Introduction à la cryptographie – Chapitre II

Systèmes à clef secrète

Plan du cours

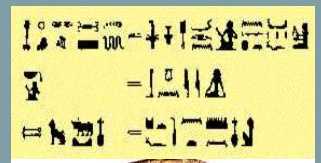
- **0. Courte introduction.**
- **I. Systèmes à clef publique.**
- **II. Systèmes à clef secrète.**
- **III. Authentification.**
- **IV. Exemples.**

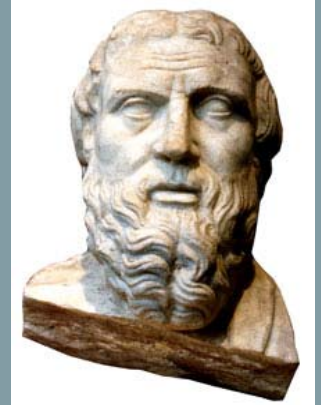
- 2.1. Un peu d’histoire.
- 2.2. Entropie et information.
- 2.3. Le DES.
- 2.4. L’AES.
- 2.5. Autres protocoles.
- 2.6. Conclusion.

- 2.1. Un peu d’histoire.
- 2.2. Entropie et information.
- 2.3. Le DES.
- 2.4. L’AES.
- 2.5. Autres protocoles.
- 2.6. Conclusion.

2.1. Un peu d'histoire.

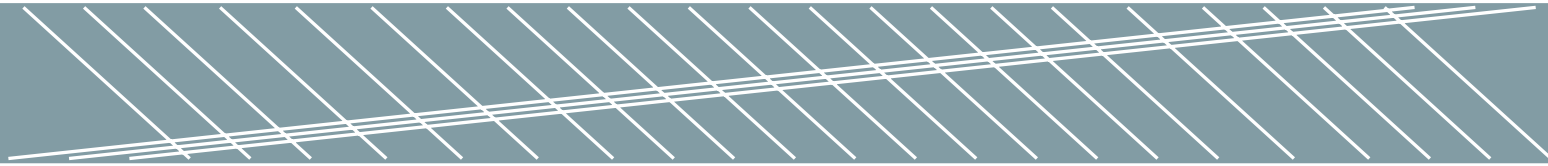
- **- 2000** En Egypte: inscription chiffrée sur la tombe de Khnumhotep.
- **- 1700** En Crête: disque de Phaïstos qui contient une prière chiffrée.
- **- 500** En Grêce: Hérodote raconte que Damatarus, exilé en Perse, prévient les grecs d'une attaque en recouvrant une tablette de cire.
- **- 400** A Sparte: Plutarque parle de la scytale, bâton qui servait à déchiffrer des messages.
- **- 50** Jules César, dans « la guerre des Gaules » présente le chiffrement de César.





- Tous ces systèmes utilisent la **substitution mono-alphabétique**.

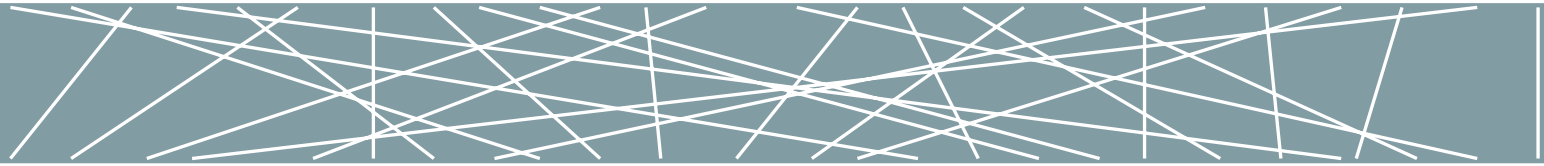
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z



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

- Ou une **permutation mono-alphabétique**.

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

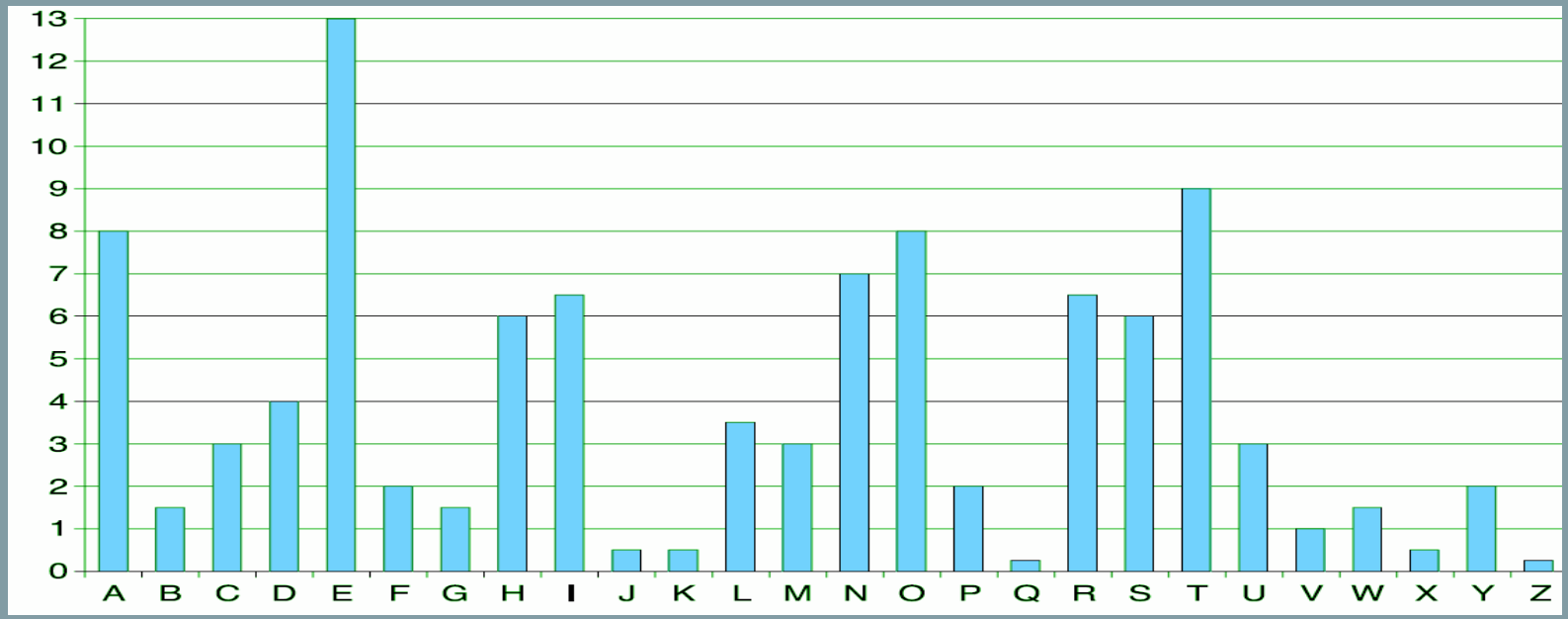


A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

- Pour déchiffrer un tel système, on regarde la fréquence des lettres dans le langage où le message a été écrit.

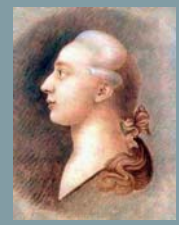
OD#FUBSWRORJLH#HVW#O#DUW#GX#VHFUHW

LA CRYPTOLOGIE EST L ART DU SECRET



- **1586** Vigenère : Le traité des secrètes manières d’écrire.
- **1586** Marie Stuart, reine d’Ecosse, complote à l’aide de messages chiffrés pour assassiner Elisabeth d’Angleterre.
- **1640** John Trevanion, partisan de Charles Ier, est enfermé au château de Colchester par les hommes de Cromwell et reçoit un message : « panel a east of chapel slides ».

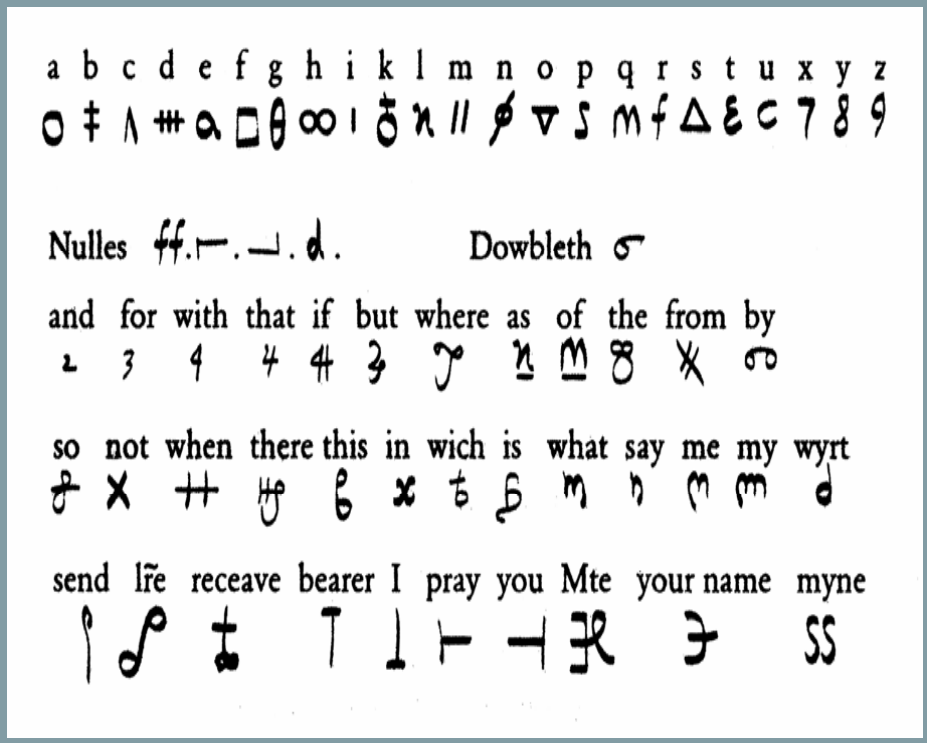
- **1691** Antoine Rossignol : le grand chiffre de Louis XIV.



- **1757** Mémoires de Casanova : « she asked me if I have deciphered the manuscrit ».
- **1830** Correspondance d’Alfred de Musset et George Sand.

- **1839** Edgar Allan Poe : le scarabée d’or.

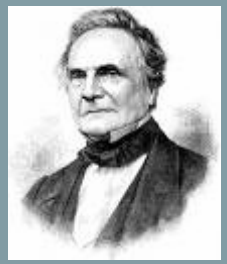
Chiffre de Marie Stuart et lettre de George Sand.



Je suis très émue de vous dire que j'ai
bien compris, l'autre jour, que vous avez
toujours une envie folle de me faire
danser. Je garde un souvenir de votre
baiser et je voudrais que ce soit
là une preuve que je puisse être aimée
par vous. Je suis prête à vous montrer mon
affection toute désintéressée et sans cal-
cul. Si vous voulez me voir ainsi,
dévoiler, sans aucun artifice, mon âme
toute nue, daignez-donc me faire une visite.
et nous causerons en amis et en chemin.
je vous prouverai que je suis la femme
sincère, capable de vous offrir l'affection
la plus profonde et la plus étroite
amitié, en un mot, la meilleure amie
que vous puissiez rêver. Puisque votre
âme est libre, alors que l'abandon où je
vis est bien long, bien dur, et bien souvent
pénible, ami très cher, j'ai le cœur
gros, accourez vite et venez me le
faire oublier. A l'amour, je veux me sou-
mettre.

25/01/2010

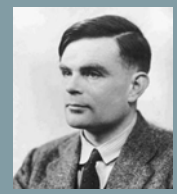
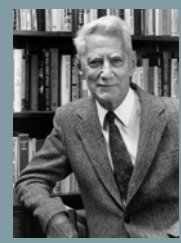
- Blaise de Vigenère invente le système de substituion **poly-alphabétique** en 1586.



- Babbage en effectue la cryptanalyse en 1854, par analyse des fréquences.
- Lorsque la clef est courte et le texte est long, il est très facile d'effectuer la cryptanalyse.

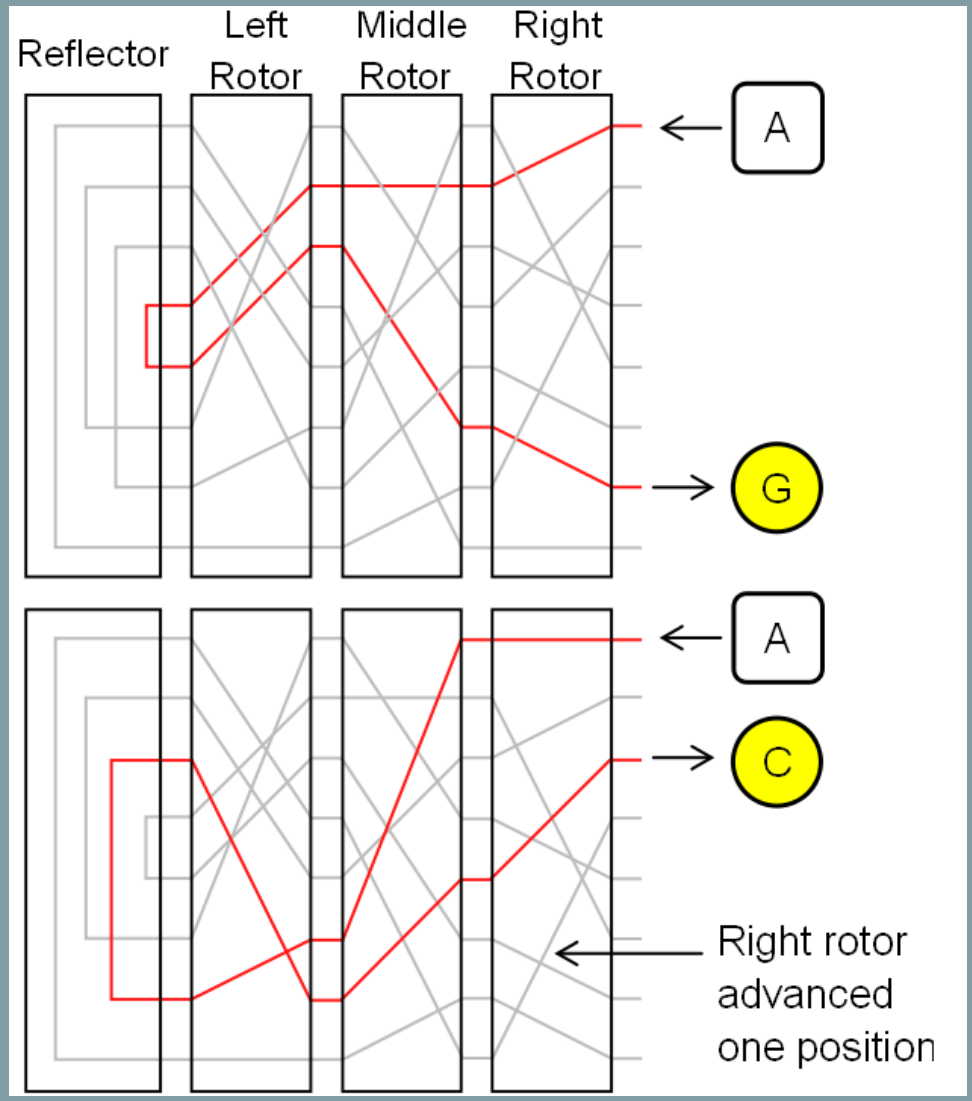
- Les principes de Kerckhoffs (**1883**) surprennent les cryptologues : Ne pas garder l'algorithme secret.
- Gilbert Vernam (**1917**) invente le système du masque jetable « one time pad ».
- Georges Painvin (**1918**) déchiffre un code allemand juste avant une attaque.
- Alan Turing (**1940**), Colossus et la machine Enigma.
- Claude Shannon (**1946**) « communication theory of secrecy systems » : Notion d'entropie et de quantité d'information.



- Elle est construite en 1918 par un ingénieur allemand, Arthur Scherbius.
- Elle est achetée, modifiée et utilisée par l'armée allemande dès 1926.
- Un membre du ministère de la défense allemand (Hans-Thilo Schmidt) en donne les plans aux français en 1931, qui les transmettent aux britanniques et aux polonais.
- Dès 1933, des mathématiciens polonais (Marian Rejewski) analysent le fonctionnement d'Enigma, en fabriquent des copies ainsi que des machines à déchiffrer.

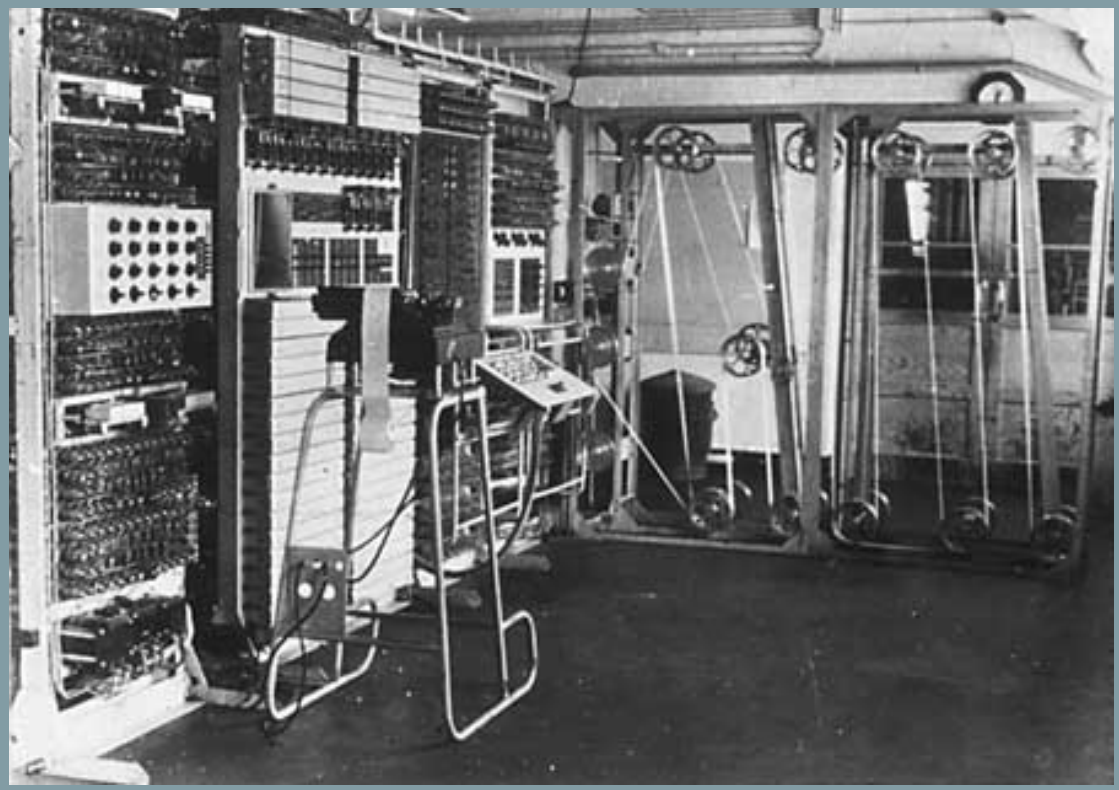
- Enigma est constituée de 3 rotors de 26 contacts reliés par un câblage électrique.
- On tape des lettres au clavier et des lampes s'allument pour montrer le texte chiffré.
- A chaque frappe, les tambours des rotors effectuent une rotation de 1/26 de tour.



- En 1939, juste avant l'invasion de la pologne, Rejewski remet aux français et aux anglais les répliques d'Enigma et les plans des bombes. En août, elles traversent la manche vers Londres dans les bagages de Sacha Guitry et de sa femme.
- A partir de l'automne 1939, le projet Ultra s'installe au manoir de Bletchley park, au Nord de Londres. D'abord 200 personnes, puis jusqu'à 8000 en 1945.

- Alan Turing, logicien et mathématicien anglais, construit à Bletchley des « bombes » de plus en plus rapides pour déchiffrer les messages allemands.

- Il participe également à la création du Colossus, premier ordinateur construit au monde (1943). Le premier colossus possède 1500 tubes à vide et le second (1944) en possède 2500, soit l'équivalent de 25 bits de mémoire.



- A partir de la seconde guerre mondiale, la cryptologie s'est mécanisée. Il n'est plus possible de chiffrer ou déchiffrer des messages à la main.

« Wolff s'approcha du buffet où il dissimulait l'émetteur radio. Il prit le roman anglais et la feuille de papier sur laquelle était inscrit le chiffre du code. Il l'étudia. On était aujourd'hui le 28 mai. Il fallait ajouter 42 -le chiffre de l'année- à 28 pour arriver au numéro de la page du roman qu'il devait utiliser pour coder son message. Mai était le cinquième mois de l'année, aussi allait-il supprimer une lettre sur cinq dans la page. [...] »

« Il décida d'envoyer comme message SUIS ARRIVE. M'INSTALLE. ACCUSEZ RECEPTION. Commençant en haut de la page 70 du livre, il chercha la lettre S. En supprimant une lettre sur cinq, le S était le dixième caractère de la page. Dans son code, il serait donc représenté par la dixième lettre de l'alphabet, le J. Il lui fallait ensuite un U. Dans le livre, la troisième lettre après le S était un U. Le U de SUIS serait donc représenté par la troisième lettre de l'alphabet, le C. Il y avait des façons particulières pour représenter les lettres rares, comme le X, par exemple. »

« Ce type de code était une variation de la feuille unique de bloc, la seule forme de code indéchiffrable en théorie comme en pratique. Pour décoder le message, il fallait avoir tout à la fois le livre et la clef. »

Le code Rebecca (Ken Follet)

- 2.1. Un peu d’histoire.
- 2.2. Entropie et information.
- 2.3. Le DES.
- 2.4. L’AES.
- 2.5. Autres protocoles.
- 2.6. Conclusion.

- Un tout petit peu de maths (mais pas beaucoup et pas longtemps) :
- ***x*** est le texte en clair et ***y*** est le texte chiffré.

Un système cryptographique est à confidentialité parfaite si $\mathbb{P}(x/y) = \mathbb{P}(x)$

- La connaissance du texte chiffré n'apporte aucune information sur le texte en clair.
- Gilbert Vernam en 1917 a inventé le système du masque jetable :

$$y = x \oplus k$$

- Claude Shannon démontre, en 1946, que cette méhtode est la seule qui soit à confidentialité parfaite.

- La théorie de l'information fournit une appréciation quantitative de l'information contenue dans un texte.
- C'est une mesure de l'inattendu, de l'improbable.
- Qu'est-ce qu'on entend par information ? Il faut distinguer syntaxe et sémantique :
- Exemple : « hier, un crocodile a mordu un homme. »

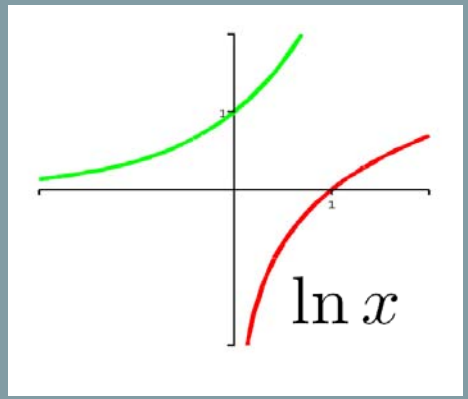
  « hier, un homme a mordu un crocodile. »
- Il faut tenir compte du sens et du contexte et pas seulement du contenu :

Il y a nécessité d'utiliser le langage des probabilités.

- En 1946, Shannon en donne une définition rigoureuse. Il définit la quantité d’information apportée par la réalisation d’un évènement **A** :
  - C’est une fonction décroissante de la probabilité.
  - L’évènement certain apporte une quantité d’information nulle.
  - La réalisation de deux évènements indépendants apporte une information égale à la somme des informations de chacun.

- Pour un évènement **A** d'une expérience aléatoire dont on connaît la loi de probabilité, on pose donc :

$$I(A) = \log \frac{1}{\mathbb{P}(A)} = -\log \mathbb{P}(A)$$



- $I(A) \geq 0$
- $I(\Omega) = -\log \mathbb{P}(\Omega) = -\log 1 = 0$
- $I(A \cap B) = -\log \mathbb{P}(A \cap B)$

$$= -\log(\mathbb{P}(A)\mathbb{P}(B))$$

$$= -\log \mathbb{P}(A) - \log \mathbb{P}(B) = I(A) + I(B)$$

- Initialement, la quantité d'information se mesurait en bit (binary unit). La norme ISO la mesure aujourd'hui en Shannon.
- Un Shannon est la quantité d'information apportée par la réalisation d'un évènement équiprobable parmi deux.

$$1 \text{ Sh } = \log_2(1/2)$$

- Exemple : Pile ou face.
- Autre exemple : Si nous choisissons une lettre au hasard.

$$I = -\log_2(1/26) = \log_2(26) \simeq 4,7 \text{ Sh}$$

Ou bien un objet parmi n : $I = \log_2 n \text{ Sh}$

- L'entropie d'une variable aléatoire représente la valeur moyenne de la quantité d'information.

X une variable aléatoire prenant comme valeurs $x_1, ..., x_n$ avec probabilité $p_1, ..., p_n$

l'entropie de X *est le réel* $H(X) = -\sum_{i=1}^{n} p_i \log p_i$

- L'entropie mesure le désordre, l'incertitude, la quantité d'aléatoire contenue dans la variable.

- Une variable aléatoire de Bernoulli pouvant prendre deux valeurs équiprobables a une entropie de : $1 \text{ Sh } = \log_2(1/2)$

- $H(X) \geq 0$
- $H(X) = 0 \iff \exists i / \ p_i = 1$
- $H(X) \leq \log n$
- $H(X) = \log n \iff p_i = 1/n \ \forall i = 1...n$

- L’entropie d’un langage naturel mesure la moyenne de l’information fournie par une lettre, dans un message qui a un sens.

- Dans un langage contenant tous les mots possibles d’un un alphabet de 26 lettres, l’entropie d’une lettre est $\log_2(26) \simeq 4,7 \text{ Sh}$

- Mais dans tout langage, les lettres successives ne sont pas équiprobables. Il y a beaucoup de **E**, les **Q** sont souvent suivis de **U**, etc.

  Il est plus probable de trouver une phrase comme « **merci Bernard** » que « **&@-zs!!df** ».

- L’entropie d’un langage (ou son taux) se calcule par la formule

$$r = \lim_{n \to +\infty} \frac{H_n}{n}$$

H_n étant l’entropie d’un message de longueur ***n***.

- Le taux de l’anglais est estimé à environ **1,3** (4,6 Sh / lettre).
- Le taux du français est estimé à environ **1,1** (3,9 Sh / lettre)
- On peut également calculer le taux de n’importe quel langage informatique, celui de l’ADN ou bien de la musique.

- L’entropie d’un cryptosystème est : $H = \log_2(|\mathcal{K}|)$

  $|\mathcal{K}|$ étant le nombre de clefs possibles.

- Nous pouvons également définir l’entropie conditionnelle de la clef connaissant le texte chiffré (cela s’appelle l’***ambiguïté de la clef***)

$$H(k/y) = K(k) + H(x) - H(y)$$

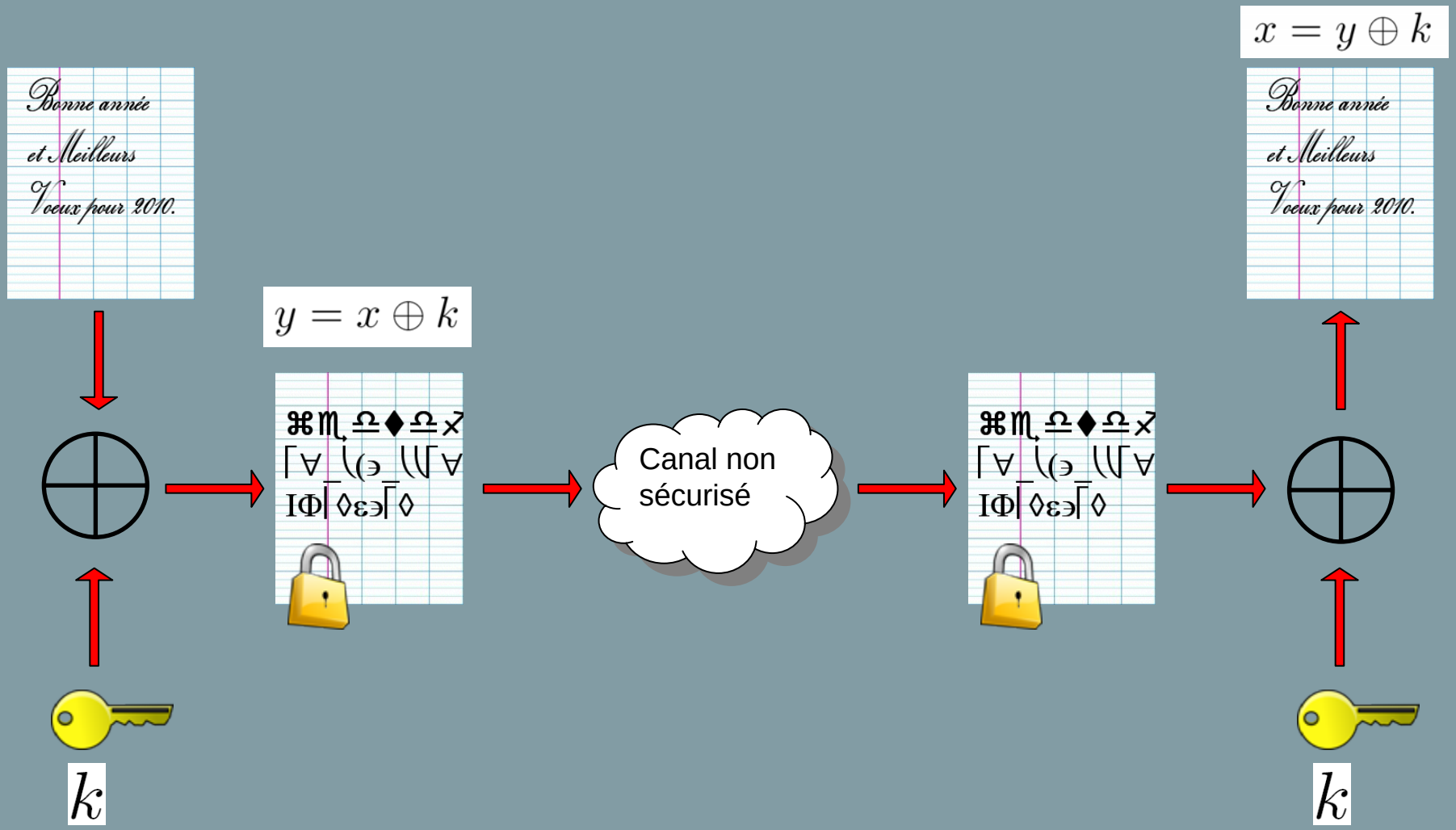
- Connaissant le texte chiffré ***y***, le cryptanalyste essaie de deviner la clef ***k***. Un système est à confidentialité parfaite si :

$$H(k/y) = K(k)$$

- Inventé par Gilbert Vernam en 1917, c'est le système du masque jetable. Shannon a démontré en 1946 que ce cryptosystème était le seul à confidentialité parfaite.
- De plus, il est très simple à mettre en œuvre : Le texte chiffré est égal au ou exclusif du texte en clair et de la clef (l'opération est effectuée bit par bit) : $y = x \oplus k$
- Le chiffrement est exactement la même opération que le déchiffrement ! $x = y \oplus k$

  En effet, $y \oplus k = x \oplus k \oplus k = x$

Le cryptosystème parfait : « One time pad ». (2)



- Exemple:

x	**Ce système est à confidentialité parfaite.**
k	**ñ@ ´¤ ªÁ÷WÜP├ ½‰ðŠÀÃ¹ ïmP¼¬ôÆ€0:½~ÏÓÍÿ}t&▯’¨**
y	**²%+ÇvÙµ :¹p|ÎýÐjà Ög‰˩ 4ÙÂ€¯á\SÉ—ï£¬▯← ⊥ Oõ÷†**

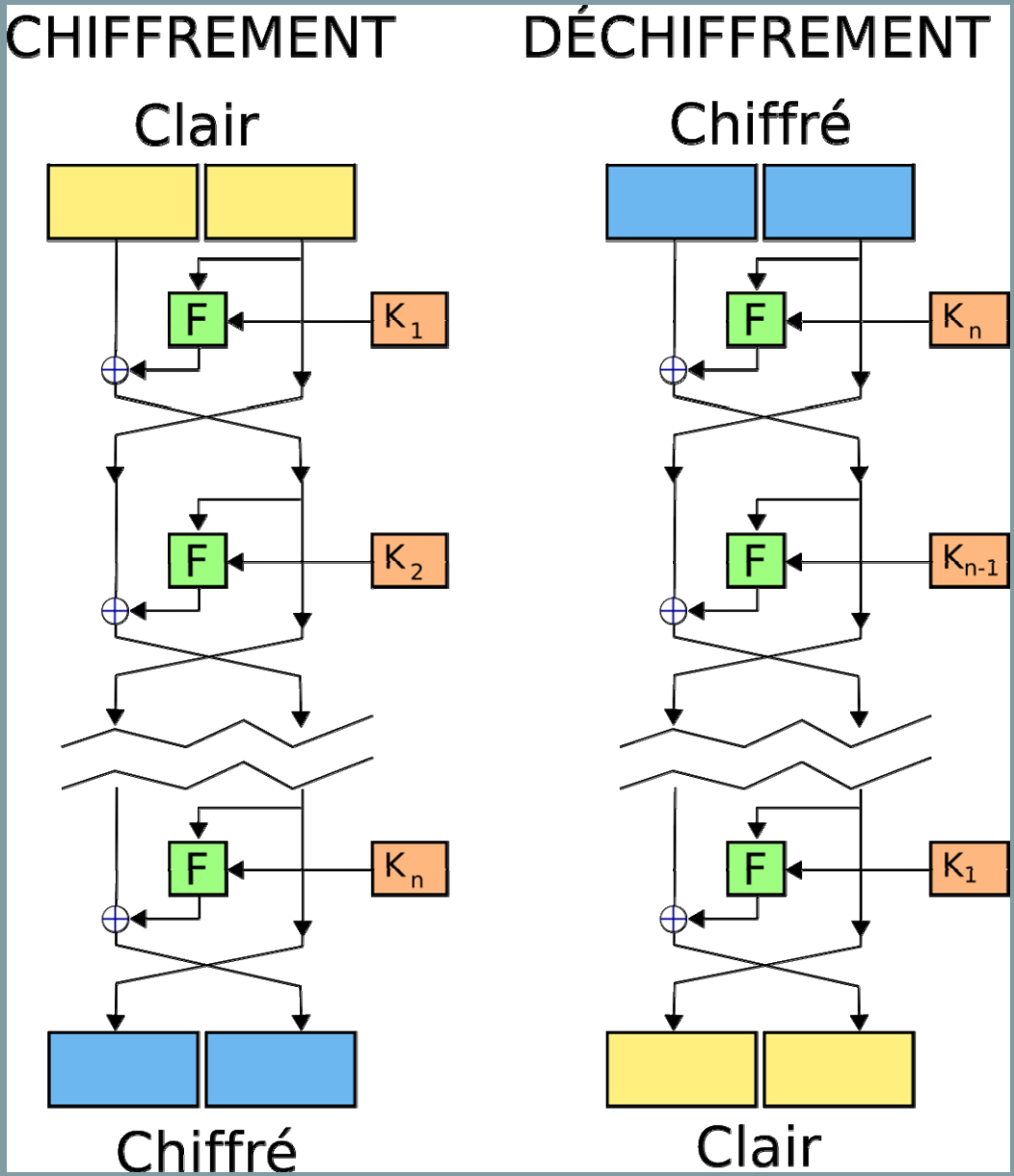
- Le principe est le même que pour le système de Vigenère, mais avec une clef aussi longue que le texte et parfaitement aléatoire.

- Si ce système est parfait, pourquoi ne l'utilise-t-on pas ?
- En fait, il est utilisé :
  - Pour des applications militaires.
  - Pour les communications entre ambassades (téléphone rouge).
- Les gros inconvénients :
  - La clef doit être aussi longue que le fichier à transmettre.
  - Elle ne doit jamais être utilisée deux fois. La sûreté de l'algorithme repose sur cette utilisation unique.
  - Elle doit être parfaitement aléatoire.
  - Elle doit être échangée par un canal sûr.

- Un traitement itératif des messages est effectué par bit ou par bloc de longueur fixe :
  - Par bit : chiffrement en continu (« stream-cipher »).
  - Par bloc : DES, AES, Idea.
- Le protocole doit respecter deux principes :
  - **Confusion** : Le chiffré doit dépendre du clair de façon la plus complexe possible pour empêcher la cryptanalyse.
  - **Diffusion** : Chaque symbole du chiffré doit dépendre de chaque symbole du clair et de la clef ; le texte en clair et la clef sont dilués (dissous) dans le chiffré.

- Shannon propose des réseaux de substitution (S-box) et / ou permutation (P-box) généralisant la méthode de Vigenère.
- La répétition des opérations de chaque réseau imitera la méthode du masque jetable et augmentera la diffusion et la confusion dans le message.
- Horst Feistel présente un système général simple permettant d'effectuer ces opérations à chaque itération :
  - Chaque bloc est découpé en deux parties (gauche et droite)
  - Une substitution est effectuée sur la partie gauche.
  - Une fonction de brouillage mélange une partie de la clef et la partie droite.
  - Les parties gauche et droite sont permutées.

Schéma de Feistel.



- 2.1. Un peu d’histoire.
- 2.2. Entropie et information.
- 2.3. Le DES.
- 2.4. L’AES.
- 2.5. Autres protocoles.
- 2.6. Conclusion.

- Le Data Encryption Standard a été développé à partir de 1973 par IBM à partir de l'algorithme LUCIFER et validé en 1975 par la NSA.
- Sa sécurité était évaluée tous les cinq ans. Il a été officiellement abandonné en 2000.
- Il a été l'algorithme à clef secrète le plus utilisé au monde.
- Il est environ 1000 fois plus rapide que le RSA : ses implémentations matérielles sur puces dédiées effectuent plusieurs centaines de millions de chiffrement par seconde et traitent plusieurs gigabits de données par seconde.

- Le DES traite des données par blocs de 64 bits qu'il chiffre en blocs de même taille.
- L'algorithme de chiffrement et de déchiffrement est le même.
- La taille de la clef secrète est de 56 bits et toute la sécurité de l'algorithme repose dessus.
- Chaque bloc subit une répétition de 16 opérations élémentaires appelées ***rondes*** ou ***étages***, durant lesquelles les bits du texte en clair sont mélangés à l'aide de la clef.
- Chaque ***ronde*** est définie à l'aide d'une opération de brouillage qui dépend de la clef secrète. Ce brouillage doit comprendre au moins une opération ***non linéaire***.

- Le DES suit donc le schéma de Feistel. Nous allons en préciser chaque étape.
- On considère un bloc de texte clair $x = (x_1 x_2 .... x_{64})$
- Une permutation initiale **P** lui est appliquée avant la première ronde, afin de rendre plus facile son implantation sur des puces. A la fin de l'algorithme, la permutation inverse sera appliquée sur le bloc en sortie.

P							
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

P^{-1}							
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

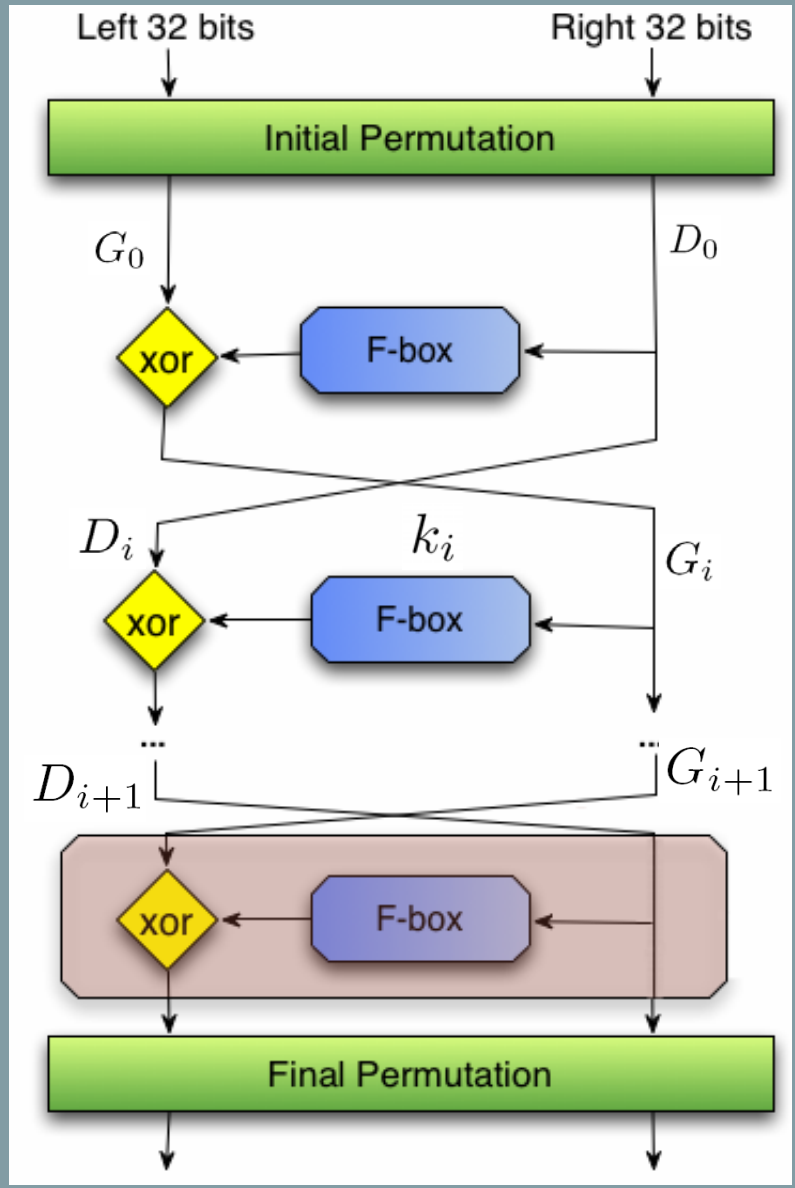
- Les données permutées de 64 bits sont séparées en deux groupes de 32 bits qui formeront la partie gauche et la partie droite initiales : G_0 D_0
- On applique alors le processus de brouillage 16 fois de suite.
- Lors de la ième ronde, les 32 bits de gauche G_i et les 32 bits de droite D_i sont mélangés pour produire en sortie les bits formant G_{i+1} et D_{i+1} sur lesquels seront appliqués la ronde suivante.

- A chaque ronde, la partie droite D_i des bits est mélangée par la fonction de brouillage f pour former $f(D_i, k_i)$
- $f(D_i, k_i)$ dépend des bits D_i mais également d'une sous clef notée k_i qui change à chaque ronde et qui calculée à partir de la clef secrète initiale k
- k_i est calculcée lors d'une opération appelée diversification de la clef k (ou aussi permutation compressive).

- Finalement, la sortie de la fonction de brouillage est additionnée à la partie gauche des bits par un ou exclusif.
- Le résultat devient la nouvelle partie gauche, tandis que l'ancienne partie gauche devient la nouvelle partie droite. Résumons :

$$\begin{cases} D_{i+1} = G_i \oplus f(D_i, k_i) \\ G_{i+1} = D_i \end{cases}$$

Principe du DES. (5)



$$\begin{cases} D_{i+1} = G_i \oplus f(D_i, k_i) \\ G_{i+1} = D_i \end{cases}$$

- Initialement, la clef fait 64 bits, mais 8 de ces bits sont utilisés comme bits de parité et seuls 56 bits servent donc réellement. A chaque ronde, on décale les bits de la clef de façon différente pour obtenir la sous clef k_i
- Les bits de k sont d’abord ordonnés à l’aide de la permutation suivante :

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

- Puis le résultat est séparé en deux parties de 28 bits et chacune des partiés est décalée à gauche d’une ou deux positions :

N° de ronde	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

- Après ce décalage, 48 bits parmi les 56 sont sélectionnés par la permutation pour donner finalement k_i :

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

$\Rightarrow k_i$

- Elle est formée d'une permutation expansive (non linéaire) suivie d'une série de substitutions par une série de S-boites (effet d'avalanche).
- La moitié droite, formée de 32 bits, est transformée en un vecteur X_i de 48 bits en répétant et permutant certains bits initiaux :

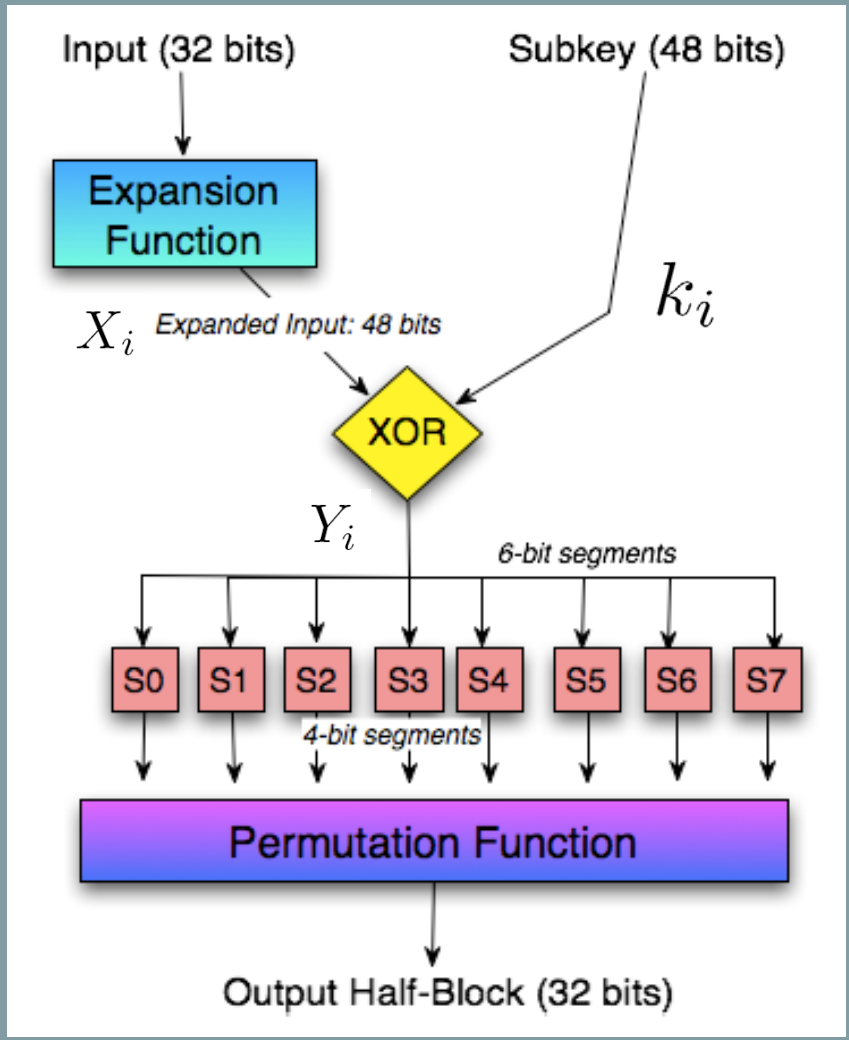
32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

- Ce vecteur est ajouté à la clef de ronde pour donner :

$$Y_i = X_i \oplus k_i$$

- Les 48 bits de Y_i vont être transformés par passage dans 8 tables appelées des S-boites (substitution box).
- Chaque table prend 6 bits en entrée et en restitue 4 en sortie. On divise donc les 48 bits en 8 paquets de 6 qui sont chacun envoyés dans une S-boite différente.

La fonction de brouillage. (3)



- Chaque S-boite est composée de 4 lignes et 16 colonnes qui contiennent un nombre de 4 bits (entier entre 0 et 15).
- Nommons les bits d'entrées de la S-boite $b_1, ..., b_6$
- On concatène b_1 et b_6 pour former un nombre de 2 bits et qui correspond à un numéro de ligne.
- On concatène b_2 à b_5 pour former un nombre de 4 bits et qui correspond à un numéro de colonne.
- A l'intersection de la ligne et de la colonne se trouve un entier de 4 bits : ce sera la valeur à la sortie de la S-boite.

Le contenu des S-boites. (2)

S_1

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

S_2

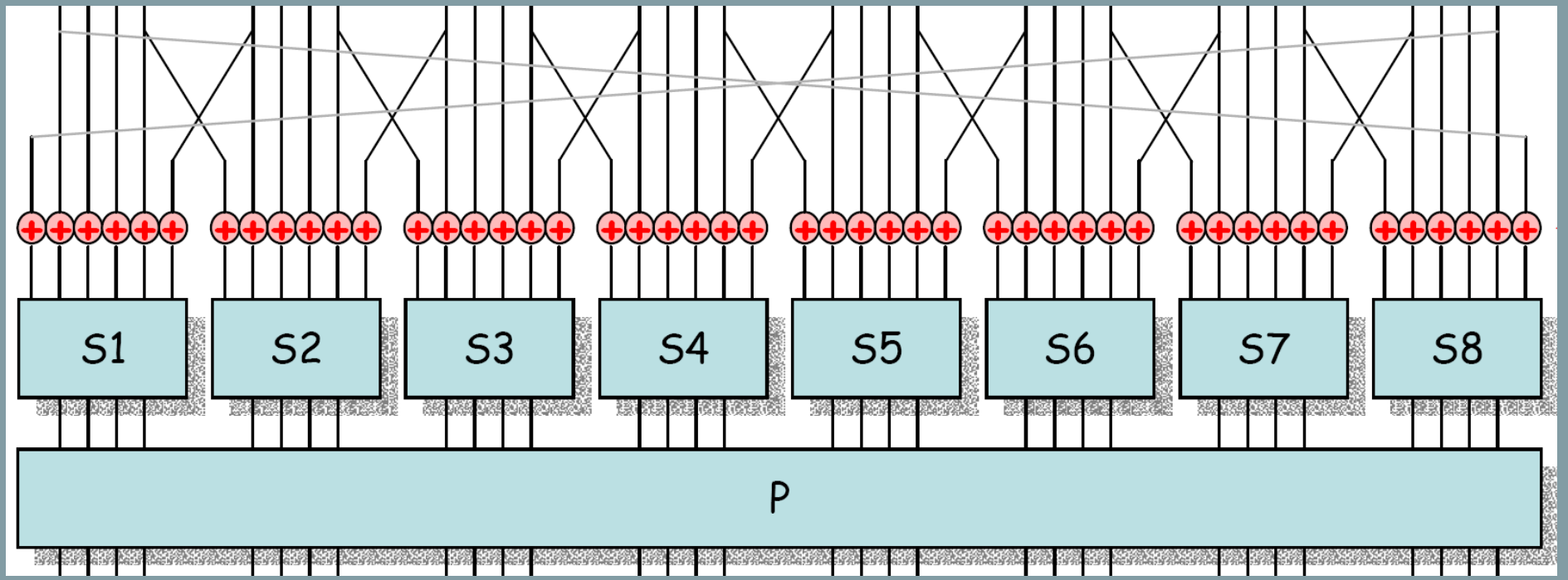
15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9

S_8

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

- A la sortie des S-boites, les 32 bits sont reconstituées et forment la sortie droite

$$f(D_i, k_i)$$



- Les S-boites forment la partie non linéaire de l'algorithme, et donc également la partie la plus sensible. Leur contenu doit :
  - Augmenter rapidement la confusion.
  - Etre fortement non-linéaire : aucun bit de sortie ne doit s'approcher d'une fonction linéaire des bits d'entrées.
- Durant plusieurs années, leur contenu fut gardé secret, ce qui provoqua la méfiance de toute la communauté des cryptologues.
- On pensait que la NSA gardait secrète une trappe permettant de déchiffrer rapidement le DES.

- Il existe plusieurs types d'attaque contre le DES :
  - Force brute : On teste toutes les clefs possibles.
  - Cryptanalyse différentielle : proposée par Shamir en 1990.
  - Cryptanalyse linéaire : proposée par Matsui en 1993.

- On compare la différence (ie. la distance) entre deux textes chiffrés et la façon dont cette distance se propage à travers les rondes.
- Avec cette technique, on casse le DES à 8 rondes en quelques minutes avec un PC de bureau.
- On s’est aperçu que les S-boites de la NSA étaient optimisées contre la cryptanalyse différentielle : elles étaient conçues pour y résister dès 1973.

2^{43}

- Plus rapide que l'attaque précédente.

2^{43}

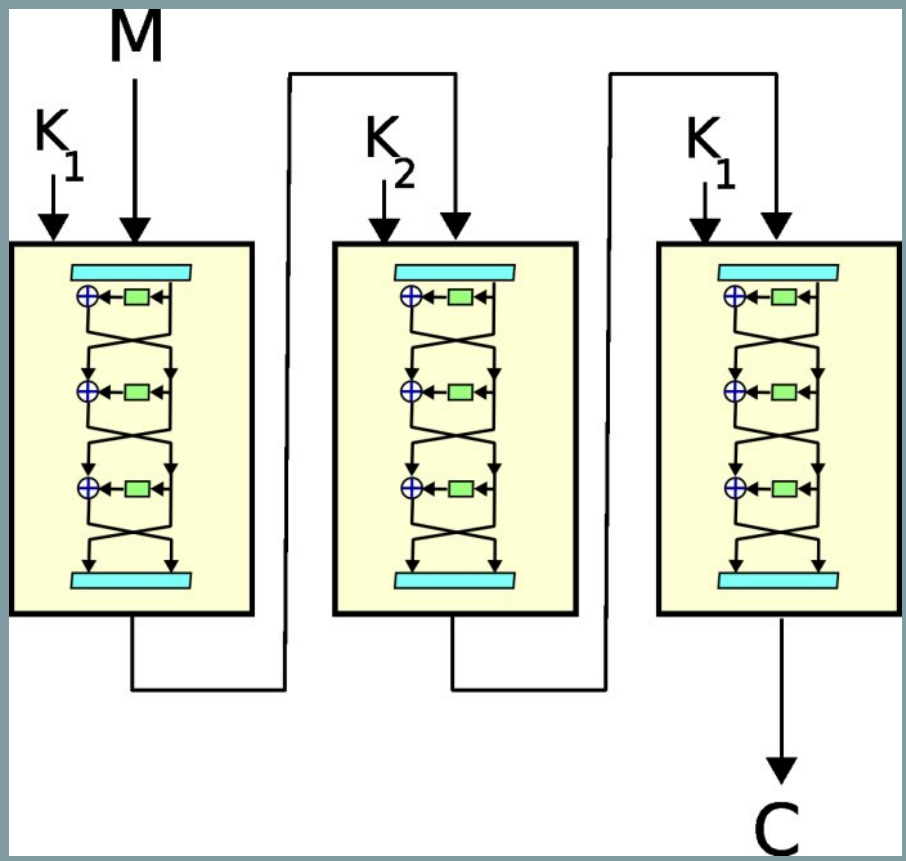
- Dans cette attaque, on cherche des combinaison linéaire entre les bits d'entrée, de sortie, et ceux de la clef.
- Lorsque de telles combinaisons apparaissent avec une probabilité supérieure à ½, alors on a détecté une corrélation exploitable.
- En utilisant 12 stations de travail durant 50 heures, Matsui a réussi à trouver une clef de 56 bits et à casser le DES.

- Finalement, c'est une attaque par force brute qui est venue à bout du DES en 1998.

$$2^{56}$$

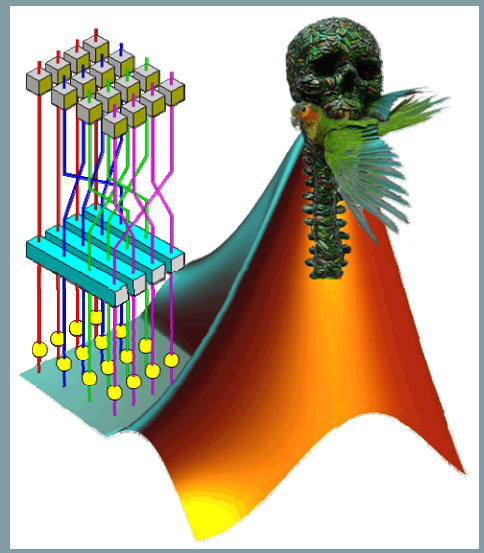
- Quelques milliers d'ordinateurs Pentium reliés par internet ont trouvé la clef en un peu plus d'un mois.
- L'année suivante, une seule machine construite spécialement pour casser le DES, a réussi à le cryptanalyser en quelques heures.
- On a toujours reproché au DES sa clef trop courte (LUCIFER possédait une clef de 128 bits).

Le triple DES. (1)



- On effectue trois fois de suite l'algorithme DES avec deux clefs différentes.
- On obtient un système avec une clef secrète de 112 bits, mais trois fois plus lent que le DES.
- Il sera sans doute cryptanalysé à court terme.
- Le DES a été officiellement abandonné en 2000 et remplacé par l'AES.

- 2.1. Un peu d'histoire.
- 2.2. Entropie et information.
- 2.3. Le DES.
- 2.4. L'AES.
- 2.5. Autres protocoles.
- 2.6. Conclusion.



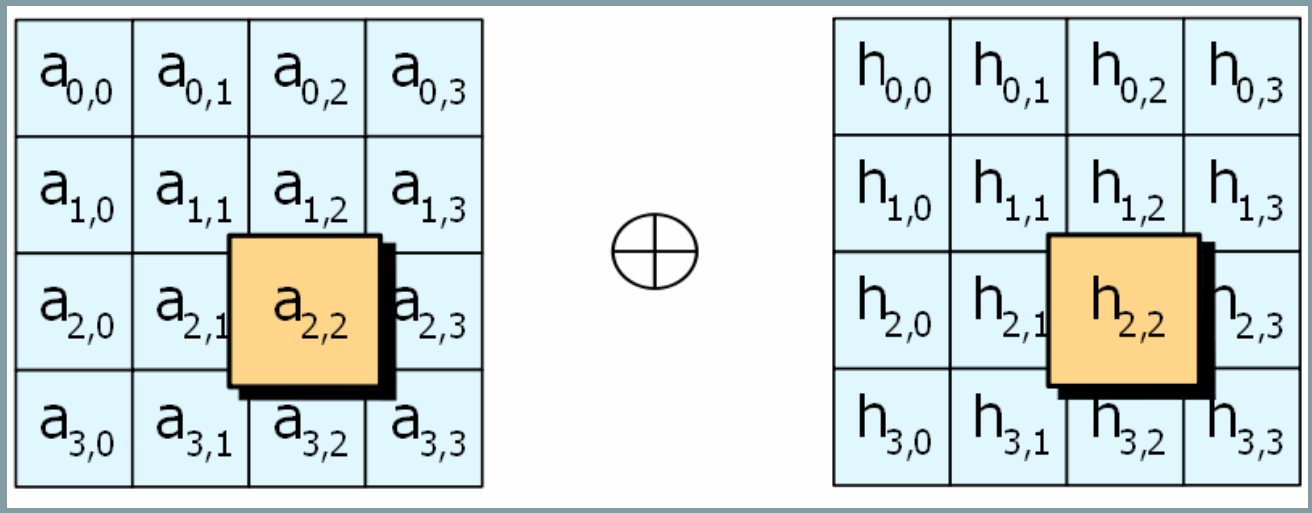
- En 2000, la NIST a lancé un appel à candidature pour le choix d’un nouveau standard de chiffrement. Le cahier des charges possédait de nombreuses contraintes :
  - Algorithme plus sûr que le triple DES.
  - La clef secrète devait être de 128, 192 ou 256 bits .
  - L’algorithme devait être rapide .
  - Facilement portable ; aussi bien sur de gros calculateurs que sur des cartes à puce .
  - Le protocole devait être libre de droit.
- En octobre 2000, l’AES de Joan Daemen et Vincet Rijmen a gagné. Il a été nommé Rijndael.

- Il ne respecte pas le schéma de Feistel mais s'inspire du DES.
- Les données sont découpées en bloc de 128 bits.
- Chaque bloc est réparti dans une matrice A de 4 lignes et 4 colonnes dont les coefficients forment des entiers de un octet.
- On notera cette matrice $A = (a_{i,j})$

$A =$

$a_{0,0}$	$a_{0,1}$	$a_{0,2}$	$a_{0,3}$
$a_{1,0}$	$a_{1,1}$	$a_{1,2}$	$a_{1,3}$
$a_{2,0}$	$a_{2,1}$	$a_{2,2}$	$a_{2,3}$
$a_{3,0}$	$a_{3,1}$	$a_{3,2}$	$a_{3,3}$

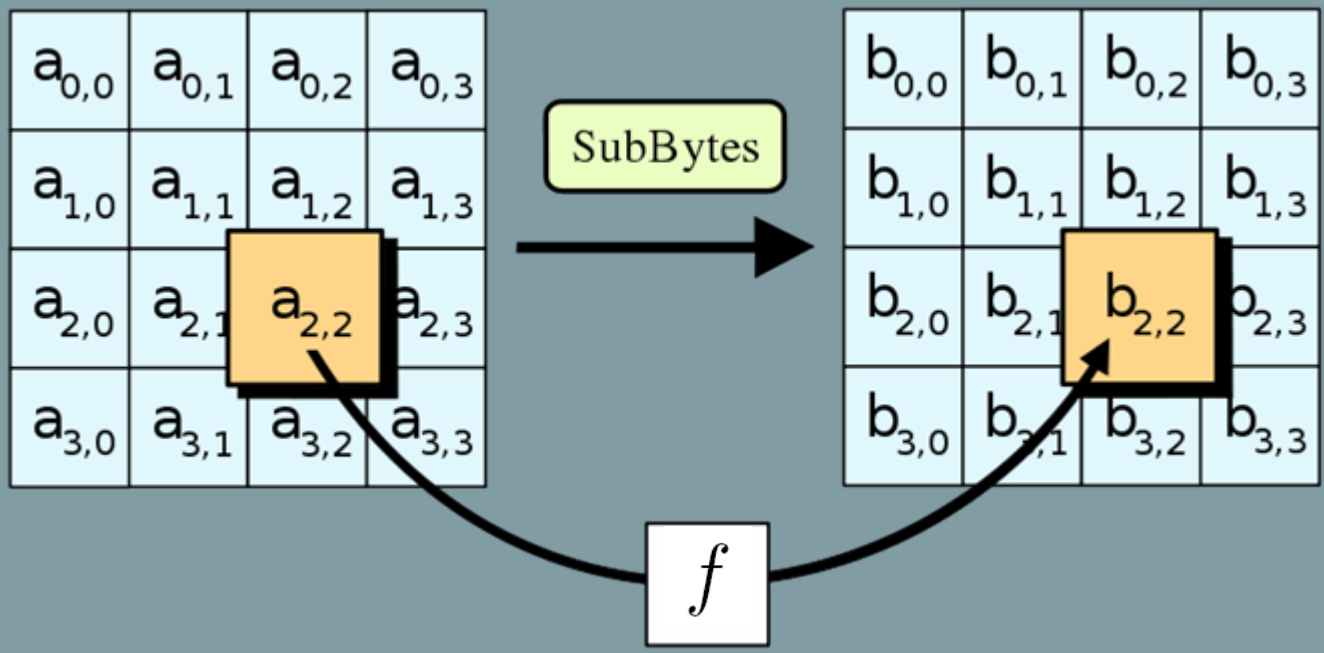
- On commence par ajouter à la matrice, la clef initiale sous la forme d'une matrice de 4 lignes par 4 colonnes :



Pour simplifier, on ne change pas de nom la matrice obtenue.

- On applique 10 rondes combinant les 4 opérations suivantes :
  - **1.** Tranformation non linéaire : appliquée à chacun des 16 éléments de la matrice A pour donner : $B = (b_{i,j})$

$$b_{i,j} = f(a_{i,j}) \quad \forall i, j = 1..4$$



- La fonction de brouillage utilise une S-boite qui associe un nombre à chaque coefficient de la matrice.
- Le contenu des S-boites est optimisé pour résister à la cryptanalyse différentielle et linéaire.
- Chaque S-boite se représente par un tableau de 16 sur 16. Une ligne représente les 4 bits les plus significatifs du coefficient en entrée et une colonne les 4 bits les moins significatifs.
- A l'intersection de la ligne et de la colonne se trouve la valeur qui sera renvoyée en sortie de la S-boite.
- Avant d'être envoyé à la S-boite, le coefficient subit une inversion calculée dans le corps fini $\mathbb{F}_{256}$ suivie d'une multiplication et de l'ajout d'une constante.

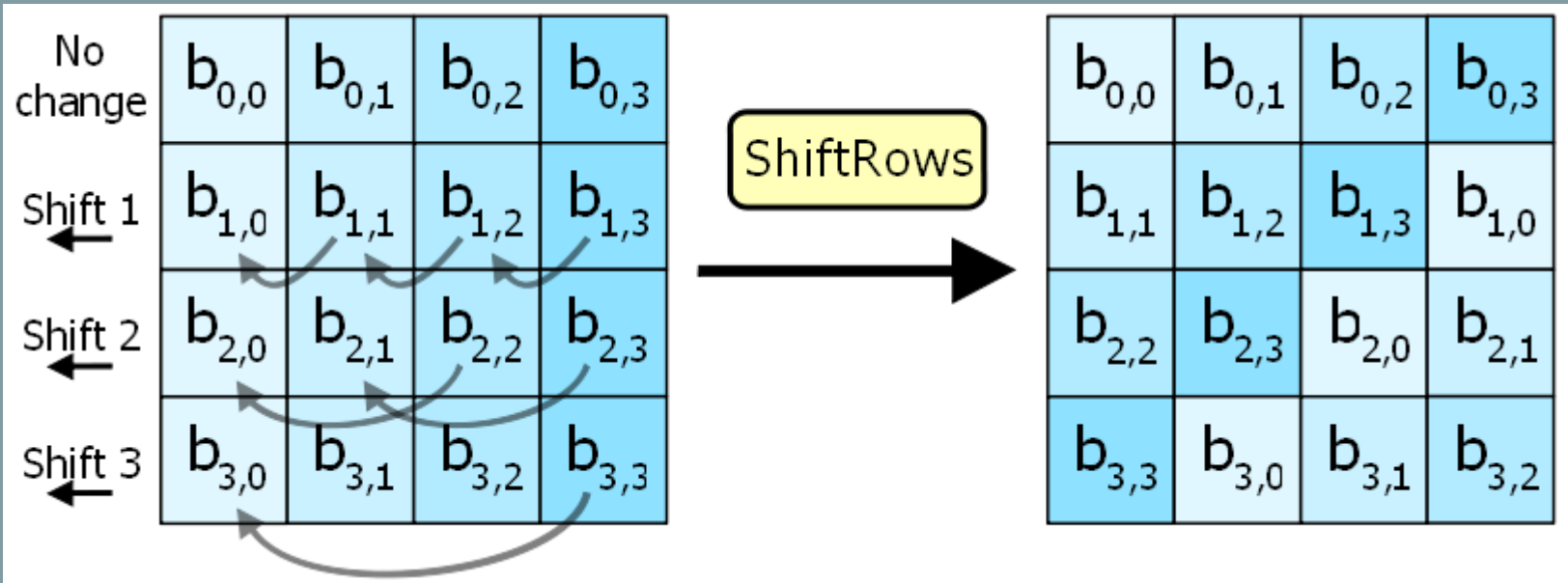
- Un exemple de S-boite :

	x0	x1	x2	x3	x4	x5	x6	x7	x8	x9	xa	xb	xc	xd	xe	xf
0x	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
1x	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
2x	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
3x	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
4x	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
5x	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
6x	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
7x	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
8x	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
9x	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
ax	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
bx	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
cx	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
dx	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
ex	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
fx	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

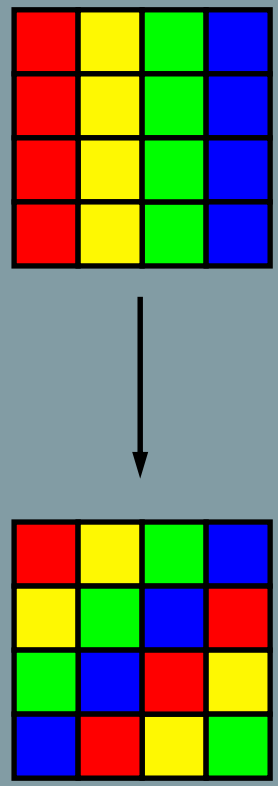
- Le vecteur x est obtenu en calculant l'inverse dans $\mathbb{F}_{256}$ du coefficient. On calcule ensuite le produit de ce vecteur par une matrice et on lui ajoute un vecteur binaire.

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

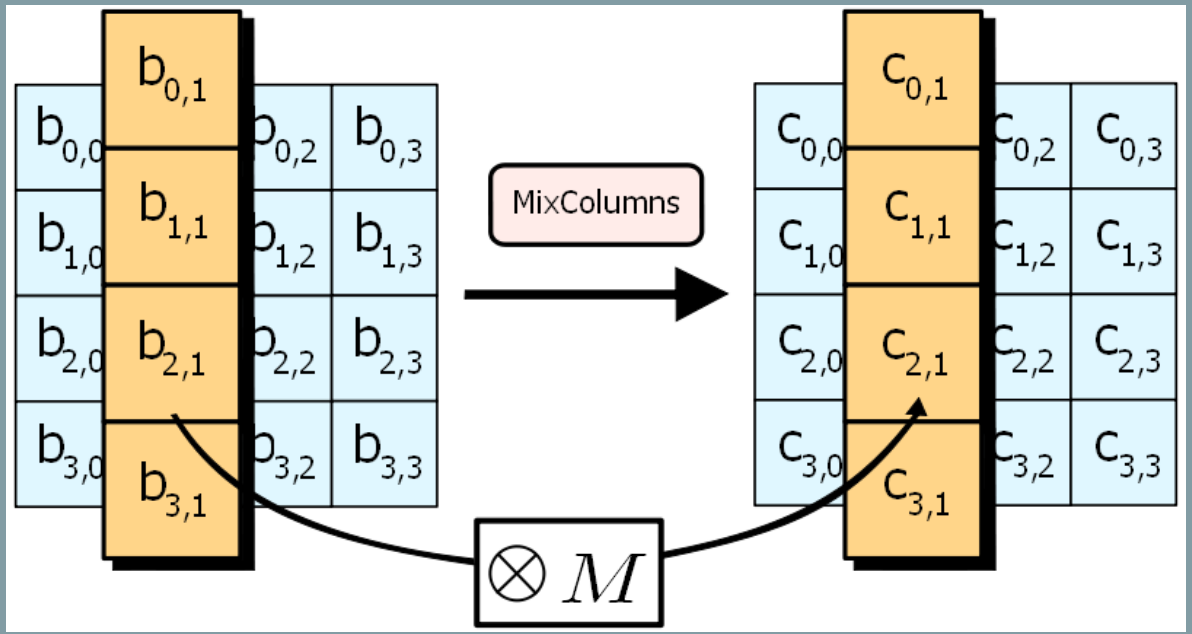
- **2.** Décalage de lignes : les trois dernières lignes de B sont décalées de façon cyclique vers la gauche avec des décalages différents.

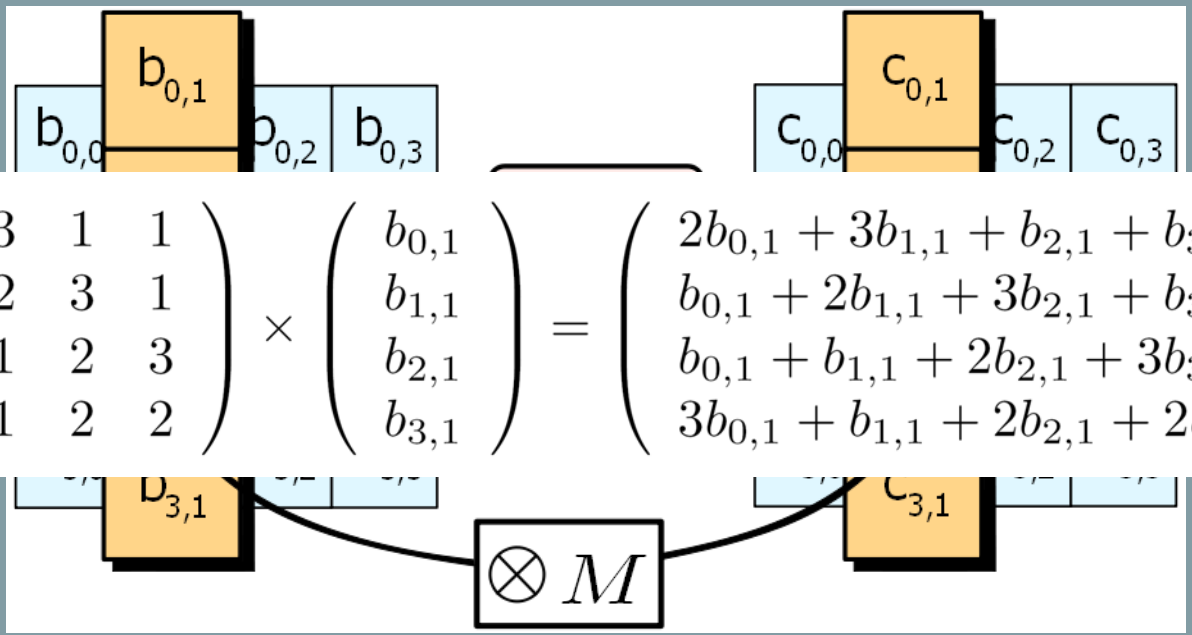


- De façon plus imagée :



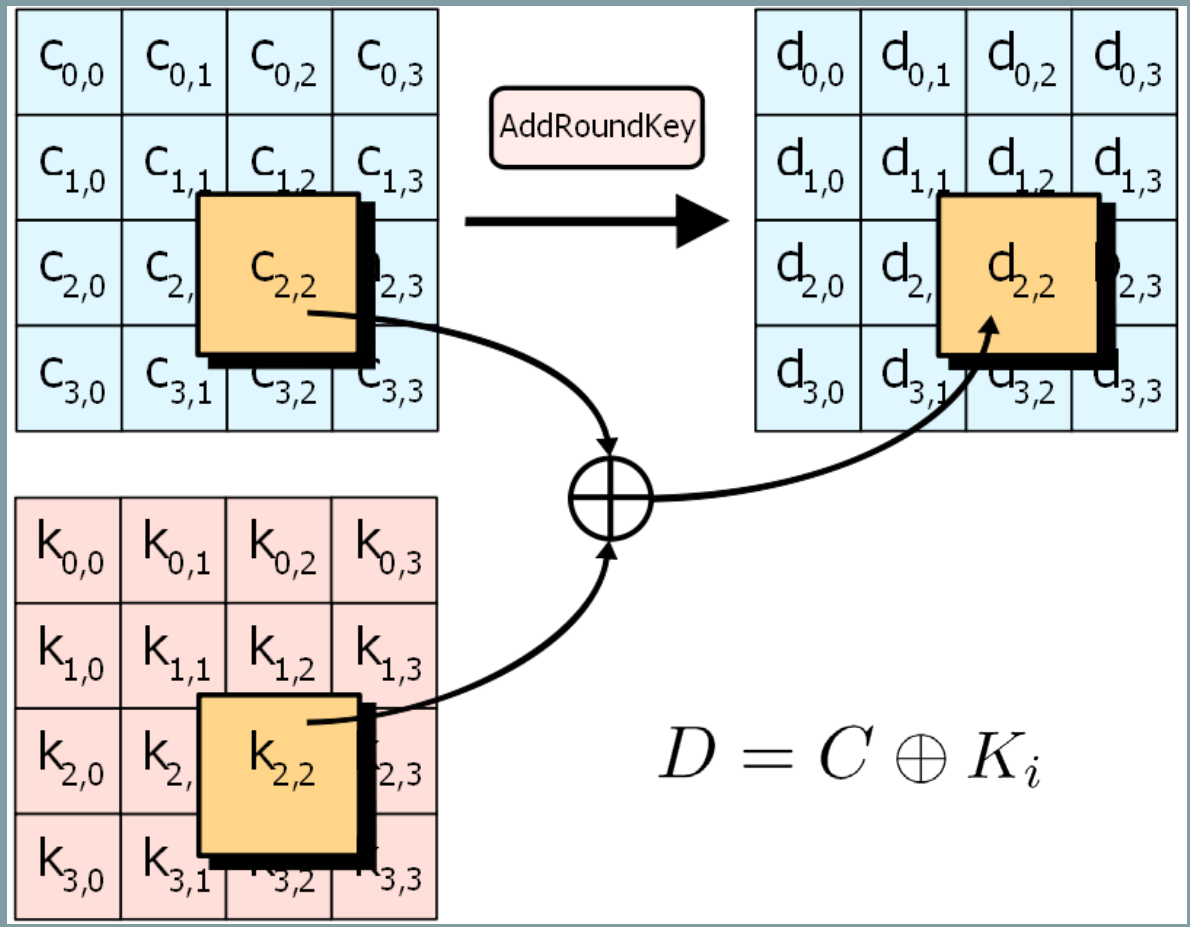
- **3.** Brouillage des colonnes : chaque colonne est transformée en une nouvelle colonne à l'aide d'une multiplication par une matrice dont les coefficients sont 1, 2 ou 3. Ce produit est effectuée dans le corps fini $\mathbb{F}_{256}$





$$\begin{pmatrix} c_{0,1} \\ c_{1,1} \\ c_{2,1} \\ c_{3,1} \end{pmatrix} = \begin{pmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 2 & 2 \end{pmatrix} \times \begin{pmatrix} b_{0,1} \\ b_{1,1} \\ b_{2,1} \\ b_{3,1} \end{pmatrix} = \begin{pmatrix} 2b_{0,1} + 3b_{1,1} + b_{2,1} + b_{3,1} \\ b_{0,1} + 2b_{1,1} + 3b_{2,1} + b_{3,1} \\ b_{0,1} + b_{1,1} + 2b_{2,1} + 3b_{3,1} \\ 3b_{0,1} + b_{1,1} + 2b_{2,1} + 2b_{3,1} \end{pmatrix}$$

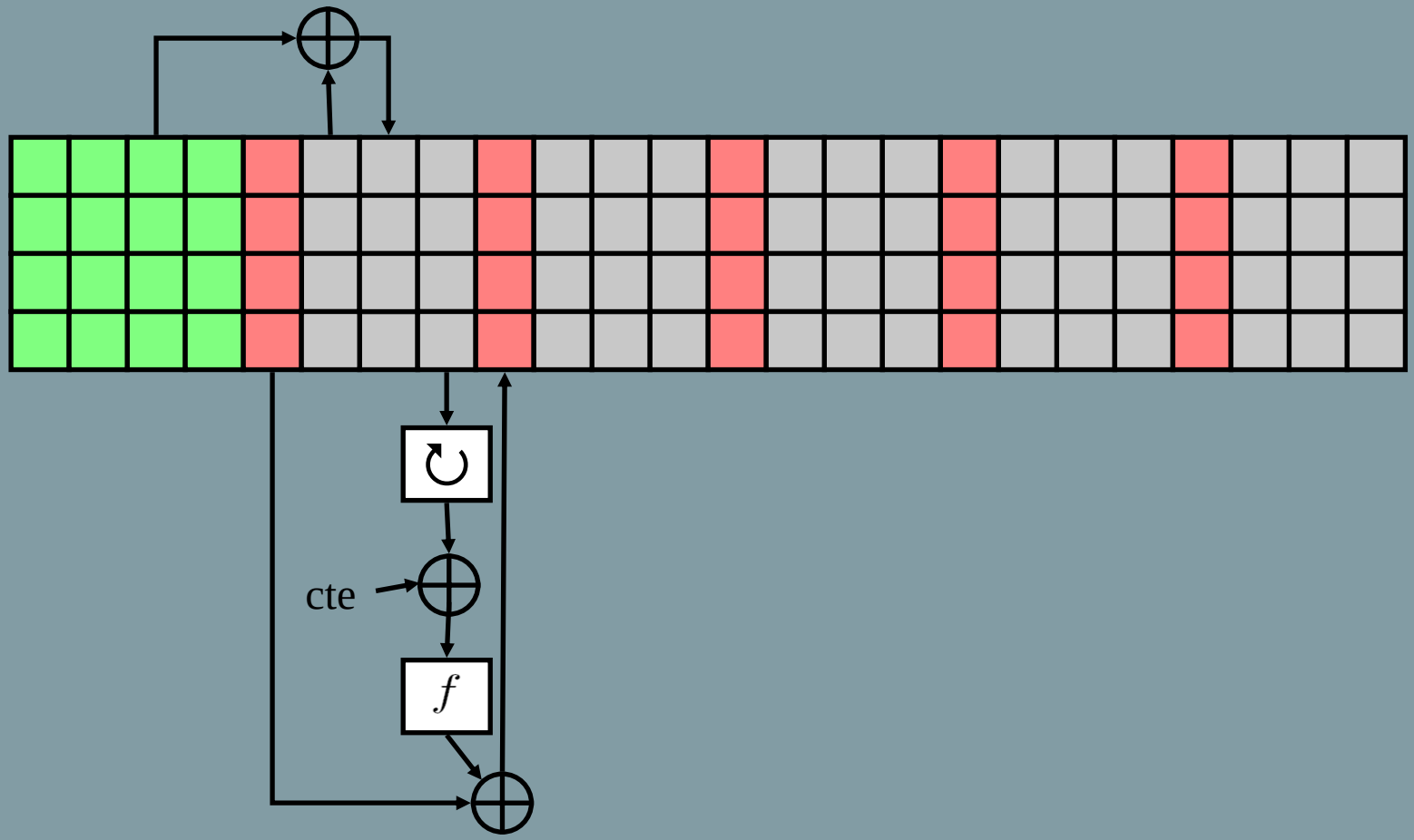
- **4.** Addition de la clef de ronde : une clef différente est ajoutée à chaque tour à l'aide d'un ou exclusif à la matrice C



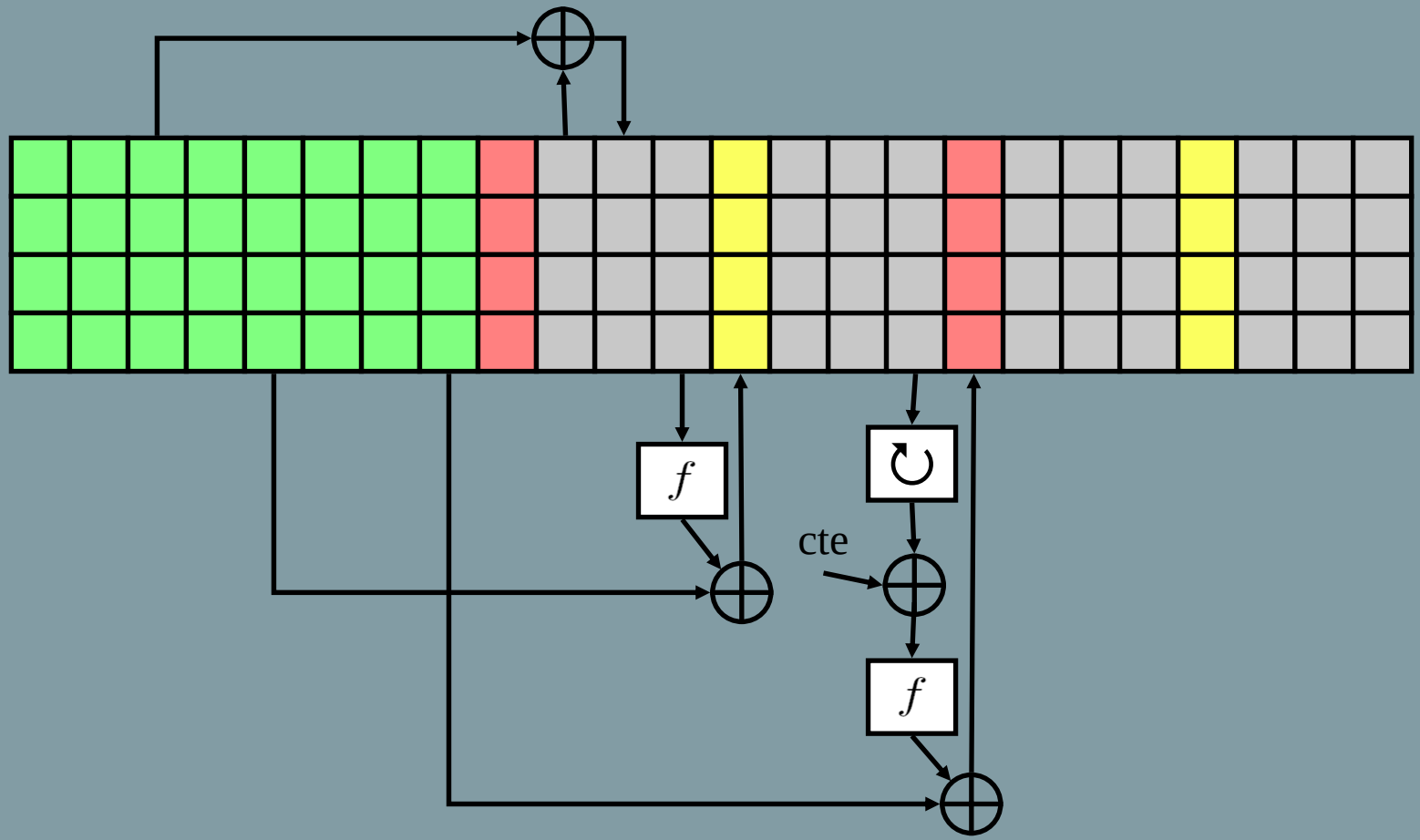
- La clef de ronde K_i est différente à chaque tour. Elle est calculée à partir de la clef secrète initiale K . Celle-ci est mise sous la forme d'une matrice de 4 lignes sur N_k colonnes.
- N_k est égal au nombre de bits de K divisé par 32 (4, 6 ou 8)
- On pose $K_0 = K$ et K_i s'obtient à partir de K_{i-1} en permutant les quatre derniers octets de la clef, puis en appliquant la fonction de brouillage f
- Après avoir ajouté une constante dépendant de i au premier octet, on ajoute à l'aide d'un ou exclusif les quatre octets obtenus et les quatre premiers de la clef K_{i-1}

- Les trois autres blocs de quatre octets de K_i sont obtenus par un ou exclusif entre le bloc correspondant de la sous clef K_{i-1} et du bloc précédent de K_i

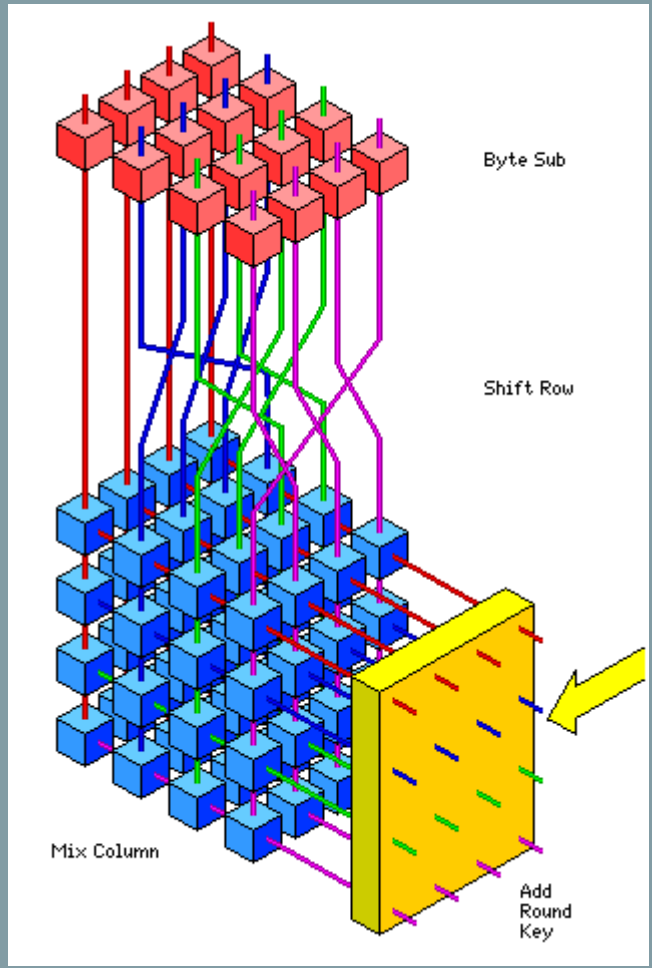
- Voici l'algorithme de cadencement pour 128 ou 192 bits.



- Voici l'algorithme de cadencement pour 256 bits.



Essayons de tout résumer en un seul schéma.

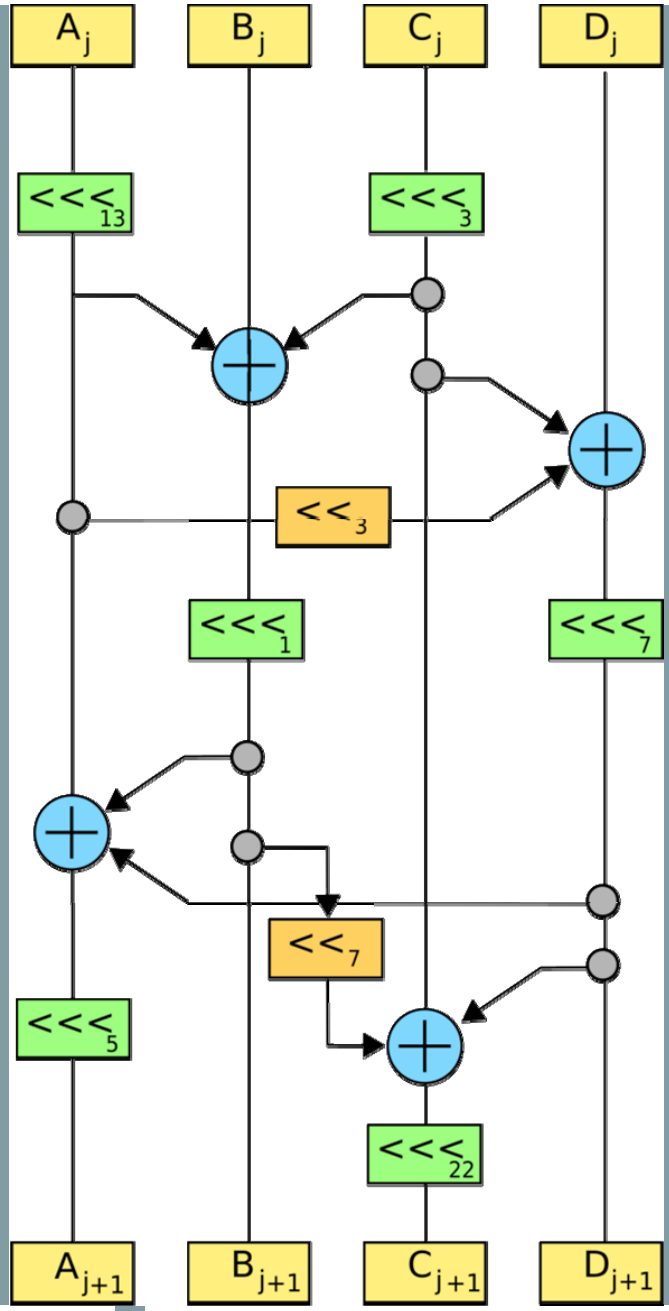


- Lorsque la clef secrète fait 128 bits, on effectue 10 rondes. Lorsqu’elle fait 192 bits, on effectue 12 rondes et lorsqu’elle fait 256 bits, on en effectue 14.
- A l’issue des rondes, chaque bloc de 128 bits sort de l’algorithme sous forme chiffrée.
- Le déchiffrement suit les mêmes étapes, mais en inversant le contenu des S-boites et les clefs de rondes.
- Regardons de façon plus imagée l’effet du chiffrement sur un bloc.

- Avantages :
  - Tous ceux demandés par le cahier des charges (rapidité, flexibilité, portabilité, etc.).
  - En 2010, c’est l’algorithme à clef secrète le plus utilisé au monde.
- Inconvénients :
  - L’AES est (relativement) jeune. Ses failles ne sont peut-être pas connues.
  - Il existe une attaque sur 6 rondes et sur 9 rondes.

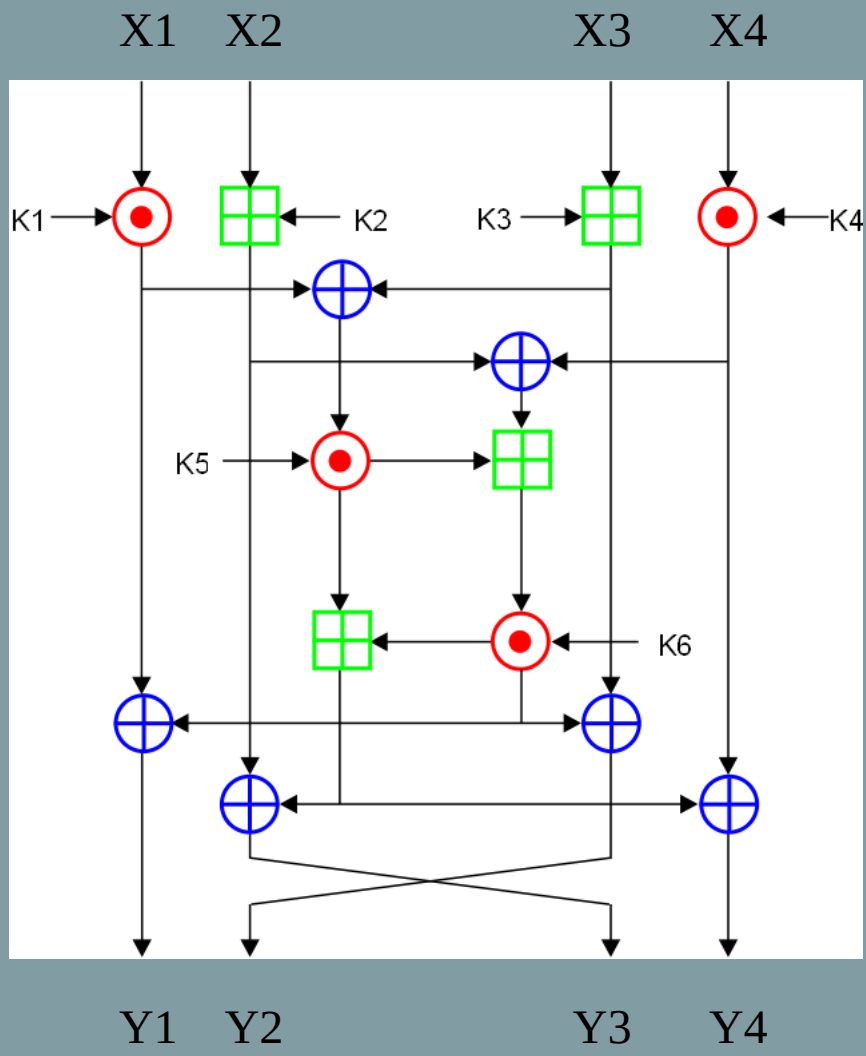
- 2.1. Un peu d’histoire.
- 2.2. Entropie et information.
- 2.3. Le DES.
- 2.4. L’AES.
- 2.5. Autres protocoles.
- 2.6. Conclusion.

- C’est l’un des autres finalistes du même concours que l’AES.
- Il traite des blocs de 128 bits avec une clef de 128, 192 ou 256 bits.
- Il comporte 32 rondes d’un réseau opérant sur 4 mots de 32 bits.
- Il utilise 8 S-boites qui alternent à chaque tour et ont été construites à partir de celles du DES.

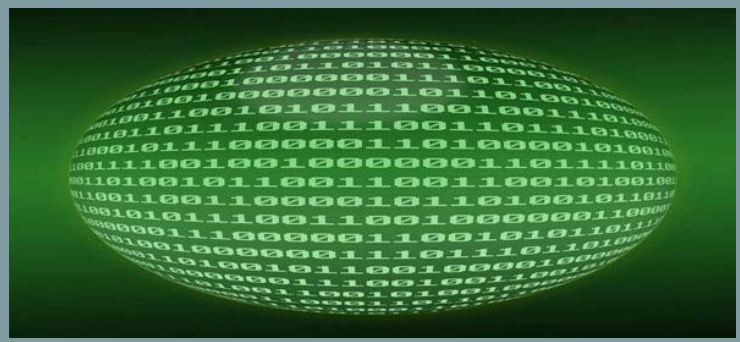

A j
B j
C j
D j
<<< 13
<<< 3
<< 3
<<< 1
<<< 7
<< 7
<<< 5
<<< 22
A j+1
B j+1
C j+1
D j+1

- Célèbre car utilisé dans le logiciel PGP.
- Il traite des blocs de 64 bits avec une clef de 128 bits.
- Il comporte 8 rondes.
- Le brouillage se fait par des ou exclusif, des additions et des multiplications dans un corps fini de grande taille.
- L’algorithme est très rapide.

- Squelette d'IDEA :
- Xi = sous bloc de 16 bits de texte en clair
- Yi = sous bloc de 16 bits de texte chiffré
- Ki = sous bloc de 16 bits de la clef
- ⊕ = ou exclusif bit à bit
- ⊞ = addition modulo 2^16
- ⊙ = addition modulo 2^16



- 2.1. Un peu d’histoire.
- 2.2. Entropie et information.
- 2.3. Le DES.
- 2.4. L’AES.
- 2.5. Autres protocoles.
- 2.6. Conclusion.



- Le débat sur l'école resurgit à nouveau :

  Public ou privé 

- Les algorithmes à clef secrète sont en gros 1000 fois plus rapides que ceux à clef publique.
- Les algorithmes à clef publique sont plus pratiques et plus sûrs (sans doute).

- On utilise en fait des cryptosystèmes hybrides : l'échange de clefs et les signatures se font par des algorithmes à clef publique tandis que le chiffrement se fait par des algorithmes à clef secrète.

- Un duo gagnant : **RSA ⊕ AES**

- La taille des clefs est cruciale dans les cryptosystèmes.

- Pour un niveau de sécurité égal, les algorithmes à clef publique nécessitent des clefs plus longues que les algorithmes à clef secrète.

La taille des clefs. (2)

Clef secrète			
En bits	En décimal	Mips / an	Année cryptanalyse
40	13	1	1995
56	17	1.00E+04	1998
64	20	1.00E+06	
128	39	1.00E+27	2100
256	78	1.00E+66	??

Clef publique			
En bits	En décimal	Mips / an	Année cryptanalyse
256	78	1	1985
512	155	8.00E+03	1999
768			2009
1024	309	1.00E+09	2030
2048	617	1.00E+19	2080
4096	1233	1.00E+31	??

Des questions ?

- Photo de couverture: © Adi Shamir.
- A compléter.
- .
- .
- .
- .
- .
- .
-
- .
- .
- Autres sources d’inspiration:
  - Cryptographie appliquée de Bruce Schneier, éditions Vuibert.
  - L’art du secret, in « dossier pour la science ».
  - Histoire des codes secrets, de Simon Singh, éditions livre de poche.