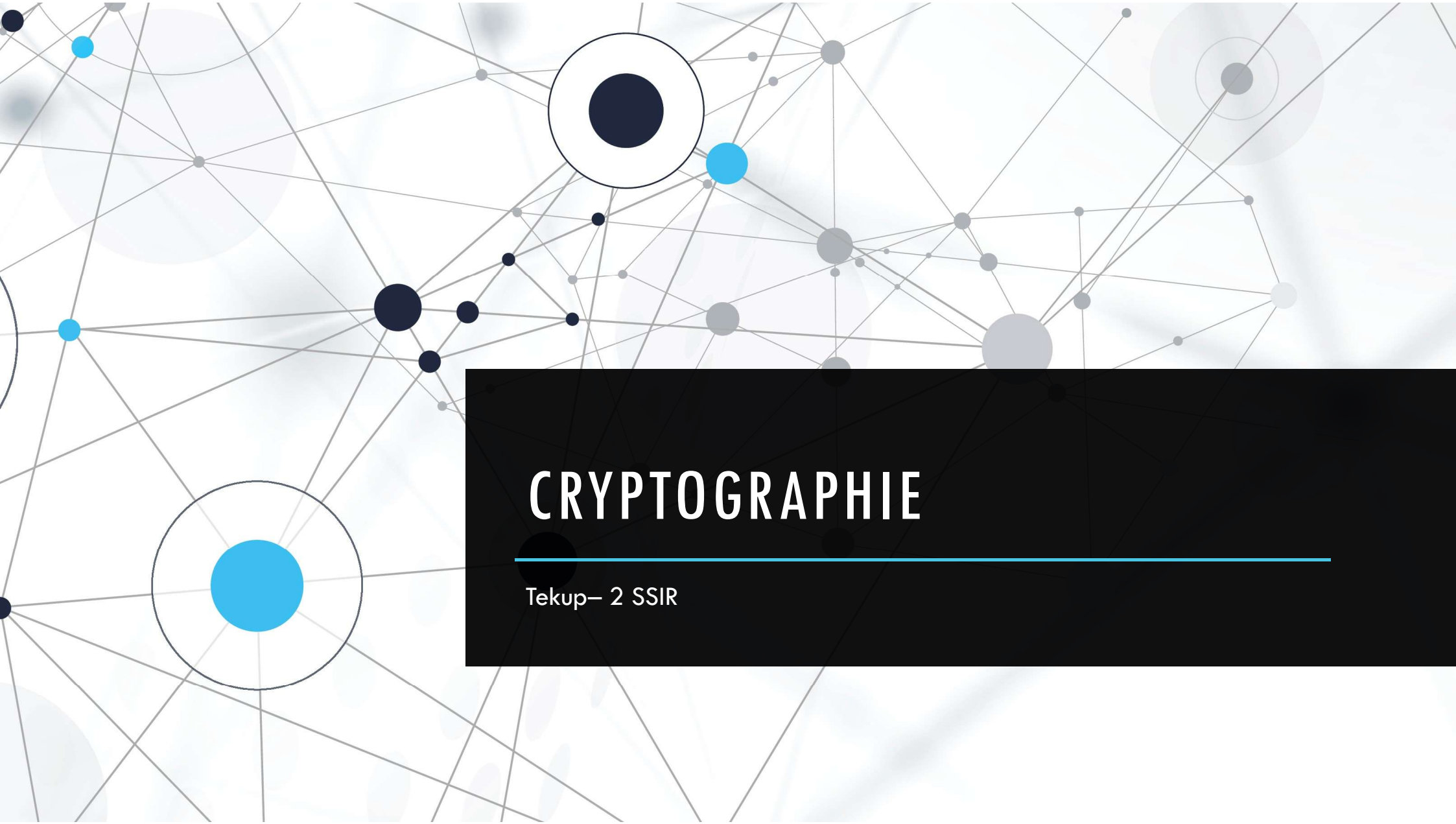
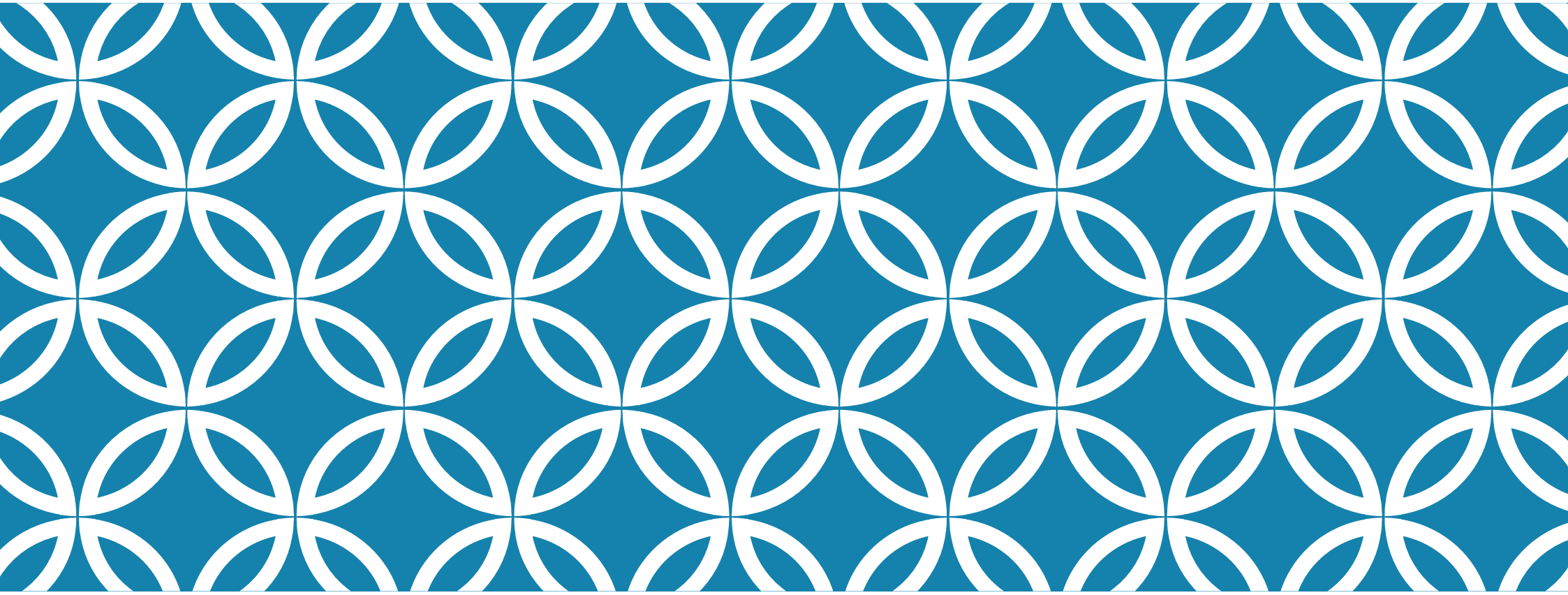


CRYPTOGRAPHIE

Tekup– 2 SSIR



CRYPTAGE SYMÉTRIQUE

RAPPEL (1)

- **Cryptologie:** science du secret
- **Cryptographie :** l'art de rendre inintelligible, de crypter, de coder, un message
- **Cryptanalyse:** art de "casser" des crypto systèmes
- **Crypto-système:** mécanismes assurant le chiffrement/déchiffrement des messages
- **Un algorithme cryptographique** est l'ensemble des fonctions (mathématiques ou non) utilisées pour le chiffrement et le déchiffrement selon ces deux relations:

$$\begin{cases} E_{K_e}(M) = C \\ D_{K_d}(C) = M \end{cases}$$

RAPPEL (2)

- Le type de relation qui unit les clés K_e et K_d permet de définir deux grandes catégories de systèmes cryptographiques: Les systèmes à **clés secrètes ou symétriques** et les systèmes **à clés publiques ou asymétriques**
- Les clefs de chiffrement et de déchiffrement n'ont aucune raison d'être identiques. Seule la clef de déchiffrement doit impérativement être secrète.
- Les fonctions de codage E et de décodage D peuvent fonctionner de deux façons:
  - en continu (**par flots**): chaque nouveau bit ou octet est manipulé directement
  - **par blocs**: chaque message est d'abord partitionné en blocs de longueur fixe. Les fonctions de chiffrement et déchiffrement agissent alors sur chaque bloc.

PRINCIPE DE KIRCHHOFF

- En 1883 , Auguste Kirchhoff posa les principes de la cryptographie moderne essentiellement:

La sécurité d'un crypto système ne doit pas reposer sur le secret de l'algorithme de codage mais qu'elle doit uniquement reposer sur la clef secrète qui est un paramètre facile à changer, de taille réduite et facile à transmettre secrètement

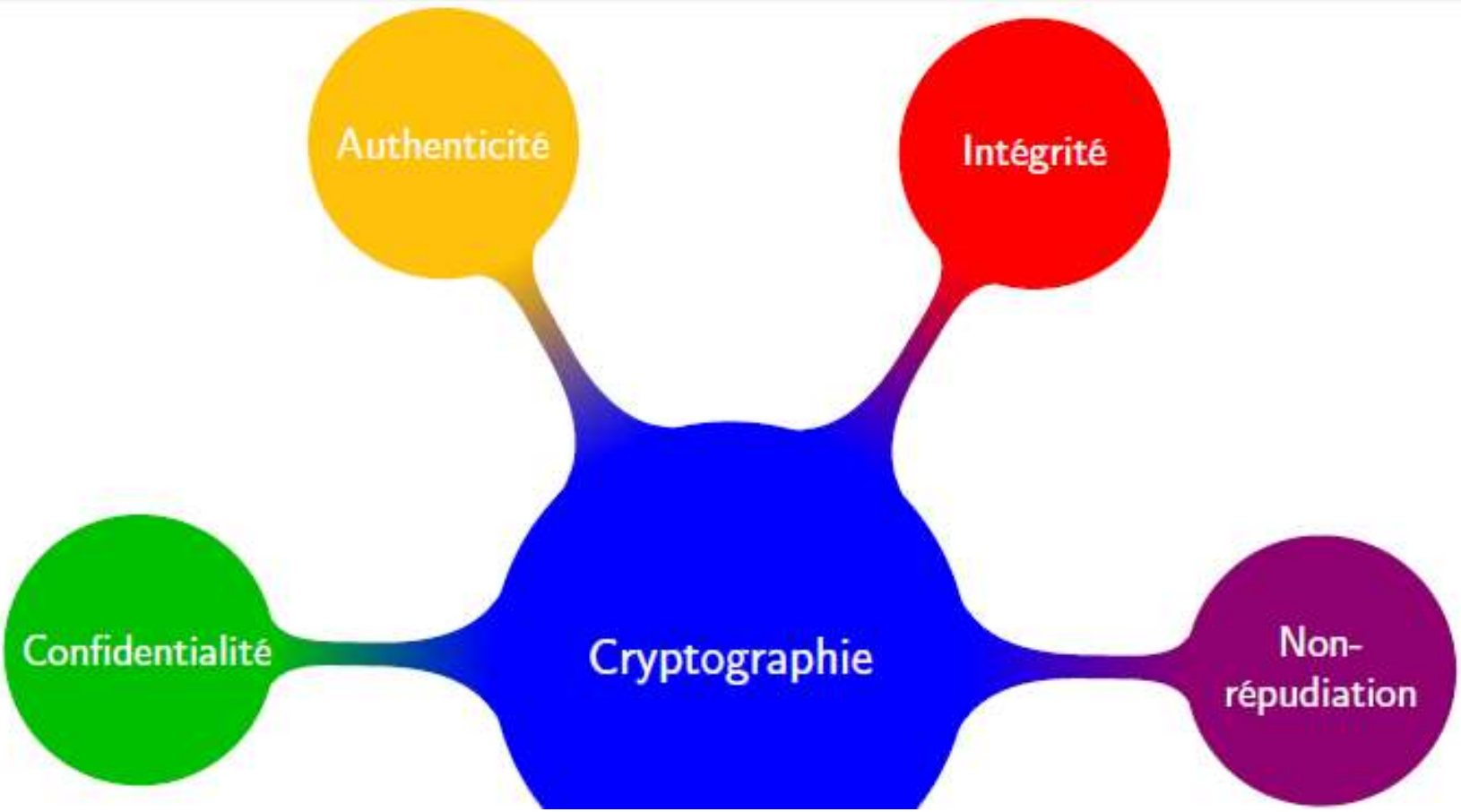
- Ce principe a été toujours respecté par les standards internationaux y compris conçus par **NIST** (National Institute of Standards an Technology)

- **Fondements:**

  - Un crypto système sera d'autant plus résistant et sûr qu'il aura été conçu, choisi et implémenté avec la plus grande transparence et soumis ainsi à l'analyse de l'ensemble de la communauté cryptographique.

  - Si un algorithme est suppose être secret, il se trouvera toujours quelqu'un soit pour vendre l'algorithme, soit pour en découvrir une faiblesse ignorée de ses concepteurs. A ce moment l`a c'est tout le crypto système qui est à changer et pas seulement la clé.

PROPRIÉTÉS



On peut les ajouter une cinquième propriété: **La disponibilité**

Confidentialité

Le fait de s'assurer que l'information n'est seulement accessible qu'à ceux dont l'accès est autorisé

Authenticité

Le fait de s'assurer que l'expéditeur est bien celui qu'il prétend être

- **Authentification:**

– l'émetteur est sûr de l'identité du destinataire c'est `a dire que seul le destinataire pourra prendre connaissance du message car il est le seul à disposer de la clef de déchiffrement.
– le receveur est sûr de l'identité de l'émetteur

Intégrité

Le fait de s'assurer que l'information ne subisse aucune altération ou destruction volontaire ou accidentelle, et conserve le format initial

Non-répudiation

d'origin

L'émetteur ne peut nier avoir écrit le message et il peut prouver qu'il ne l'a pas fait si c'est le cas

de réception

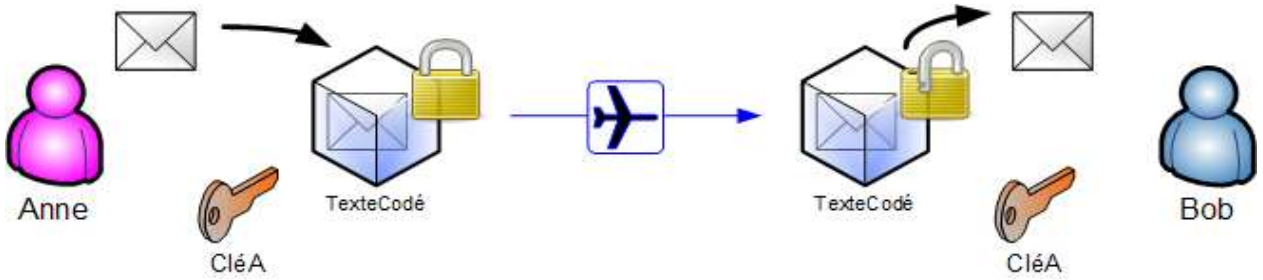
Le receveur ne peut nier avoir reçu le message et il peut prouver qu'il ne l'a pas reçu si c'est effectivement le cas

de transmission

L'émetteur du message ne peut nier avoir envoyé le message et il peut prouver qu'il ne l'a pas fait si c'est le cas

INTRODUCTION AU CHIFFREMENT SYMÉTRIQUE

Le chiffrement symétrique, également appelé **cryptographie à clé symétrique** ou **chiffrement à clé secrète,** est l'une des deux principales méthodes de chiffrement. Il fonctionne en créant une seule clé partagée pour chiffrer et déchiffrer les données sensibles.



- **Son principal avantage**: **simplicité** et **efficacité** pour sécuriser les données.

- **Inconvénients : 1) moins sûr** que le chiffrement asymétrique, principalement parce qu'il repose sur un échange de clés sécurisé et une **gestion rigoureuse des clés**. Toute personne qui intercepte ou obtient la clé symétrique peut accéder aux données.

2) Il faut autant de clefs que de couples de correspondants (2-> 1 clef,...,n-> n(n-1)/2 clefs)
3) La non-répudiation n'est pas assurée. mon correspondant possédant la même clef que moi peut fabriquer un message en usurpant mon identité

- **Choix:** les organisations choisissent généralement le chiffrement symétrique lorsque **l'efficacité est primordiale**, par exemple pour chiffrer **de gros volumes** de données ou sécuriser **des communications internes** dans un système fermé

Si la sécurité est primordiale comme pour chiffrer des données sensibles ou sécuriser les communications au sein d'un système ouvert → plutôt l'asymétrique

- **Quelques algorithmes de chiffrement symétrique :**

- **Chiffre de Vernam** (le seul offrant une sécurité théorique absolue, à condition que la clé ait au moins la même longueur que le message, qu'elle ne soit utilisée qu'une seule fois à chiffrer et qu'elle soit totalement aléatoire)
- DES et 3DES
- AES
- IDEA
- RC4 et RC5
- MISTY1

CHIFFREMENT SYMÉTRIQUE PAR BLOCS

Le schéma général de base du chiffrement par blocs symétrique est le suivant:

1. coder l'information source en binaire. On obtient ainsi une chaine composée de 0 et de 1.

2. découper cette chaine en blocs de longueur donnée (par exemple 64 bits ou 128 bits ou 256 bits…).

3. chiffrer un bloc en faisant un OU exclusif (ou XOR) bit à bit avec une clé secrète qui est une suite de 0 et de 1 de même longueur,

XOR est donc l'addition sans retenu en base deux.

$$m = 1001111010001111, \quad n = 1011111000010111$$
$$m \oplus n = 0010000010011000$$

4. déplacer et permuter certains bits du bloc.

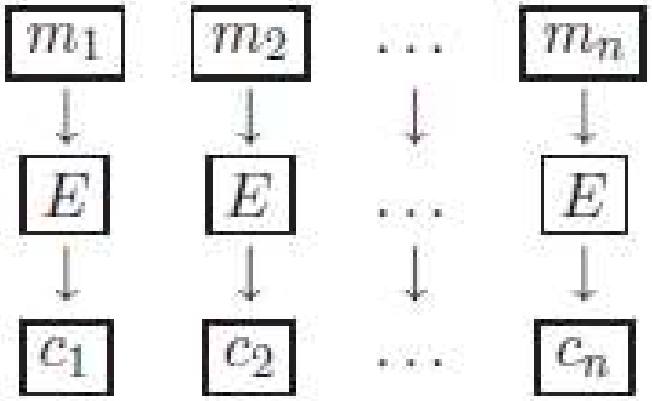
5. recommencer un certain nombre de fois l'étape précédente, on appelle cela **une ronde**.

6. passer au bloc suivant et retourner à l'´étape 3 jusqu'`a ce que tous les blocs soient chiffres.

- Plusieurs modes de chiffrement par blocs existent, certains sont plus vulnérables que d'autres :

- Dictionnaire de codes (Electronic Code Book, ECB)
- Enchaînement des blocs (Cipher Block Chaining, CBC)
- Chiffrement à rétroaction (Cipher Feedback, CFB)
- Chiffrement à rétroaction de sortie (Output Feedback, OFB)
- Chiffrement basé sur un compteur (CounTeR, CTR)
- Chiffrement avec vol de texte (CipherText Stealing, CTS)
- Compteur avec CBC-MAC
- EAX (inventé par David Wagner et al.)
- CWC (à deux passes)

Mode ECB: chaque bloc est crypté séparément



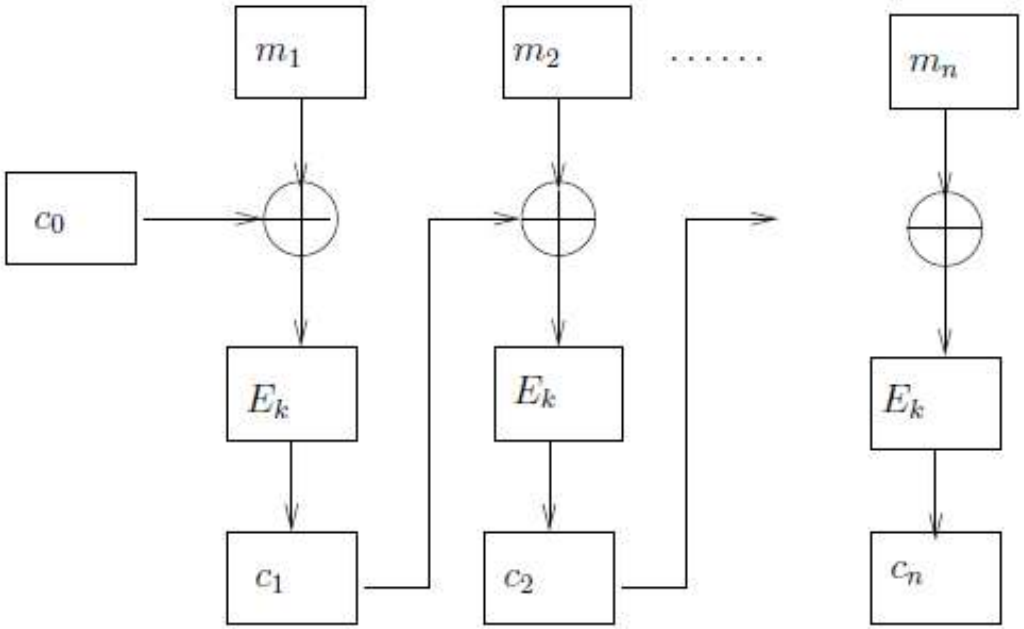
E = algorithme de chiffrement en fonction de k

$$E = E_k$$

Sécurité du mode ECB

- Un bloc de message m_i sera toujours codé de la même manière, ce qui nuit à la sécurité du codage.
- un attaquant malveillant peut permuter deux blocs ou remplacer un bloc par un autre sans que le destinataire s'en aperçoive.

Mode CBC a été introduit pour qu'un bloc ne soit pas codé de la même manière s'il apparait dans deux messages différents ou s'il apparait deux fois dans un message.



Le bloc initial (IV) c_0 peut être choisi de l'une des maniérés suivantes:

1. On génère c_0 aléatoirement et on le transmet en clair avec le message ou de manière confidentielle
2. On utilise un c_0 fixe qui fait partie des constantes du crypto système ou de la clé sécrète du crypto système

La première est la plus conseillée

Déchiffrement

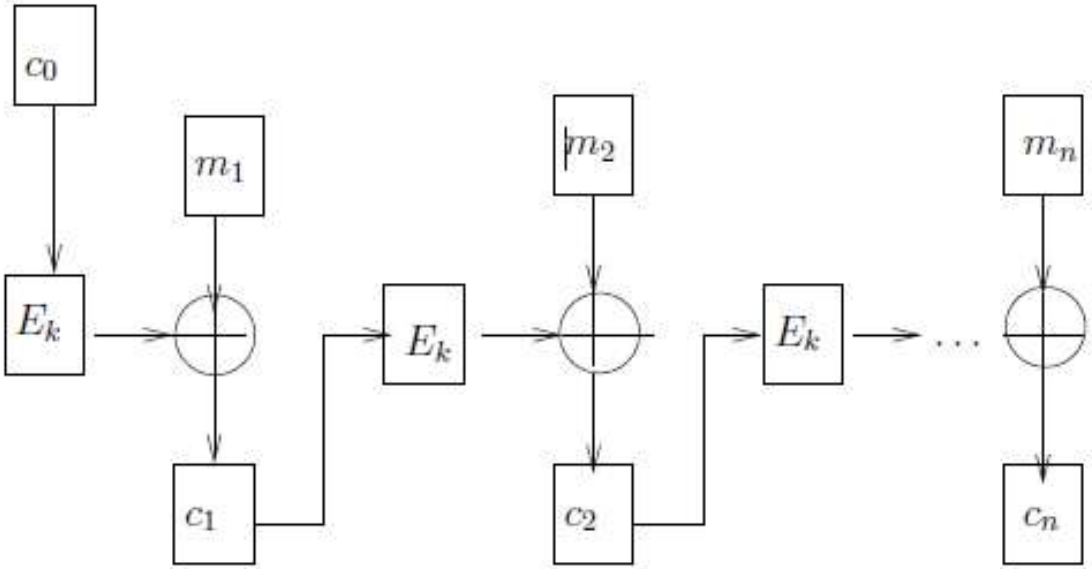
$$D_k = E_k^{-1}$$

$$m_i = c_{i-1} \oplus D_k(c_i)$$

Sécurité du mode CBC

Ce mode a une bonne sécurité et n'affaiblit pas le cryptosystème, mais il **nécessite de connaitre la fonction inverse D_k de E_k.**

Un mode similaire a été introduit pour ne pas avoir à calculer la fonction inverse, D_k, de la fonction de chiffrement E_k.



$$c_0 = E_k(m_0)$$
$$c_1 = m_1 \oplus E_k(c_0)$$
$$c_2 = m_2 \oplus E_k(c_1)$$
$$\vdots$$

Sécurité:

Ce mode est moins sûr que le CBC. L'intérêt est que le déchiffrement **ne nécessite pas** de calculer D_k, en effet:

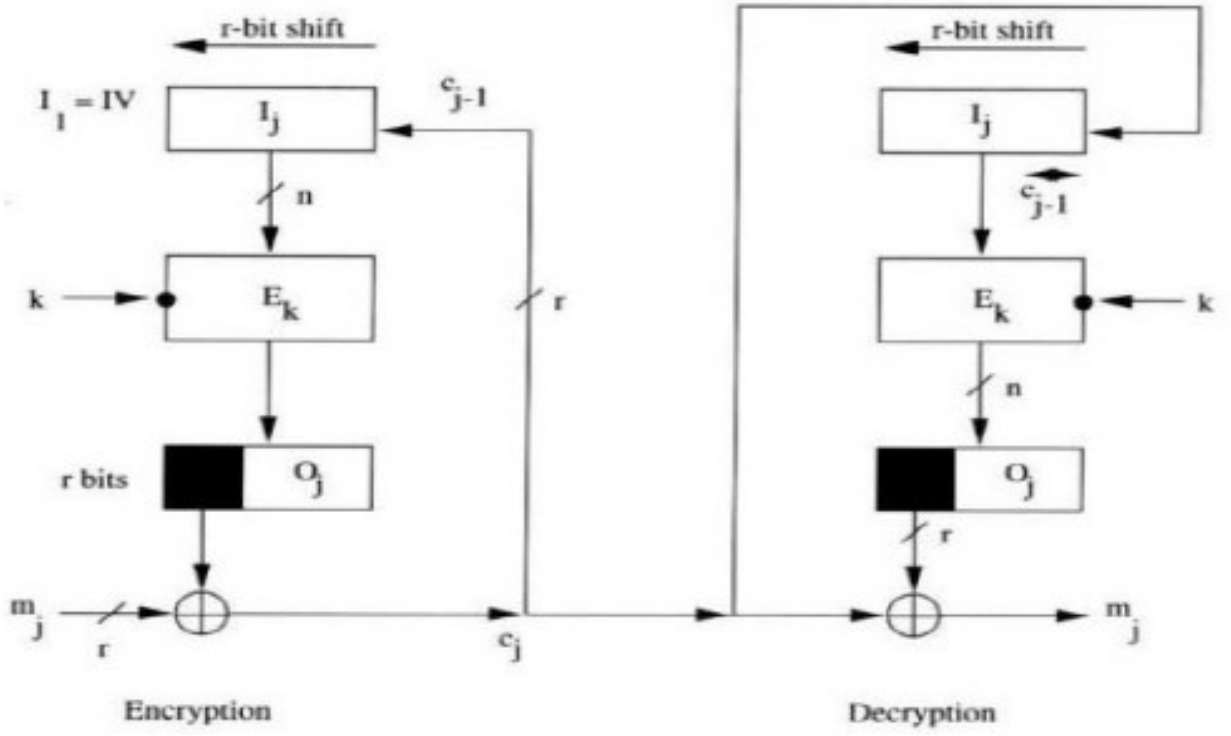
$$m_i = c_i \oplus E_k(c_{i-1})$$

Mode CFB:

- Avec les modes ECB ou CBC, le chiffrement ne peut pas commencer avant qu'un bloc complet de données n'ait été reçu, ce qui peut poser un problème dans certaines applications réseaux, où les données à traiter se présentent sous forme de paquets de la taille de l'octet.
- Le mode de chiffrement à rétroaction, en anglais "Cipher Feedback"(CFB), permet de chiffrer les données par unités plus petites que la taille de bloc.
- On appelle mode de **chiffrement à rétroaction à r bits** le mode CFB qui chiffre les données par unités de r bits. Souvent on chiffre un caractère ASCII à la fois, c'est-à-dire r = 8,
- r peut en principe prendre n'importe quelle valeur, pourvu qu'elle ne dépasse pas la taille d'un bloc.

En mode CFB, on manipule un registre de décalage de la taille d'un bloc du message en clair.

1) Pour commencer, le registre de décalage est rempli par un vecteur d'initialisation.
2) On chiffre ensuite le registre de décalage avec l'algorithme de chiffrement utilisé
3) On combine les r bits les plus significatifs, c'est-à-dire les r bits les plus à gauche du résultat par un ou exclusif avec les r premiers bits du message en clair pour obtenir ainsi les r premiers bits du message chiffré qu'on peut alors transmettre.
4) On place ce bloc de message chiffré dans les r bits les plus à droite du registre de décalage et on décale tous les autres bits du registre de décalage de r positions vers la gauche en coupant ses r premiers bits.
5) On continue alors de chiffrer le reste du message en clair de la même manière.



En mode CFB, le déchiffrement se fait en appliquant de nouveau la fonction de chiffrement de l'algorithme au même registre de décalage que pour le chiffrement et en ajoutant les *r* premiers bits de ce résultat au bloc précédent du Message chiffré.

Sécurité du mode CFB

- Le mode CFB offre une grande sécurité.
- Le seul problème est que le message en clair est seulement soumis à un XOR. On peut inverser les bits du message chiffré au même endroit où on inverse les bits du message en clair (bit-flipping attack).

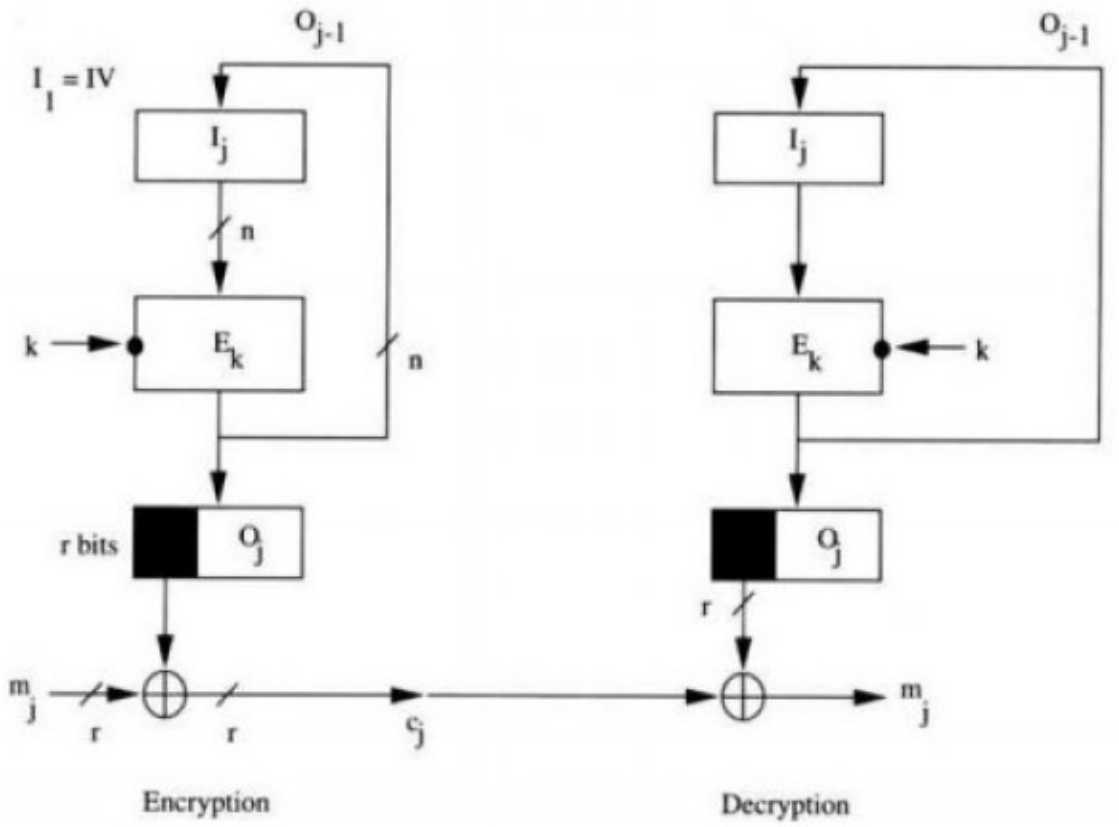
Propagation d'erreurs

- Une erreur dans le message chiffré d'un seul bit provoque une erreur d'un seul bit dans le bloc de message en clair correspondant. Puis, elle entre dans le registre à décalage, où elle embrouille tout jusqu'elle en ressorte.

Après cela, le système se récupère et les blocs suivants de message en clair sont déchiffrés correctement.

- Le mode CFB est également **auto réparateur** pour ce qui est des erreurs de synchronisation. L'erreur entre dans le registre à décalage, embrouille un bloc de données, puis ressort à l'autre bout après quoi tout fonctionne à nouveau correctement.

Mode OFB: Le mode de rétroaction de sortie, en anglais "Output Feedback" (OFB), ressemble beaucoup au mode CFB, sauf que ce sont les r premiers bits du résultat du chiffrement du registre de décalage qui sont ajoutés à droite du registre de décalage et non les r premiers bits du texte chiffré.



- Cette méthode est parfois appelée rétroaction interne, parce que le mécanisme de rétroaction est indépendant à la fois des flots de message en clair et de message chiffré.
- Par construction, le chiffrement des blocs, qui est le travail lourd, peut donc se faire avant qu'on connaisse le message en clair. Ceci permet de chiffrer respectivement déchiffrer **plus rapidement** en utilisant le mode OFB.

Sécurité du mode OFB

- A partir d'un certain moment le registre de décalage devient constant, ce qui engendre de graves problèmes de sécurité. Pour éviter ce problème, il faut toujours que la taille de rétroaction soit égale à la taille des blocs du message que manipule l'algorithme utilisé.
- Comme dans le mode CFB, le message en clair est seulement soumis à un XOR. Le mode OFB est donc également vulnérable à une "bit-flipping attack".

Propagation d'erreurs

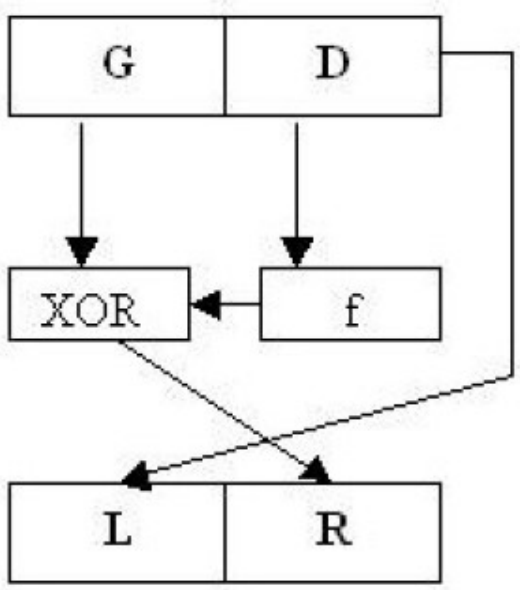
- L'avantage du mode OFB est qu'il n'y a pas d'amplification d'erreur. Une erreur d'un seul bit dans le message chiffré occasionne une erreur d'un seul bit dans le message en clair récupéré.
- La perte de synchronisation ne peut par contre pas être récupérée. Si les contenus des registres à décalage de chiffrement respectivement de déchiffrement ne sont pas identiques, alors le message en clair récupéré sera complètement embrouillé.
- Un système qui utilise le mode OFB doit donc avoir un mécanisme de détection de perte de synchronisation et de resynchronisation.

Quelques définitions/techniques utiles:

- Substitution : Les substitutions consistent à remplacer des symboles ou des groupes de symboles par d'autres symboles ou groupes de symboles dans le but de créer de la confusion.
- Transposition : Les transpositions consistent à mélanger les symboles ou les groupes de symboles d'un message clair suivant des règles prédéfinies pour créer de la diffusion.
- Produit : La substitution combinée à la transposition afin d'avoir un chiffrement plus robuste. Un « round » est complété lorsque les deux transformations ont été faites une fois (substitution et transposition)

SCHÉMAS DE FEISTEL

Cette structure fut décrite en 1973 par Feistel, employé chez IBM. La plupart des chiffrements de la fin du XXè siècle sont basés sur cette structure



- Le bloc d'entrée d'un round est séparé en deux parties. La fonction de chiffrement (dépend d'une clé ou sous-clé) est appliquée sur la partie à droite du bloc et l'opération binaire OU-Exclusif est appliquée sur la partie sortante de la fonction et la partie à gauche. Ensuite les deux parties sont permutées et le prochain round commence.
- Cette transformation est **bijective**, car si on a un couple (L,R), on retrouve bien (G,D) par:

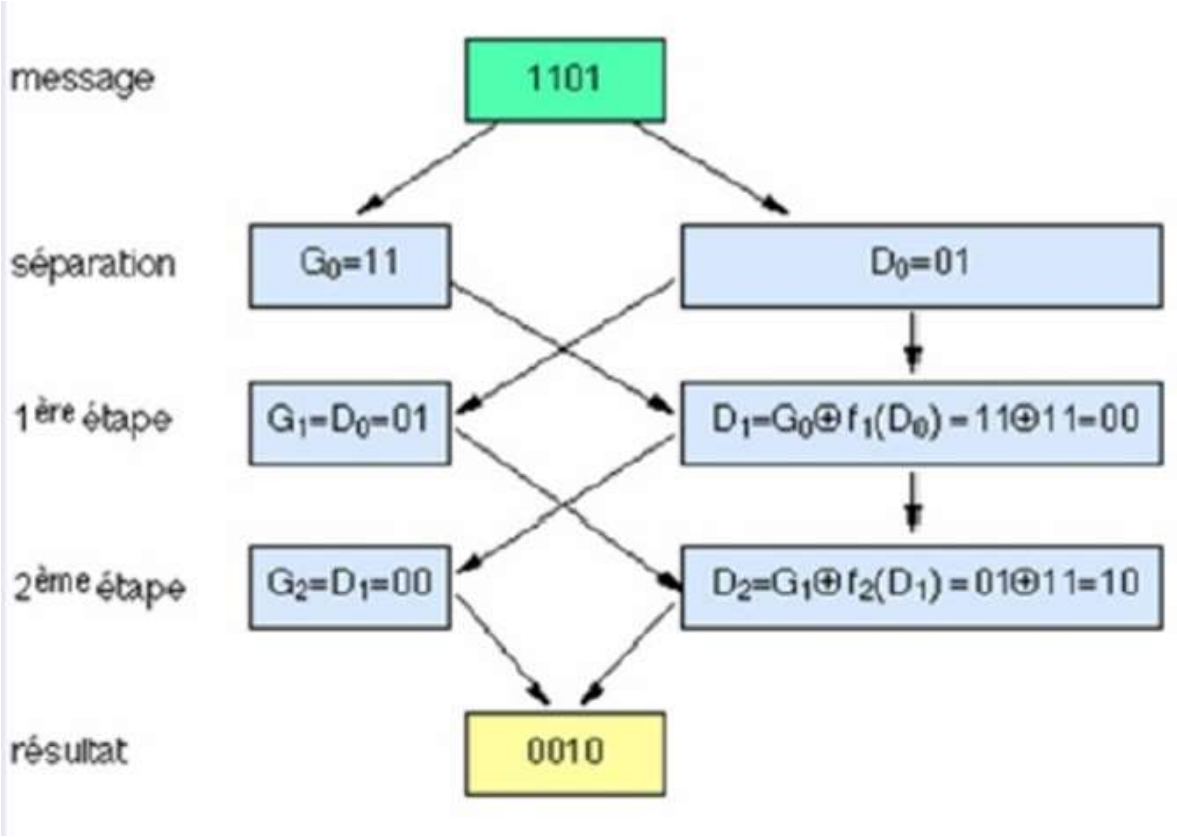
$$L=D \text{ et } R = G \oplus f(D)$$

L'avantage principal est que le chiffrement et le déchiffrement sont structurellement identiques

Exemple:

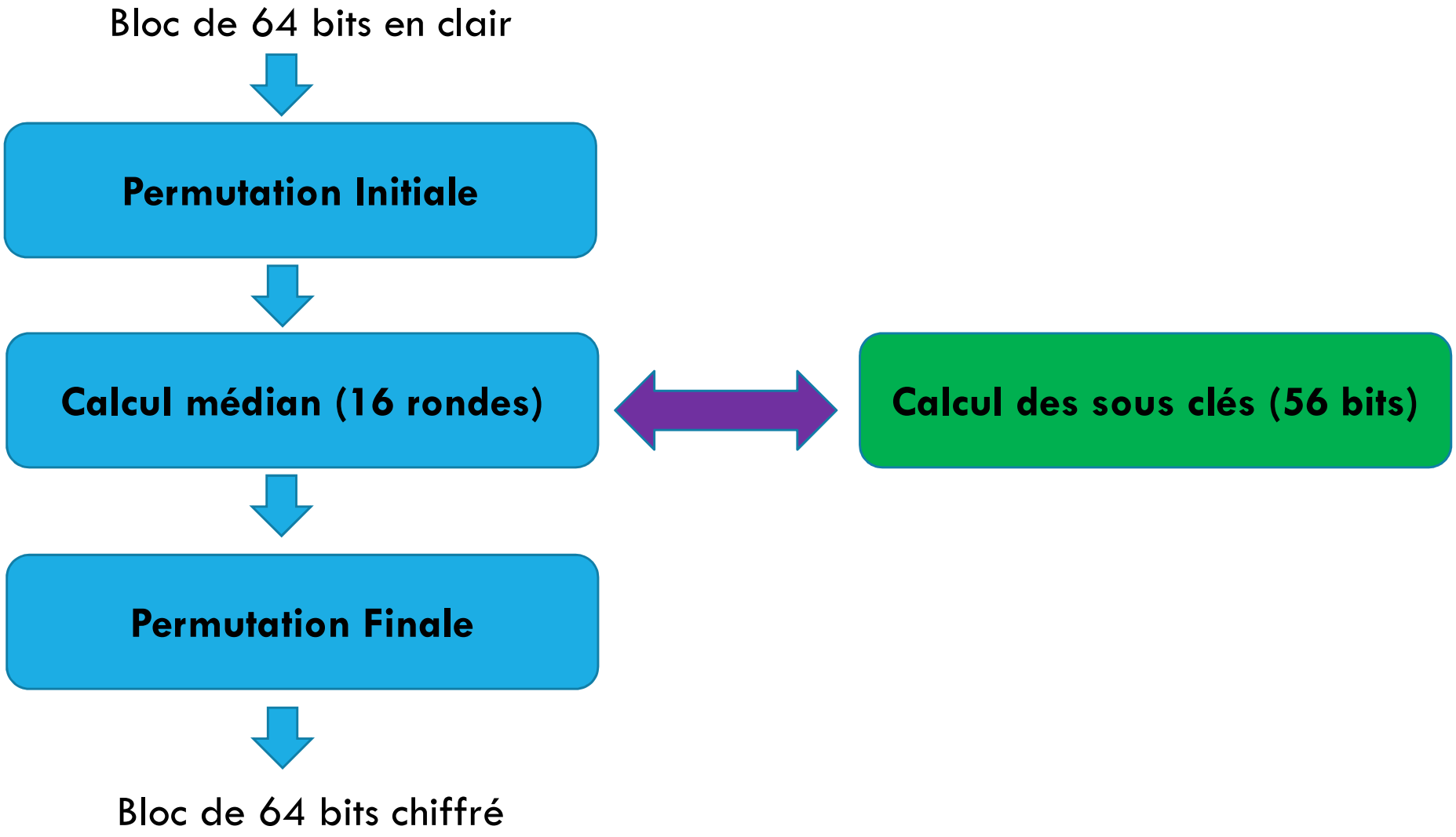
Entree	f1	Sortie
00	→	00
01	→	11
10	→	10
11	→	01

Entree	f2	Sortie
00	→	11
01	→	00
10	→	10
11	→	01



CRYPTO SYSTÈME DES (DATA ENCRYPTION STANDARD)

- La sécurité de tout système Feistel dépend de plusieurs paramètres :
  - Taille du bloc : si elle augmente, la sécurité augmente également
  - Taille de clef : si elle augmente, la sécurité aussi
  - Nombre de rondes : plus il y en a, plus la sécurité est renforcée
  - Algorithme de génération des sous-clés : plus il est complexe, plus la compréhension est rendue difficile.
- DES est une implémentation du chiffrement de Feistel depuis 1977, officiel à l'administration américaine jusqu'à 1999
- Le développement des communications entre ordinateurs a nécessité la mise en place d'un standard de chiffrement de données (DES) pour limiter la prolifération d'algorithmes différents ne pouvant pas communiquer entre eux.
- DES est rapide, exportable, facile à implémenter et repose sur une clef relativement petite, qui sert à la fois au chiffrement et au déchiffrement


Bloc de 64 bits en clair
Permutation Initiale
Calcul médian (16 rondes)
Calcul des sous clés (56 bits)
Permutation Finale
Bloc de 64 bits chiffré

Permutation initiale (PI ou IP)

Les 64 bits du bloc d'entrée subissent la permutation suivante:

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

Exemple: Le premier bit de sortie sera le bit 58, le second le bit 50…

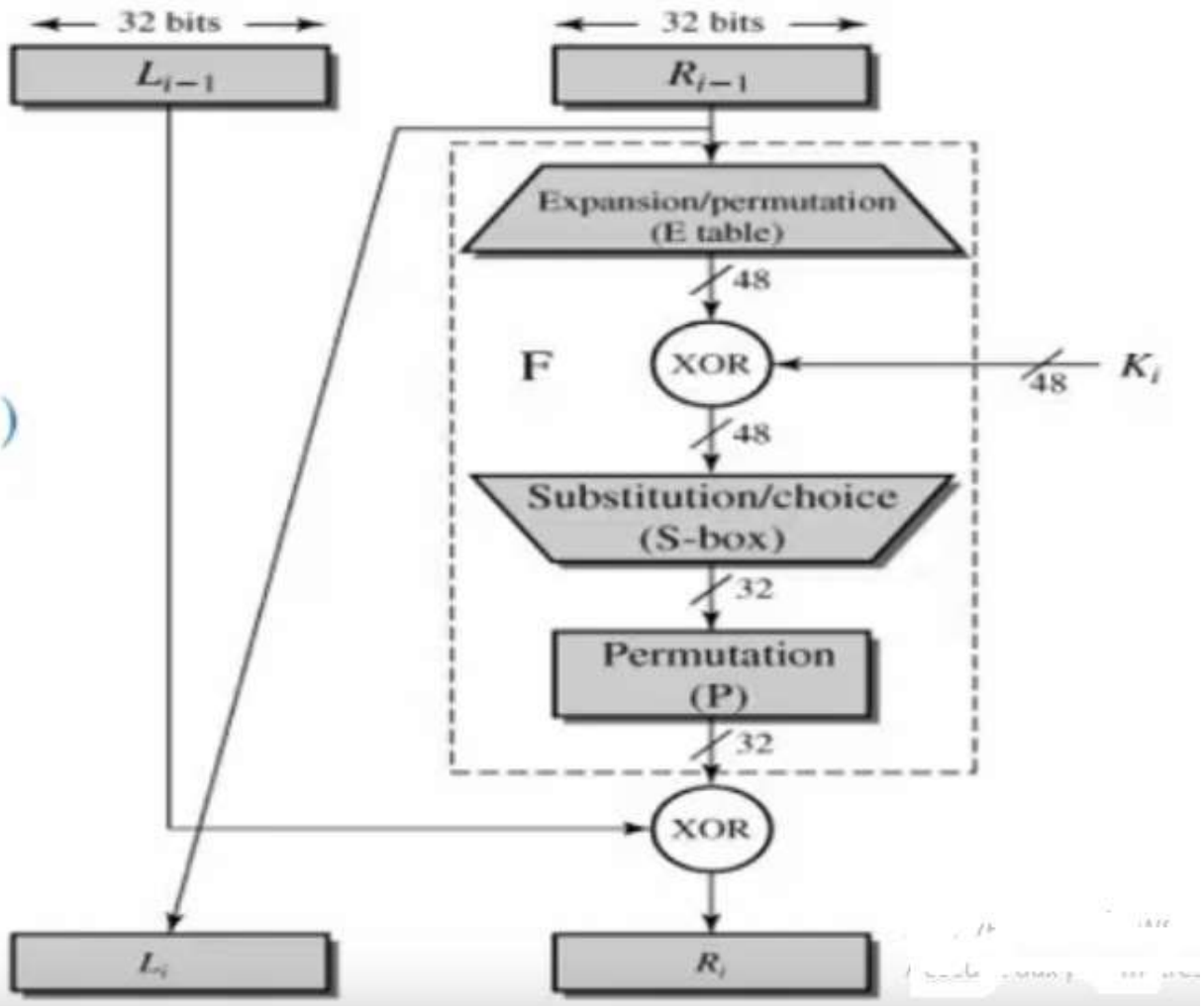
Permutation finale (PF ou FP)

Les 64 bits des blocs d'entrée subissent la permutation suivante:

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

Une ronde du calcul médian

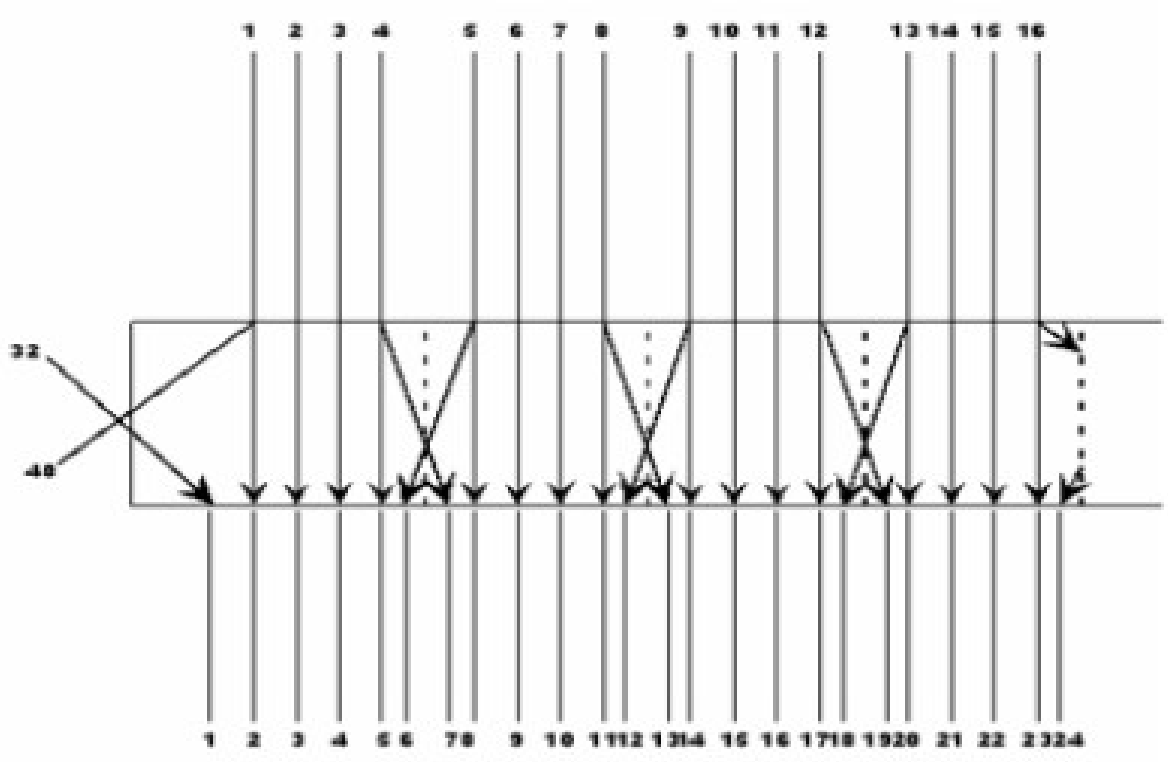
- $L_i = R_{i-1}$
- $R_i = L_{i-1} \oplus F(R_{i-1}, K_i)$
- $K_i = G(K, i)$



Expansion-Permutation

Les 32 bits sont étendus à 48 bits grâce à une table d'expansion (également appelée matrice d'expansion ou table E)

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



Addition à la sous-clé K_i

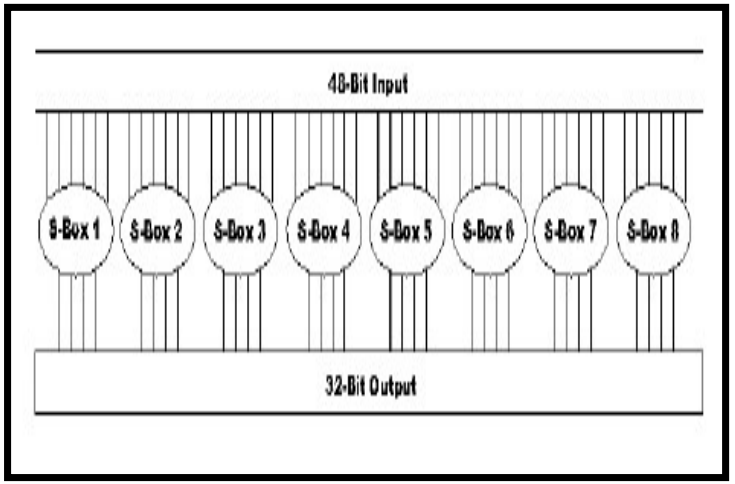
Le résultat de l'expansion est additionné (par XOR) à la sous-clé K_i de longueur 48 bits correspondant à l'itération selon la formule en donnant 8 sous-blocs B_i de taille 6 bits:

$$E(R_{i-1}) \oplus K_i = B_1 B_2 \dots B_8$$

$$B_j = b_1 b_2 b_3 b_4 b_5 b_6$$

Transformations par S-Boxes

Chaque bloc B_i constitue ensuite l'entrée de l'opération de substitution réalisée sur base des S-Box.



row	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13
0	15	1	2	14	6	11	3	4	9	7	2	13	2	0	5	10
1	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
2	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
3	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9
0	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
1	13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
2	13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
3	1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12
0	7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
1	13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
2	10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
3	3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14
0	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
1	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
2	4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
3	11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3
0	12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
1	10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
2	9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6
3	4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13
0	4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
1	13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
2	1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
3	6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12
0	13	2	8	4	6	15	11	1	10	9	3	14	5	0	122	7
1	1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
2	7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
3	2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

Pour chaque S-Box on à déterminer:

b_1b_6: numéro de la ligne

b_2b_3 b_4b_5 : numéro de la colonne

Puis effectuer la substitution par la valeur de la case correspondante en binaire.

N° de colonne

	0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

N° de ligne

Transformations par P-Box

L'opération de permutation est réalisée sur le résultat de la substitution des S-box et est basée sur cette matrice:

P – Box Table															
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

Calcul de la sous-clé K_i

La clé est constituée de 64 bits dont 56 sont utilisés dans l'algorithme de calcul des sous-clés. Les 8 autres sont les bits de parité des 7 groupes de 8 bits.

1- Permutation sur les 56 bits (sans les bits de parité):

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

2- Division en deux sous clés de 28 bits chacune.

3- A chaque itération, chaque sous-clé de 28 bits subit une rotation d'1 ou 2 bits vers la gauche selon:

Round	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Key bit shifted	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

4- Réduction : après concaténation des deux sous-clés précédentes, la clé résultante (56 bits) est réduite à une sous-clé de 48 bits sur base de la matrice de réduction suivante D-Box:

14	17	11	24	1	5	3	28	15	6	21	10
23	19	12	4	26	8	16	7	27	20	13	2
41	52	31	37	47	55	30	40	51	45	33	48
44	49	39	56	34	53	46	42	50	36	29	32

Il y'a eu suppression des bits: 9,18,22,25,35,38,43,54

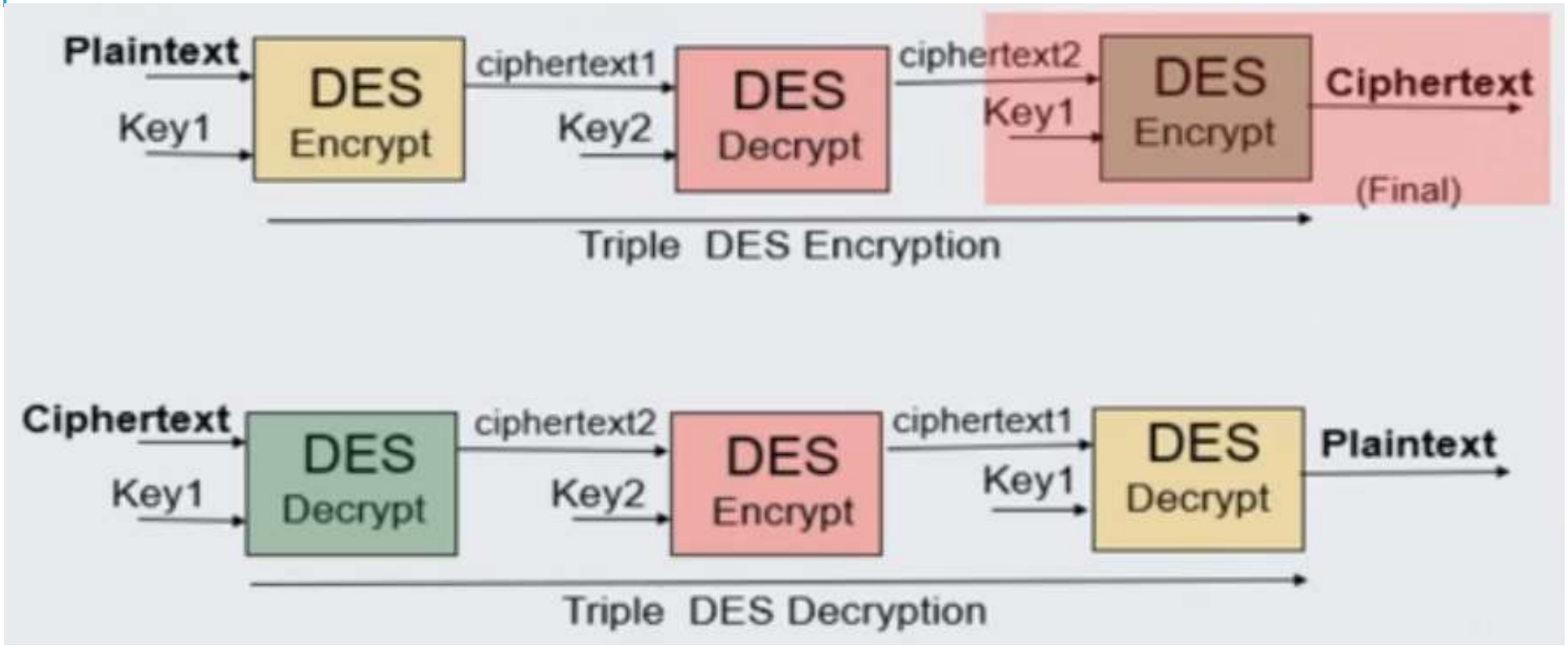
Sécurité DES

- DES sur la diffusion et le confusion (qui est contre toute attaque de cryptanalyse)
- Shift Round + compression créant les 16 sous-clés produit **la confusion** qui obscurat sur les 16 rondes la relation entre la clé d'origine et le cipherText
- S-Box et P-Box sur les 16 rondes (+ mode CBC) produisent **la diffusion** tel qu' un seul bit change sur le plainText la moitié du cipherText change
- Au début DES était incassable car essayer 2^{56} clés est impossible (**attaque Brut Force**). Par exemple, un essai de 1 sec exige 2,3 milliards d'années pour toutes les clés,

En 1997, avec les avancées technologiques 70000 machines connectées sur Internet ont réussi à trouver une clé secrète en 96 jours → **NIST** a lancé un challenge pour remplacer DES

- On a déjà commencé à utilise **3-DES** (Triple DES)
- En fin de 1997 cinq crypto systèmes ont été proposé dont celui des belges **AES** (ADVANCED ENCRYPTION STANDARD) qui a été validé comme le meilleur en 2001

TRIPLE DES

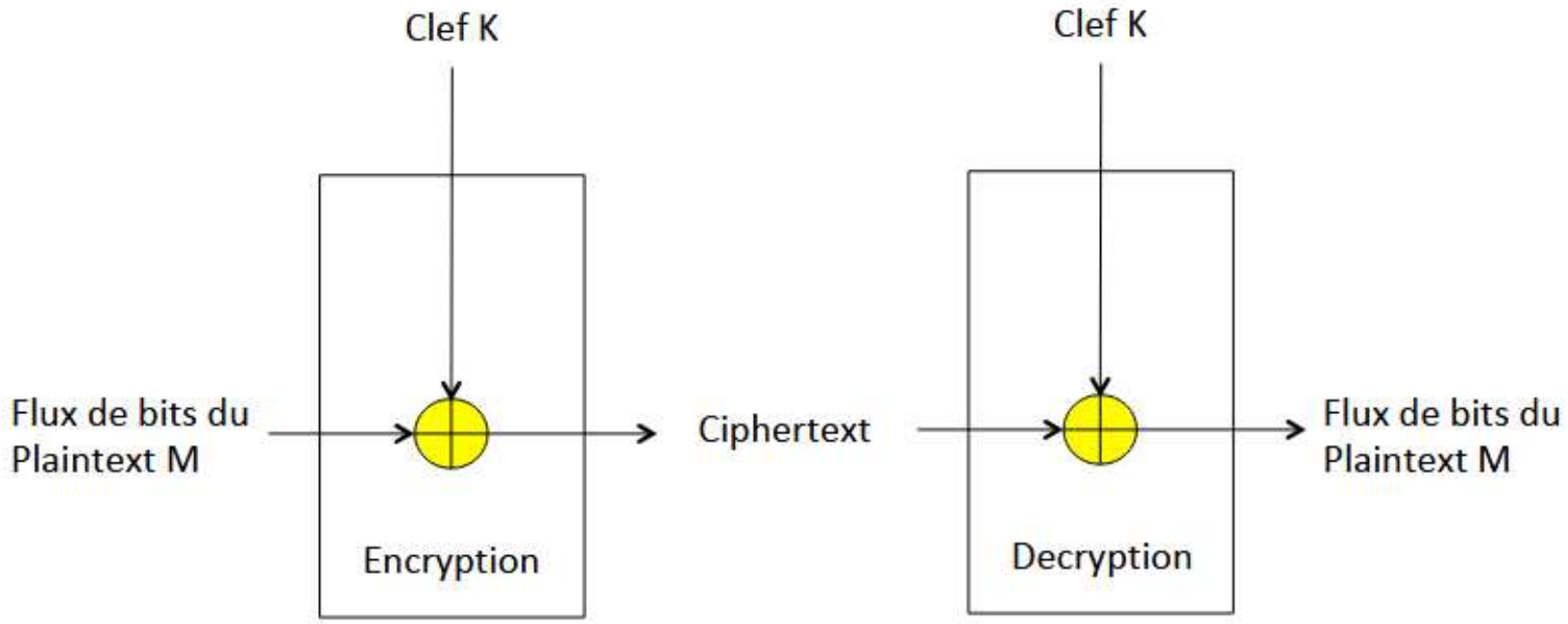


Effectivement (Key1, key2) = 112 bits

USES CASES OF DES

- It is used when encryption of a not-so-high standard is needed.
- It is used for random number generation.
- It is used to develop the Triple DES (a 168-bit key using three keys), which provides a higher amount of security and is approved for use by NIST until 2030.
- Even though it is not currently used for security purposes, DES has provided the basis behind many modern ciphers, such as CAST, Blowfish, IDEA, etc.

CHIFFREMENT SYMÉTRIQUE PAR FLOTS (STREAM CIPHER)



Exemple parfait: Chiffrement de Vernam (1918)

Clef = $k_1k_2k_3k_4$…… (aléatoire, utilisée une seule fois "One-time PAD")

Plaintext = $m_1m_2m_3m_4$……

Chiffré (cipher) : $c_1c_2c_3c_4$….. avec $c_i = m_i \oplus k_i$

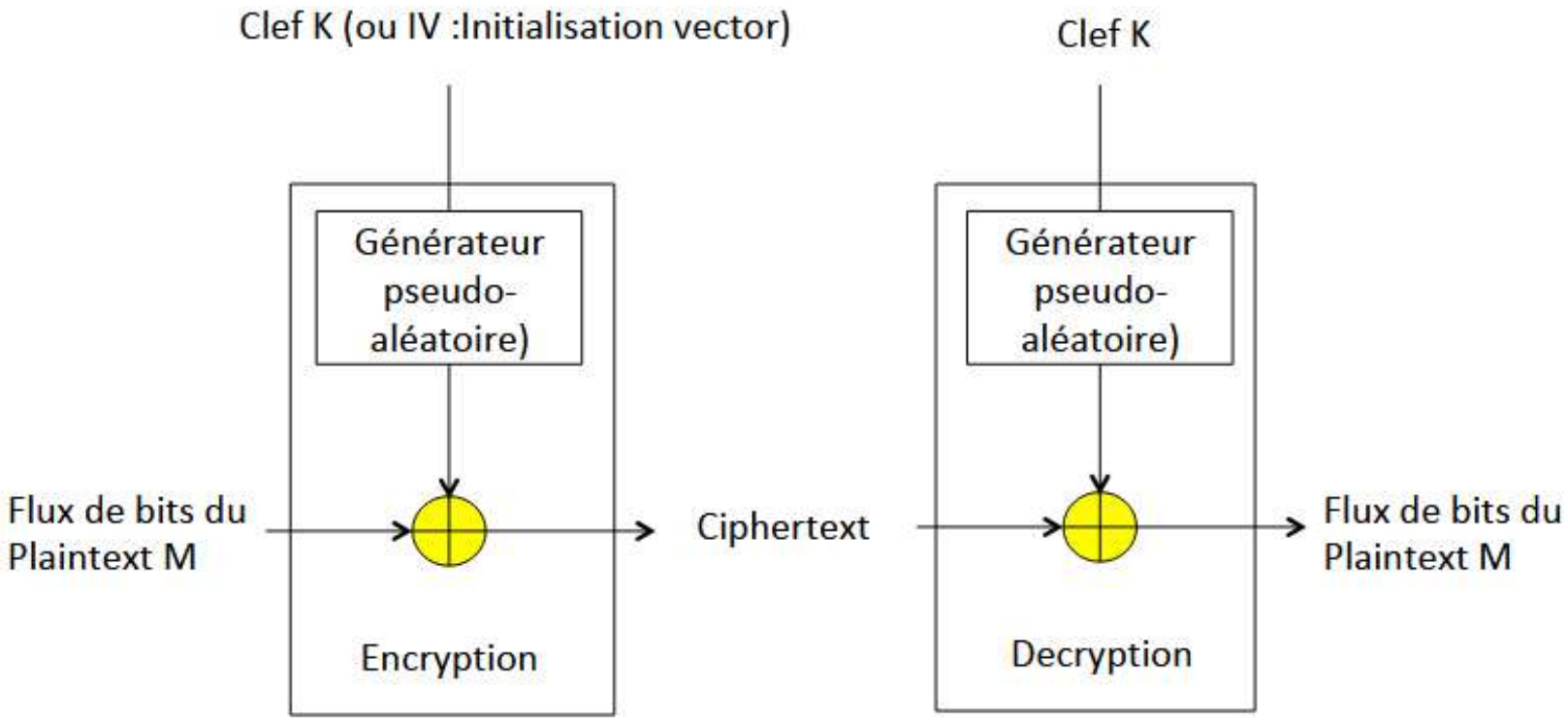
- Il est prouvé **inconditionnellement sûr**, sous la condition que la clef doit être aussi longue que le message (+ aléatoire utilisée une seule fois comme prévu dans le concept de chiffrement)

- **Questions :** partage de la clef (infinie) ? comment générer une telle clef ?

- **En pratique : key-stream**

- longue suite (pseudo aléatoire)
- générée à partir d'une clef secrète courte.

Stream cipher (chiffrement de flux)



Pseudorandom Number Generators (PRNGs): On utilise des algorithmes déterministes pour générer des « nombres aléatoires »

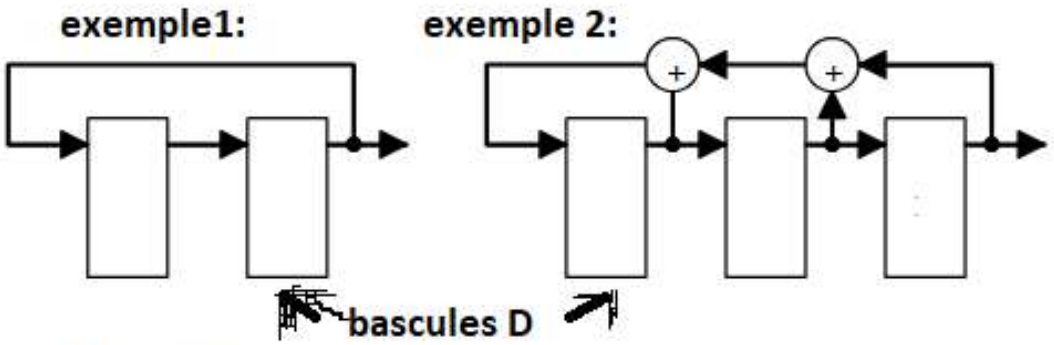
- **Objectif:** des codes par flots qui imitent les codes de Vernam mais qui soient très rapides et très faciles `a mettre en œuvre afin de pouvoir coder et décoder en temps réel (Télévision, radio, DVD, cinéma, téléphones portables et mobiles,...).

- Une famille de tels codes est basée sur les registres à décalage à rétroaction linéaire ou en connues en anglais « linear feedback shift register » en abrégé **LFSR.**

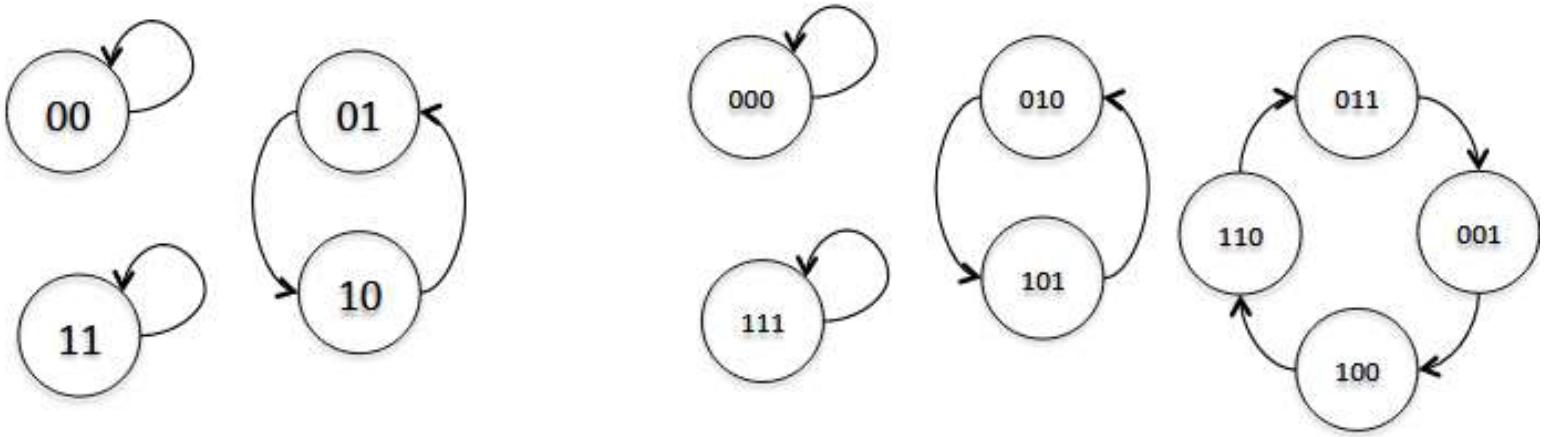
- On fabrique avec les registres à décalage des suites pseudo-aléatoires pour générer la clef du code.

- Ces codes sont beaucoup moins surs que les codes de Vernam mais ne nécessitent pas la fabrication et la transmission d'une clef aléatoire de la longueur du message à transmettre. Ils sont aussi très utilisés pour le codage par flots.

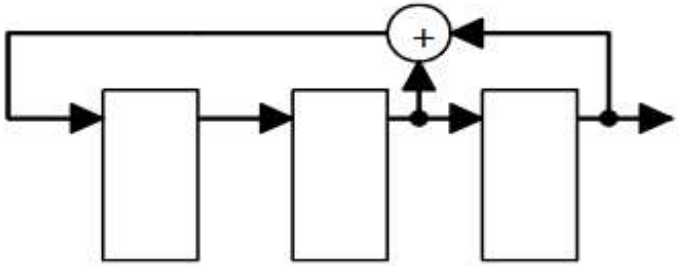
Exemples élémentaires introductifs:



Circuits "cycliques" : parcours d'états cyclique



exemple 3:



S0	0	1	1
S1	0	0	1
S2	1	0	0
S3	0	1	0
S4	1	0	1
S5	1	1	0
S6	1	1	1
S7	0	1	1

Longueur de la séquence = 2^3-1

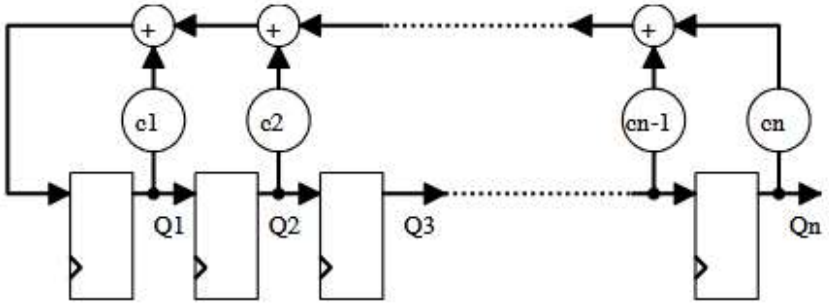
LFSR à séquence maximale

Formes générales des LFSR et types

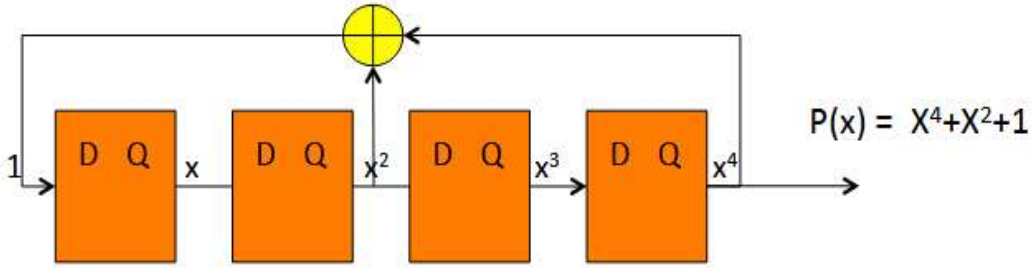
$c_i = 1$: la connexion existe

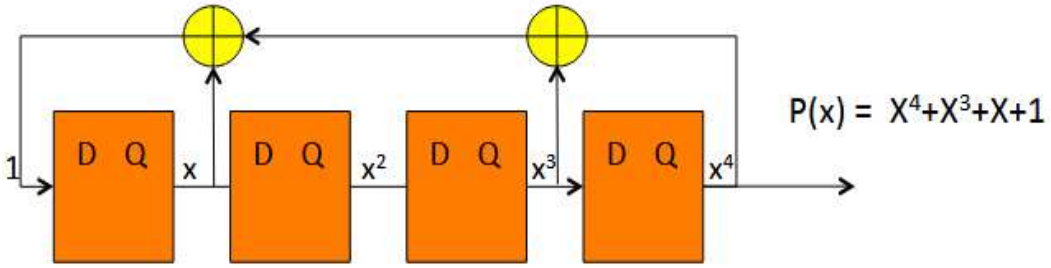
$c_i = 0$: la connexion n'existe pas

Type 1: Fibonnaci

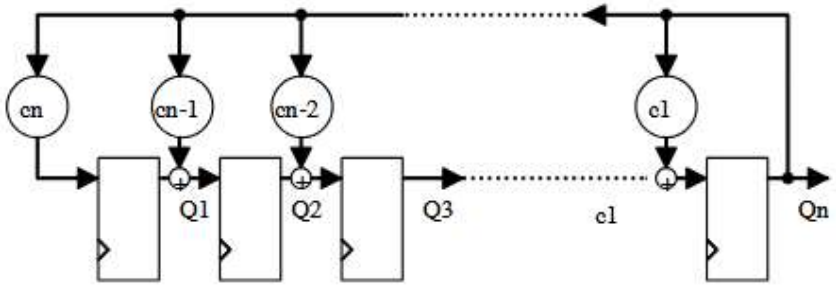


Exemples:

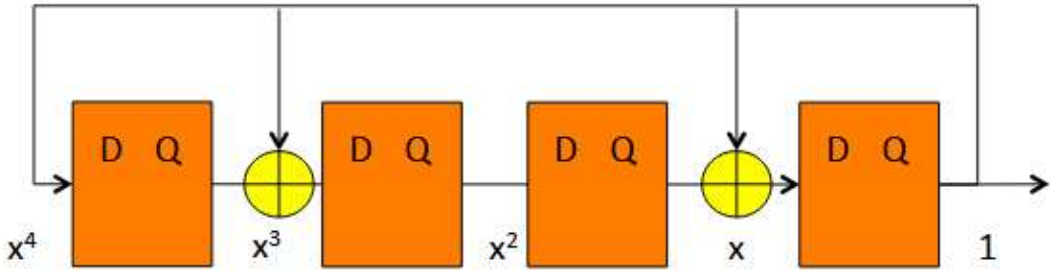




Type 2: Galois



Exemple:

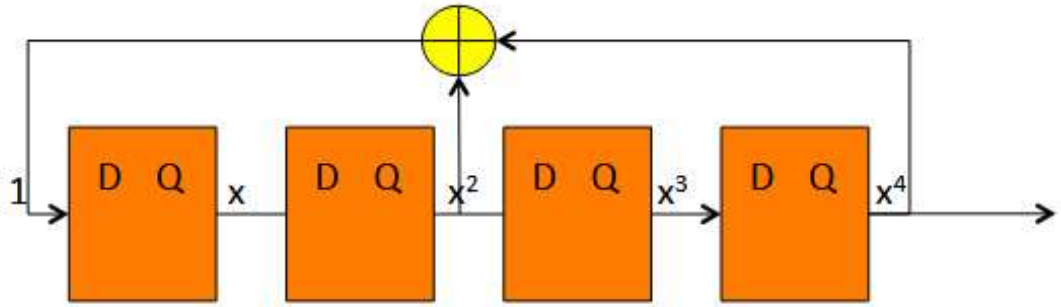


$\mathbf{P(x) = x^4 + x^3 + x + 1}$

Il est à noter que P(x) est appelé **polynôme caractéristique**

$$P(x) = 1 + \sum_{i=1}^{n} c_i . x^i$$

Etude de cas:

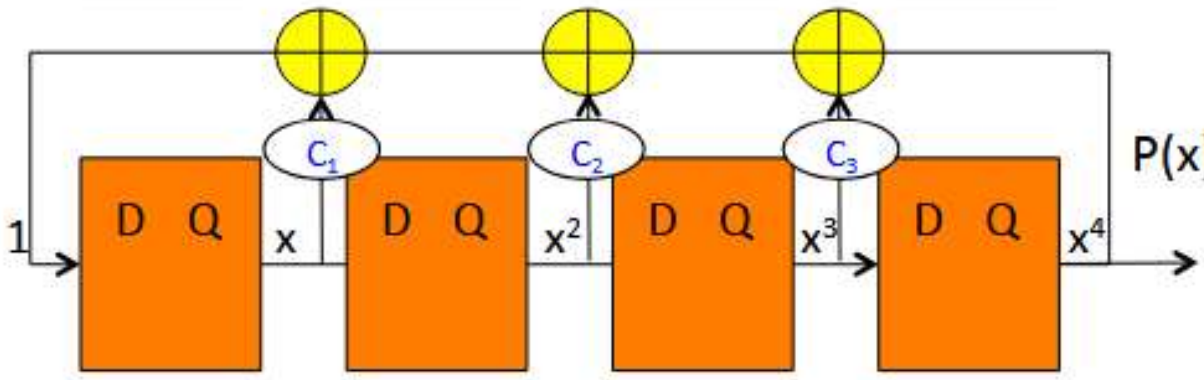


0	0	0	1
1	0	0	0
0	1	0	0
1	0	1	0
0	1	0	1
0	0	1	0
0	0	0	1

Polynôme caractéristique?

Est-il à séquence maximale?

Equivalence:

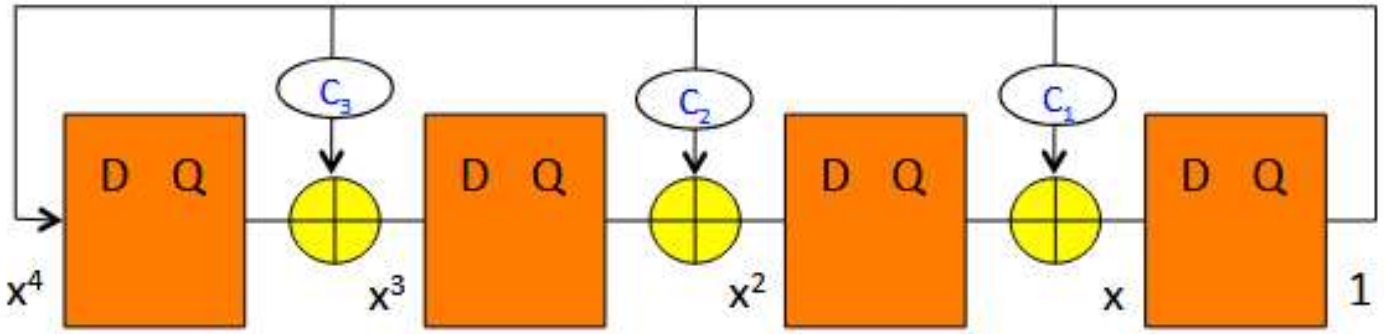


$P(x) = X^4+C_3X^3+C_2X^2+C_1X+1$

C_i=1 la connexion existe

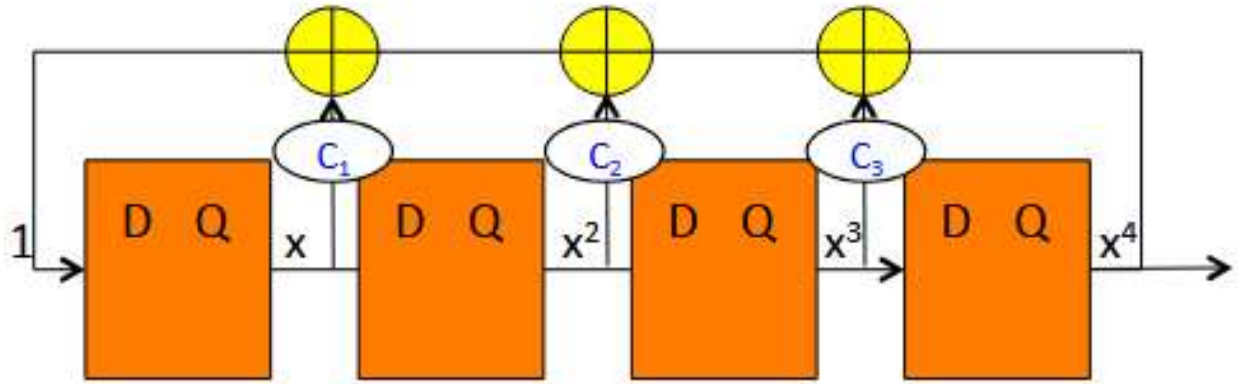
C_i=0 la connexion n'existe pas

LFSR de Fibonacci



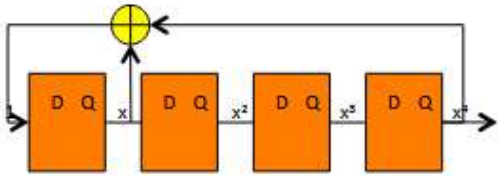
LFSR de Gallois équivalent

Nombre de LFSR possibles:



n FFS => 2^{n-1}-1 LFSR possibles => $P(x) = X^4+C_3X^3+C_2X^2+C_1X+1$

Exemples à comparer

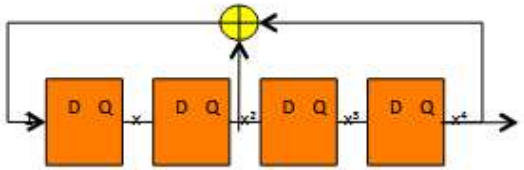


$P(x) = X^4+X+1$

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

2^n-1 valeurs

séquence cyclique de période p=15



$P(x) = X^4+X^2+1$

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

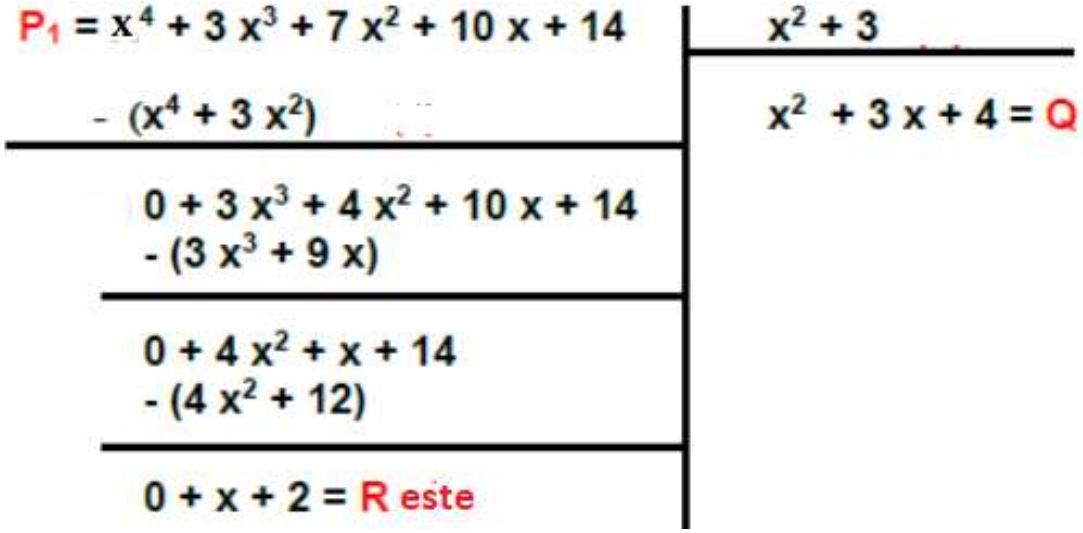
6 valeurs

séquence ,cyclique de période p=6

Différence de polynômes caractéristiques

$P(x) = X^4+X+1$ => Polynôme primitif

Exemple d'explication de la division euclidienne des polynômes



Définition:

Un polynôme **divise** un autre polynôme **si le reste** de leur division euclidienne est égal **à zéro**.

→ Quelques notations, définitions et théorème:

Notation:

$\{a_m\} = a_0, a_1, a_2, \ldots$ représente la séquence de sortie générée par le LFSR avec $a_i = 0$ ou 1

Théorème : Si l'état initial d'un LFSR est $a_{-1} = a_{-2} = \ldots = a_{-(n-1)} = 0$ et $a_{-n} = 1$, alors la séquence $\{a_m\}$ du LFSR est périodique avec une période égale au plus petit entier k tel que P(x) divise $(1-x^k)$.

Définition: Si la séquence générée par un LFSR à n étages est de longueur 2^n-1, elle est appelée séquence de longueur maximum

Définition: Le polynôme caractéristique associé à une séquence de longueur maximum est appelé polynôme primitif

Définition: Un polynôme irréductible est un polynôme qui ne peut être factorise c'est a dire qu'il n'est divisible que par 1 et lui-même.

Théorème: Un polynôme irréductible satisfait les deux conditions suivantes :
- il a un nombre impair de termes (le terme 1 inclus)
- si il est de degré supérieur à 3 alors P(x) doit diviser $(1+x^k)$, avec $k=2^n-1$

Théorème: Un polynôme irréductible est primitif si le plus petit entier positif k tel que P(x) divise $1+x^k$ est tel que $k=2^n-1$ avec n degré du polynôme P(x).