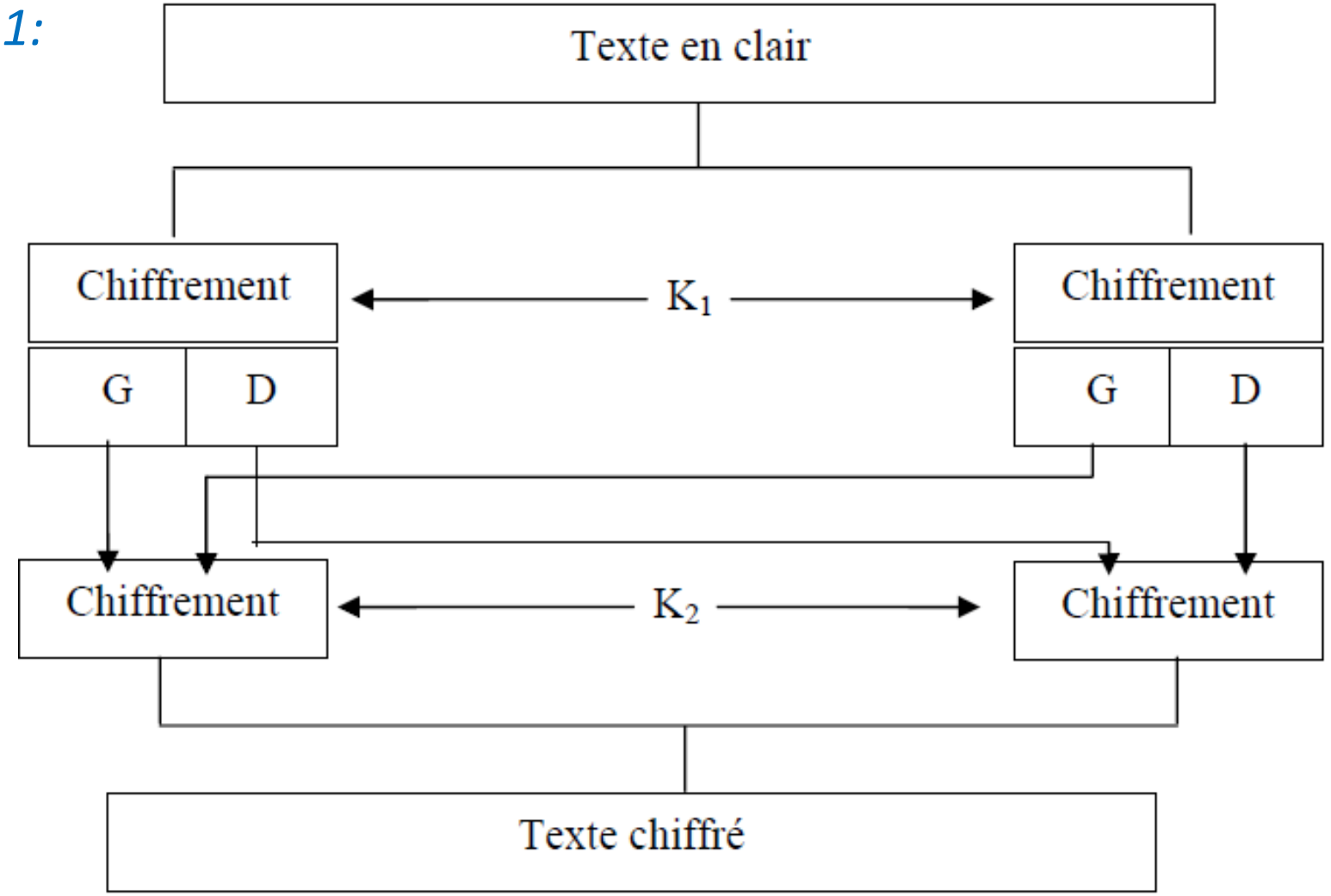
Compléter le texte avec une chaine de bits (1/2 longueur du bloc) entre le 1er et le 2ème chiffrement et entre le 2ème et 3ème chiffrement.



D- Doublement de la longueur du bloc

Doubler la longueur du bloc d'un algorithme en utilisant le sur-chiffrement.

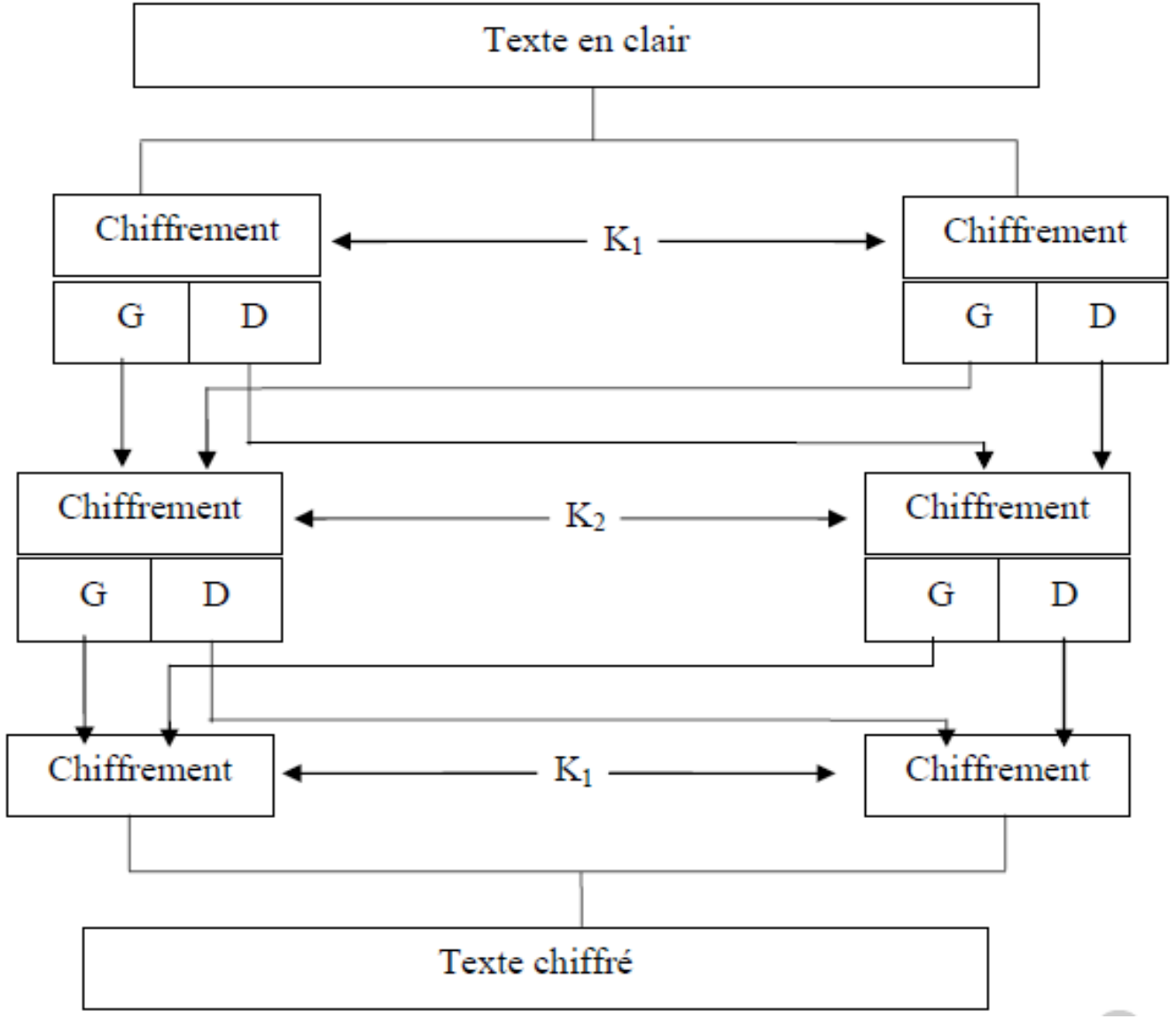
Solution 1:



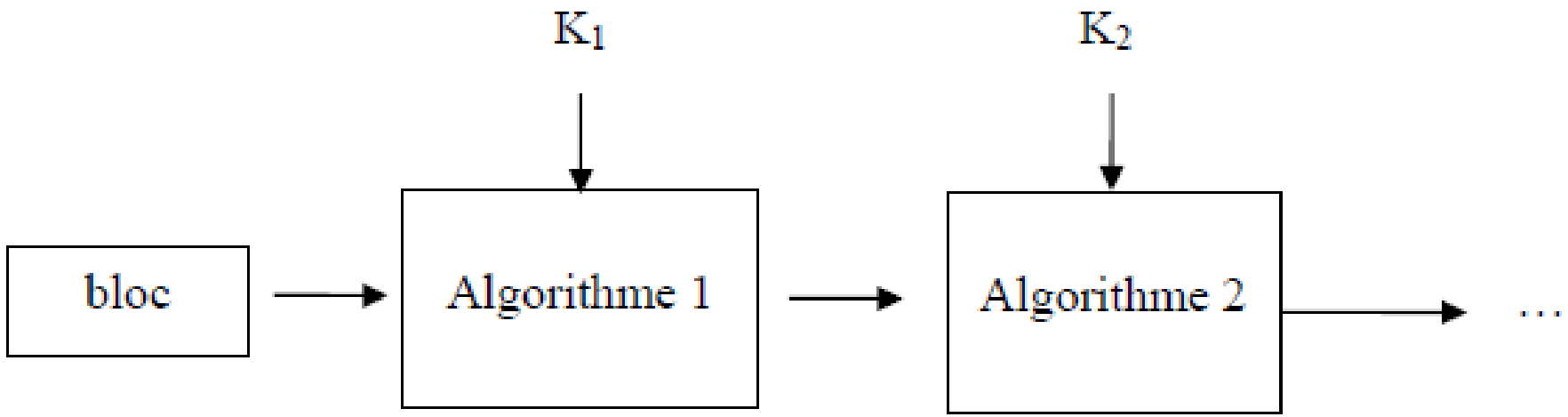
D- Doublement de la longueur du bloc

Doubler la longueur du bloc d'un algorithme en utilisant le sur-chiffrement.

Solution 2:



E- Sur-chiffrement avec plusieurs algorithmes



2. Le Chiffrement en continu

Principe : Chiffre et déchiffre 1 bit à la fois.

Inc. : Réalisation logicielle difficile, car la manipulation des bits est coûteuse en temps de calcul.

Avt. : Minimiser la Propagation d'erreurs : une erreur d'1 bit dans un texte chiffré provoque un seul bit erroné dans l'opération de déchiffrement,

contrairement au chiffrement par bloc où 1 erreur dans un texte chiffré entraîne au moins 1 bloc erroné dans l'opération de déchiffrement.

IV. Chiffrement à clé symétrique

- **Chiffrement à clé secrète**

- Une [seule et même] clé est utilisée à la fois pour le chiffrement et le déchiffrement

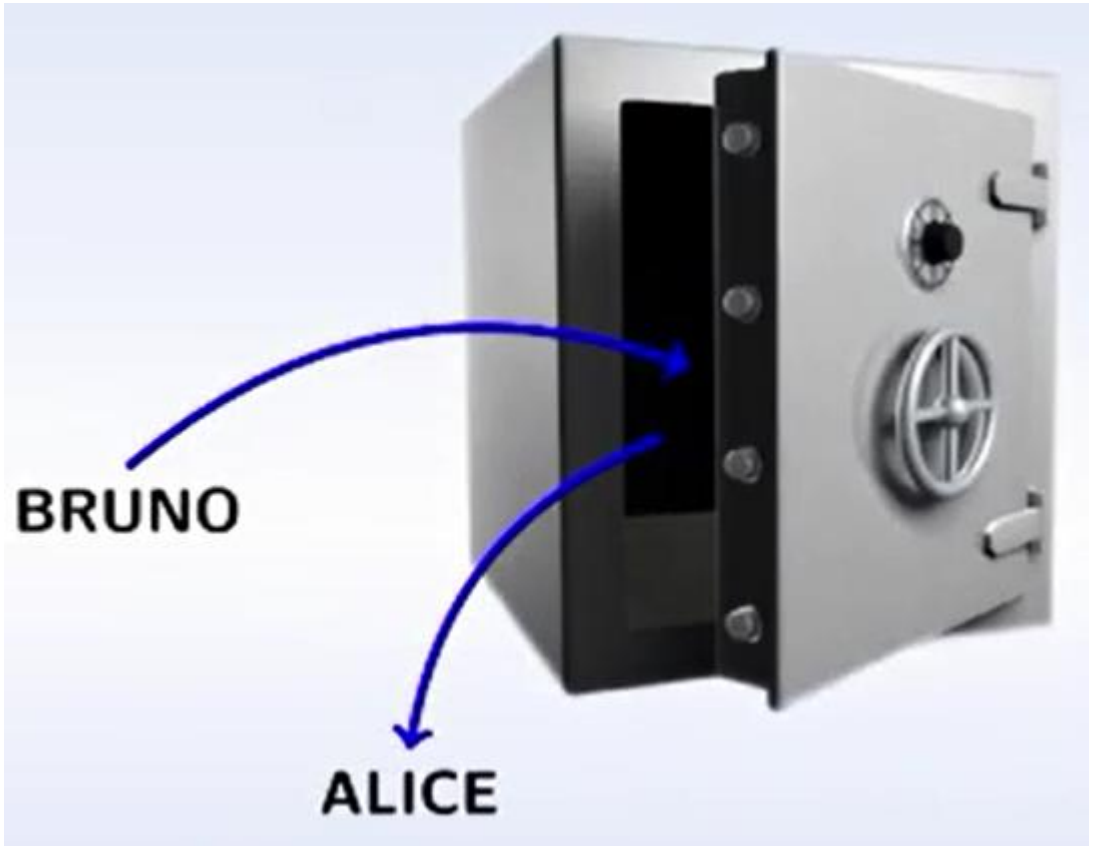
- Les deux interlocuteurs soient mis d'accord sur une clé privée auparavant

Chiffrement à clé privée

❑ Principe

Chiffrement à clé privée

❑ Principe



Les algorithmes à clé symétrique :

DES, AES, IDEA, 3DES, CAST, Skipjack, Serpent, Mars…

Avantages : Très rapide.

Inconvénient : Transfert de clés non sécurisé

Pour Communiquer de manière sécurisée

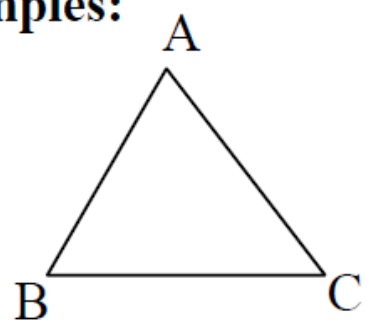
➔

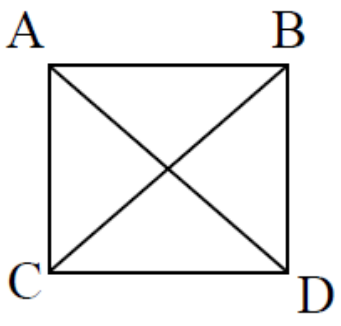
Utiliser une clé pour chaque paire

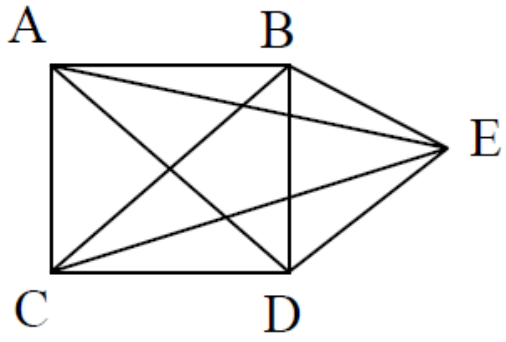
➔

$(n^2 - n) / 2$ clés

Exemples:







Pour 4 utilisateurs, il faut 6 clés différentes

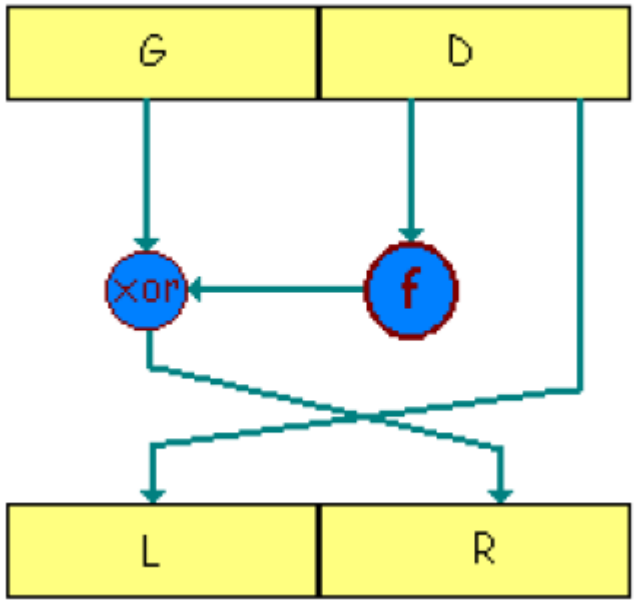
5	10 clés différentes,
10	45 clés

Objectif des algorithmes à clé secrète

✓ **Chercher la perfection = Chercher l'aléatoire**

il faut que le message codé paraisse aussi aléatoire que possible pour limiter les risques d'attaque

Bijection aléatoire de Feistel



- Choisir une fonction **f** presque aléatoire qui a comme argument **n** bits.
- Chiffrer des blocs partagés en deux parties Gauche et Droite.

➢ Chiffrement : L = D et R = G xor f(D)

➢ Déchiffrement : D = L et G = R xor f(L)

On répète le schéma de Feistel un certain nombre de fois
(on parle de tours - le DES en comporte 16)

❑ DES (Data Encryption Standard)

- **DES** est l'outil officiel de cryptographie du gouvernement américain, mis au point par IBM dans les années 1970.
- Système de chiffrement par blocs de 64 bits et à clé secrète de 64 bits.

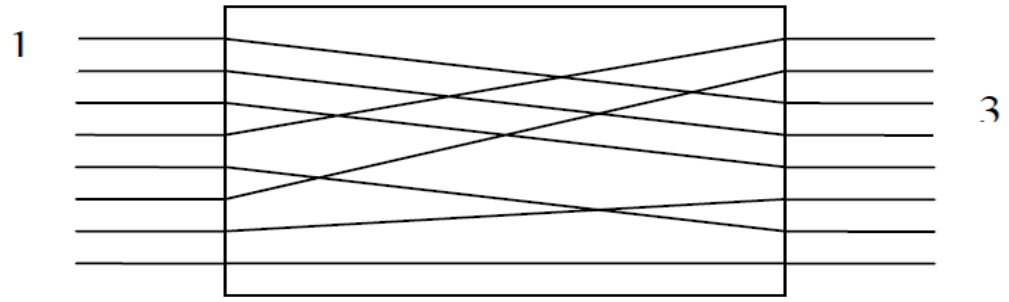
DES :

- SYMÉTRIQUE
- RÉVERSIBLE
- PAR BLOCS
- A CLÉ SECRETE.

❑ DES : principe

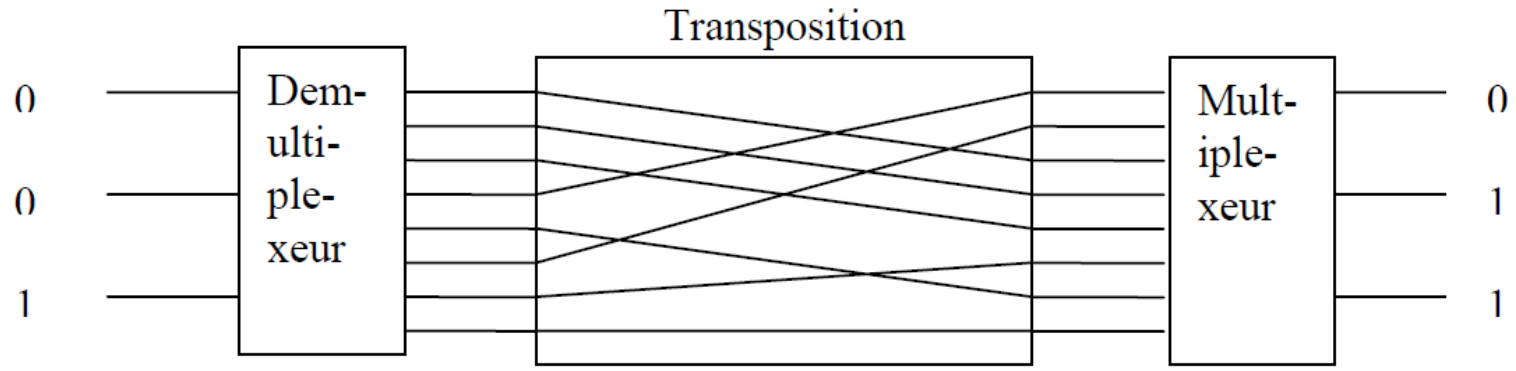
Transposition + Substitution

P-Box : Solution matérielle :



Le bit 1 remplace le 3

S-Box :- Solution matérielle



Trois bits sélectionnent un fil en sortie, ensuite l'ensemble subit une transposition et le résultat est remultiplexé sur 3 bits.

❑ DES : Clé

La clé du DES est une chaîne de 64 bits : seuls 56 bits servent réellement à définir la clé. Les 8 bits restants (8, 16, 24, 32, 40, 48, 56, 64) sont des bits de parité

2^{56} clés possibles (soit environ 72 millions de milliards possibilités)

❑ DES : Etapes de chiffrement

Message Clair = Série de blocs de 64 bits

- **Etapes :**

DES utilise une clé secrète de 56 bits, qu'il transforme en 16 "sous-clés" de 48 bits chacune (une pour chaque itération). Le cryptage se déroule sur 19 étapes :

1ère étape

La première étape est une transposition fixe (standard) des 64 bits à crypter.

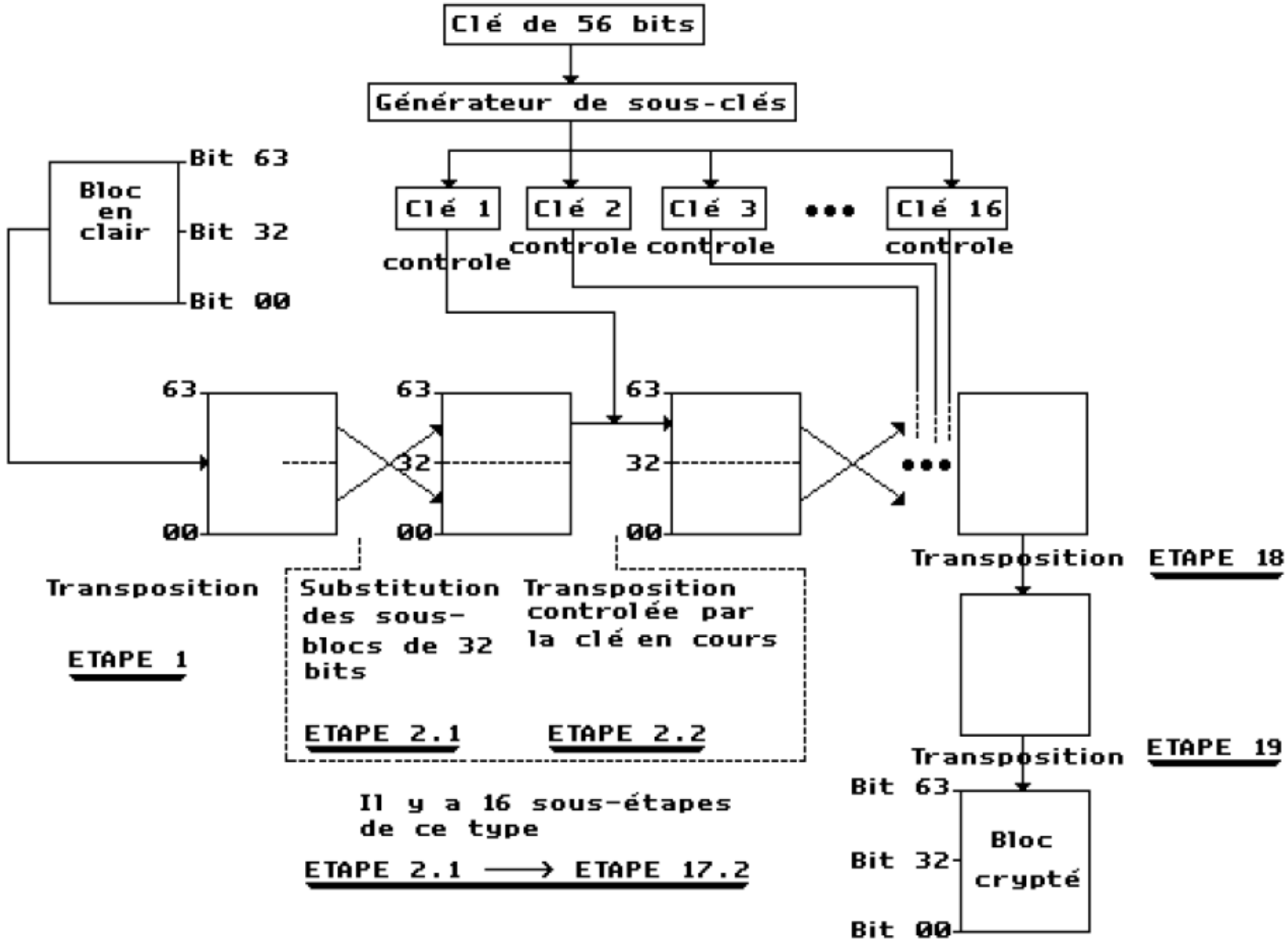
16 étapes suivantes

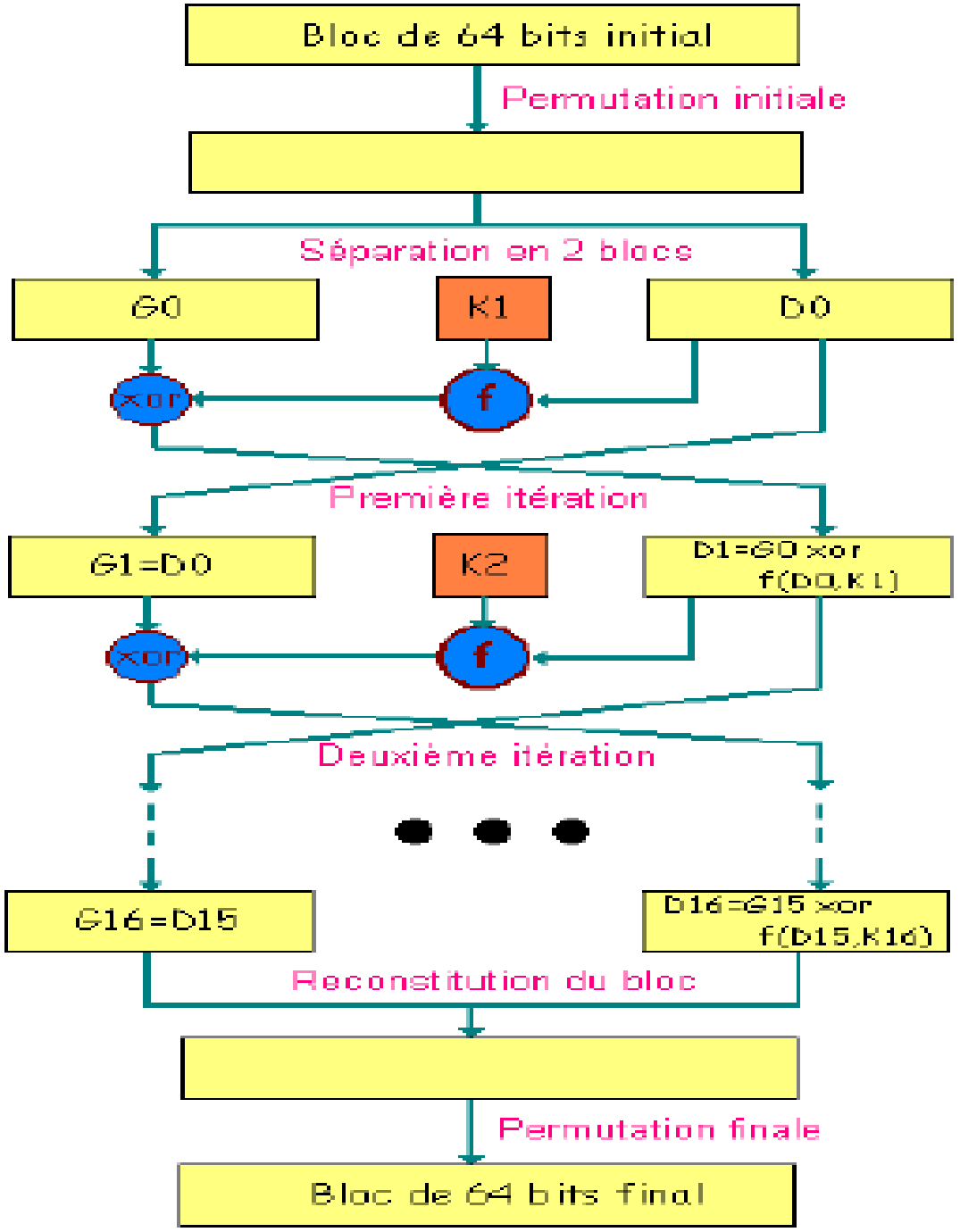
Les 16 étapes suivantes peuvent être divisées en 2 "sous-étapes" chacune. Dans un premier temps, Le bloc de 64 bits est découpé en 2x32 bits, et une substitution est effectuée entre ces deux blocs, en fait, ces deux blocs seront tout simplement échangés l'un avec l'autre. Dans un second temps, le bloc de 32 bits ayant le poids le plus fort (le bloc qui va du bit n°32 au bit n°63) subira une transposition contrôlée par la sous-clé correspondante à l'étape en cours.

Etape 18 et 19

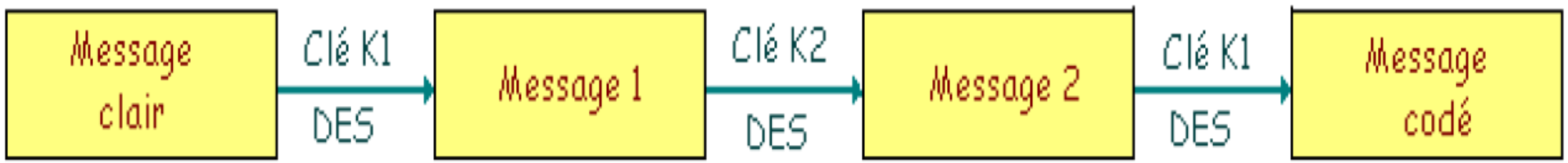
Les deux dernières étapes sont deux transpositions.

SHEMA REPRESANTANT L'ALGORITHME DES




Bloc de 64 bits initial
Permutation initiale
Séparation en 2 blocs
G0
K1
D0
xor
f
Première itération
G1=D0
K2
D1=G0 xor f(D0,K1)
xor
f
Deuxième itération
G16=D15
D16=G15 xor f(D15,K16)
Reconstitution du bloc
Permutation finale
Bloc de 64 bits final

❑ Triple-DES



T-DES avec 2 clés différentes (clé de 112 bits : k1+k2)

❑ **AES** (*Advanced Encryption Standard*)

- L’algorithme procède par blocs de 128 bits, avec une clé de 128 bits également.

- Chaque bloc subit une séquence de 5 transformations répétées 10 fois.

❑ AES : Etapes de chiffrement

1. Addition de la clé secrète (par un ou exclusif).

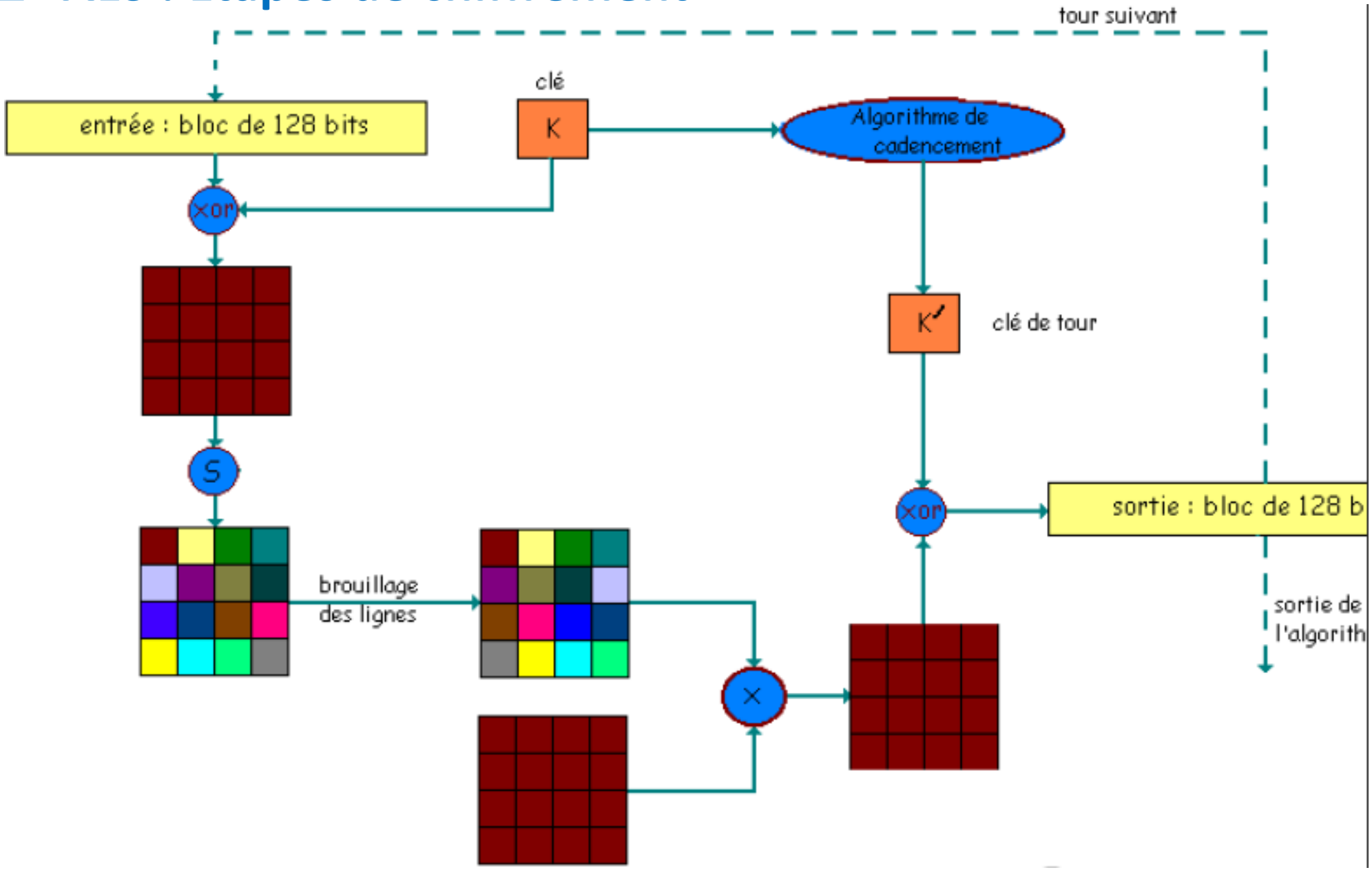
2. Transformation non linéaire d'octets : les 128 bits sont répartis en 16 blocs de 8 bits, eux-même dispatchés dans un tableau 4×4. Chaque octet est transformé par une fonction non linéaire S.

3. Décalage de lignes : les 3 dernières lignes sont décalées cycliquement vers la gauche : la 2ème ligne est décalée d'une colonne, la 3ème ligne de 2 colonnes, et la 4ème ligne de 3 colonnes.

4. Brouillage des colonnes : Chaque colonne est transformée par combinaisons linéaires des différents éléments de la colonne (ce qui revient à multiplier la matrice 4×4 par une autre matrice 4×4).

5. Addition de la clé de tour : A chaque tour, une clé de tour est générée à partir de la clé secrète par un sous-algorithme (dit de cadencement). Cette clé de tour est ajoutée par un ou exclusif au dernier bloc obtenu.

AES : Etapes de chiffrement



5 transformations répétées 10 fois