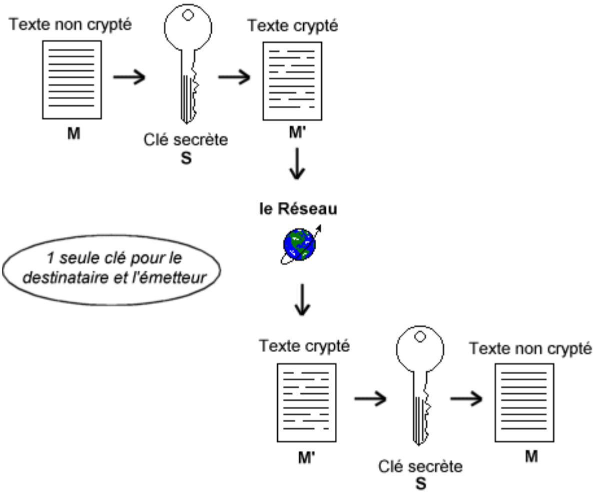
Chiffrement par bloc

January 28, 2010

Plan

Chiffrement à clef privée



Plan

Plan

Chiffrement par permutation général

- Même principe que la permutation classique.

Chiffrement par permutation général

- Même principe que la permutation classique.
- Pour un message M dont les lettres sont numéroté de 1 à n

$$M = m_1 m_2 m_3 \cdots m_n$$

Mais ici les m_i sont des bits.

Chiffrement par permutation général

- Même principe que la permutation classique.
- Pour un message M dont les lettres sont numéroté de 1 à n

$$M = m_1 m_2 m_3 \cdots m_n$$

  Mais ici les m_i sont des bits.
- Et une permutation (bijection) $P : \{1, \ldots, k\} \to \{1, \ldots, k\}$

Chiffrement par permutation général

- Même principe que la permutation classique.
- Pour un message M dont les lettres sont numéroté de 1 à n

$$M = m_1 m_2 m_3 \cdots m_n$$

Mais ici les m_i sont des bits.

- Et une permutation (bijection) $P : \{1, \ldots, k\} \to \{1, \ldots, k\}$
- On découpe M en bloc de k bits.

$$M = [M_1, M_2, \ldots, M_{n/k}] \text{ où } M_i \in \{0,1\}^k.$$

Chiffrement par permutation général

- Même principe que la permutation classique.
- Pour un message M dont les lettres sont numéroté de 1 à n

$$M = m_1 m_2 m_3 \cdots m_n$$

Mais ici les m_i sont des bits.

- Et une permutation (bijection) $P : \{1, \ldots, k\} \to \{1, \ldots, k\}$
- On découpe M en bloc de k bits.

$$M = [M_1, M_2, \ldots, M_{n/k}] \text{ où } M_i \in \{0, 1\}^k.$$

- On réordonne les lettres de M : à la place i on met le bit d'indice $P(i)$.

Chiffrement par permutation général

- Même principe que la permutation classique.
- Pour un message M dont les lettres sont numéroté de 1 à n

$$M = m_1 m_2 m_3 \cdots m_n$$

Mais ici les m_i sont des bits.

- Et une permutation (bijection) $P : \{1, \ldots, k\} \to \{1, \ldots, k\}$
- On découpe M en bloc de k bits.

$$M = [M_1, M_2, \ldots, M_{n/k}] \text{ où } M_i \in \{0,1\}^k.$$

- On réordonne les lettres de M : à la place i on met le bit d'indice $P(i)$.
- Le chiffrement d'un bloc de k bits est alors

$$C = m_{P(1)} m_{P(2)} m_{P(3)} \ldots m_{P(k)}.$$

Chiffrement par permutation général

- Même principe que la permutation classique.
- Pour un message M dont les lettres sont numéroté de 1 à n

$$M = m_1 m_2 m_3 \cdots m_n$$

Mais ici les m_i sont des bits.

- Et une permutation (bijection) $P : \{1, \ldots, k\} \to \{1, \ldots, k\}$
- On découpe M en bloc de k bits.

$$M = [M_1, M_2, \ldots, M_{n/k}] \text{ où } M_i \in \{0, 1\}^k.$$

- On réordonne les lettres de M : à la place i on met le bit d'indice $P(i)$.
- Le chiffrement d'un bloc de k bits est alors

$$C = m_{P(1)} m_{P(2)} m_{P(3)} \ldots m_{P(k)}.$$

- Le déchiffrement d'un bloc de k bits se fait en

$$c_{P^{-1}(1)}, c_{P^{-1}(2)}, \ldots, c_{P^{-1}(k)}.$$

Un exemple

- On considère $M = 1010001001011$

Un exemple

- On considère $M = 1010001001011$
- On représentera la permutation P soit avec un tableau

$$[P(1), P(2), P(3), \ldots, P(k)]$$

où encore par $P([m_1, \ldots, m_k]) = [m_{P(1)}, \ldots, m_{P(k)}]$.

- On considère $M = 1010001001011$
- On représentera la permutation P soit avec un tableau

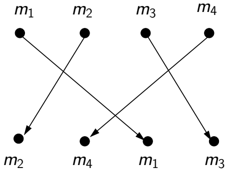
$$[P(1), P(2), P(3), \ldots, P(k)]$$

où encore par $P([m_1, \ldots, m_k]) = [m_{P(1)}, \ldots, m_{P(k)}]$.

- Ici nous prenons $P\colon \{1,2,3,4\} \to \{1,2,3,4\}$ donné par

$$[P(1), P(2), P(3), P(4)] = [3, 1, 4, 2]$$

où encore avec des fils



Un exemple

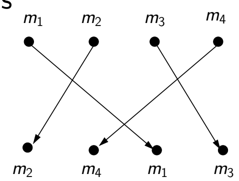
- On considère $M = 1010001001011$
- On représentera la permutation P soit avec un tableau

$$[P(1), P(2), P(3), \ldots, P(k)]$$

  où encore par $P([m_1, \ldots, m_k]) = [m_{P(1)}, \ldots, m_{P(k)}]$.
- Ici nous prenons $P\colon \{1,2,3,4\} \to \{1,2,3,4\}$ donné par

$$[P(1), P(2), P(3), P(4)] = [3,1,4,2]$$

  où encore avec des fils



- On décompose M en bloc de 4 bits et on permute chaque bloc avec P

Un exemple

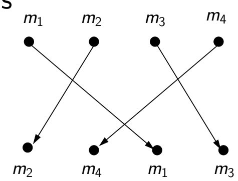
- On considère $M = 1010001001011$
- On représentera la permutation P soit avec un tableau

$$[P(1), P(2), P(3), \ldots, P(k)]$$

où encore par $P([m_1, \ldots, m_k]) = [m_{P(1)}, \ldots, m_{P(k)}]$.
- Ici nous prenons $P \colon \{1,2,3,4\} \to \{1,2,3,4\}$ donné par

$$[P(1), P(2), P(3), P(4)] = [3, 1, 4, 2]$$

où encore avec des fils



- On décompose M en bloc de 4 bits et on permute chaque bloc avec P

$$\begin{array}{cccc} M = & 1010| & 0010 & |0101 \\ & \downarrow & \downarrow & \downarrow \\ C & 1100 & 1000 & 0011 \end{array}$$

Plan

Chiffrement par subsitution, Exemple

Chiffrement par subsitution, Exemple

- Pour un message M dont les lettres sont numéroté de 1 à n

$$M = m_1 m_2 m_3 \cdots m_n$$

ici encore les m_i sont des bits.

Chiffrement par subsitution, Exemple

- Pour un message M dont les lettres sont numéroté de 1 à n

$$M = m_1 m_2 m_3 \cdots m_n$$

  ici encore les m_i sont des bits.
- Substitution S de bloc de k bits

$$S : \{0,1\}^k \rightarrow \{0,1\}^k$$

Chiffrement par subsitution, Exemple

- Pour un message M dont les lettres sont numéroté de 1 à n

$$M = m_1 m_2 m_3 \cdots m_n$$

ici encore les m_i sont des bits.

- Substitution S de bloc de k bits

$$S : \{0,1\}^k \rightarrow \{0,1\}^k$$

- On découpe M en bloc de k bits.

$$M = [M_1, M_2, \ldots, M_{n/k}] \text{ où } M_i \in \{0,1\}^k.$$

Chiffrement par subsitution, Exemple

- Pour un message M dont les lettres sont numéroté de 1 à n

$$M = m_1 m_2 m_3 \cdots m_n$$

ici encore les m_i sont des bits.

- Substitution S de bloc de k bits

$$S : \{0,1\}^k \rightarrow \{0,1\}^k$$

- On découpe M en bloc de k bits.

$$M = [M_1, M_2, \ldots, M_{n/k}] \text{ où } M_i \in \{0,1\}^k.$$

- On applique S à chaque bloc M_i de de M :

$$C = [S(M_1), S(M_2), \ldots, S(M_{n/k})]$$

Chiffrement par subsitution, Exemple

- Pour un message M dont les lettres sont numéroté de 1 à n

$$M = m_1 m_2 m_3 \cdots m_n$$

ici encore les m_i sont des bits.

- Substitution S de bloc de k bits

$$S : \{0,1\}^k \to \{0,1\}^k$$

- On découpe M en bloc de k bits.

$$M = [M_1, M_2, \ldots, M_{n/k}] \text{ où } M_i \in \{0,1\}^k.$$

- On applique S à chaque bloc M_i de de M :

$$C = [S(M_1), S(M_2), \ldots, S(M_{n/k})]$$

- Le déchiffrement de $C = (c_1, \ldots, c_n)$ se fait appliquant S^{-1} à chaque bloc de k bits de $C = [C_1, \ldots, C_{n/k}]$

$$M = [S^{-1}(C_1), \ldots, S^{-1}(C_{n/k})].$$

Exemple de chiffrement par substitution binaire

On considère

$$M = 10100010000$$

Et la subsitution S suivante

$$\begin{array}{|c|cccc|}
X & 00 & 01 & 10 & 11 \\
\downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\
S(X) & 11 & 10 & 01 & 00
\end{array}$$

Je décompose M en bloc de 2 bits et j'applique S à chacun des bloc

$$\begin{array}{c|}
\hline
M = \quad 10| \quad 10| \quad 01| \quad 00| \quad 00 \\
\hline
C = \quad 01| \quad 01| \quad 10| \quad 11| \quad 11 \\
\hline
\end{array}$$

Plan

Rappel : le ou exclusif $\bigoplus$

- On considère l'alphabet binaire $\Sigma = \{0, 1\}$.

Rappel : le ou exclusif $\bigoplus$

- On considère l'alphabet binaire $\Sigma = \{0, 1\}$.
- On notera par la suite $\oplus$ l'opérateur logique *ou exclusif*

$x \oplus y$

$x \backslash y$	0	1
0	0	1
1	1	0

Rappel : le ou exclusif $\bigoplus$

- On considère l'alphabet binaire $\Sigma = \{0, 1\}$.
- On notera par la suite $\oplus$ l'opérateur logique *ou exclusif*

$x \oplus y$

$x \backslash y$	0	1
0	0	1
1	1	0

- On étendra la notation $\oplus$ à des blocs de bit où on effectue le ou exculsif *bit à bit*

$$[0, 1, 0, 1, 1] \oplus [1, 1, 0, 0, 1] = [1, 0, 0, 1, 0].$$

Rappel : le ou exclusif $\bigoplus$

- On considère l'alphabet binaire $\Sigma = \{0, 1\}$.
- On notera par la suite $\oplus$ l'opérateur logique *ou exclusif*

$x \oplus y$

$x \backslash y$	0	1
0	0	1
1	1	0

- On étendra la notation $\oplus$ à des blocs de bit où on effectue le ou exculsif *bit à bit*

$$[{\color{red}0}, 1, 0, 1, 1] \oplus [{\color{red}1}, 1, 0, 0, 1] = [{\color{red}1}, 0, 0, 1, 0].$$

Le code de Vigenère en binaire

1. Un message M constitué d'une suite de bit et une clef en binaire

$$M = [m_1, m_2, \ldots, m_n] \quad = K = [k_1, \ldots, k_\ell] \quad m_i, k_i \in \{0, 1\}$$

2. On ajoute la clef lettre à lettre modulo 2 de façon répétitive.

$$\begin{array}{cccccccccc}
 & m_1 & m_2 & \cdots & m_\ell & m_{\ell+1} & \cdots & m_{2\ell} & m_{2\ell+1} & \cdots \\
\bigoplus & k_1 & k_2 & \cdots & k_\ell & k_1 & \cdots & k_\ell & k_1 & \cdots \\
\hline
 & c_1 & c_2 & \cdots & c_\ell & c_{\ell+1} & \cdots & c_{2\ell} & c_{2\ell+1} & \cdots
\end{array}$$

Exemple chiffrement de Vigenère en binaire

Soit la clef $K = 11011101$ de 8 bits.

- *Chiffrement.*

$$\begin{array}{ccrl}
Message ='' \ Oh'' & \overset{codageASCII}{\longrightarrow} & Message & 01111001, 01101000 \\
 & & \bigoplus Clef & 11011101, 11011101 \\ \hline
Chiffre ='' \ \tilde{n} \dashv'' & \overset{codageASCII}{\longleftarrow} & = & 10100100, 10110101
\end{array}$$

Exemple chiffrement de Vigenère en binaire

Soit la clef $K = 11011101$ de 8 bits.

- *Chiffrement.*

$$Message ='' Oh'' \xrightarrow{codageASCII} \begin{array}{rr} Message & 01111001, 01101000 \\ \bigoplus Clef & 11011101, 11011101 \\ \hline = & 10100100, 10110101 \end{array}$$

$$Chiffre ='' \tilde{n} \dashv'' \xleftarrow{codageASCII}$$

- *Déchiffrement.*

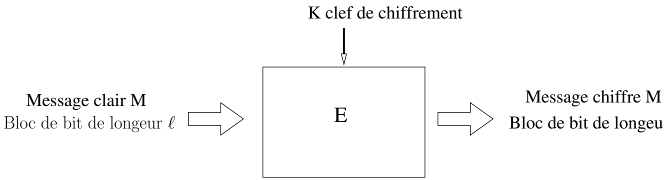
$$Chiffre ='' \grave{O} \sqsubseteq'' \xleftarrow{codageASCII} \begin{array}{rr} Chiffre & 10010101, 10111100 \\ \bigoplus Clef & 11011101, 11011101 \\ \hline = & 01001000, 01100001 \end{array}$$

$$Message ='' Ha'' \xrightarrow{codageASCII}$$

Plan

Fonction de chiffrement de bloc

- Un chiffrement par bloc de longueur ℓ



  c'est une fonction $E : \{0,1\}^\ell \times \mathcal{K} \to \{0,1\}^\ell$

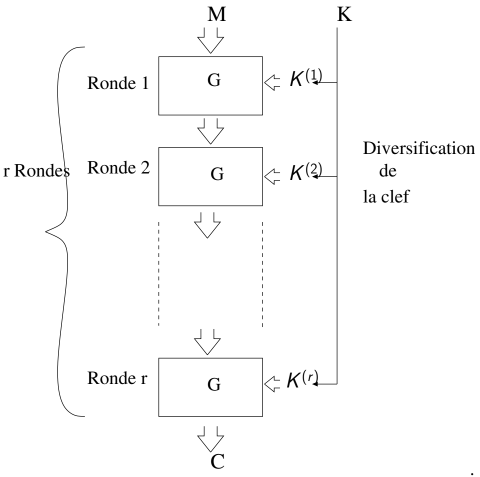
- On notera par la suite E_K la fonction

$$E_K : \begin{array}{rcl} \{0,1\}^\ell & \to & \{0,1\}^\ell \\ M & \mapsto & E_K(M) \end{array}$$

Décomposition en ronde

Les fonctions de chiffrement sont en général des itétations d'une fonction de ronde G

$$G : \{0,1\}^n \times \mathcal{K}' \longrightarrow \{0,1\}^n$$

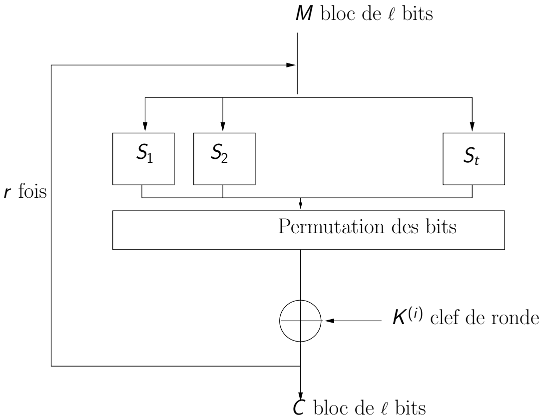


Plan

Réseau de substitution permutation (SPN)

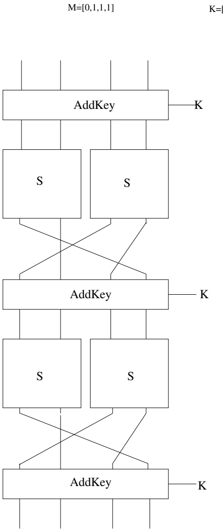
- Les chiffrements par bloc moderne sont héritiers des chiffrements classiques.
- Ils sont constitués d'une succession de
  - substitution,
  - permutation,
  - XOR bit à bit avec la clef (Vigenère).
- De tels chiffrements de blocs sont dit SPN (Substitution Permutation Network).

Réseau SPN



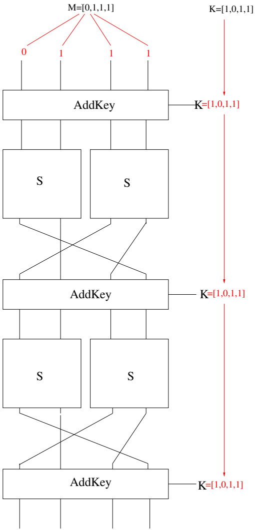
- Les boites S_i de substitution sont des fonctions $S_i\colon \{0,1\}^{\ell/t} \to \{0,1\}^{\ell/t}$
- $\ell' = \ell/t$ est suffisamment petit pour que S_i soit stocké sous forme de tableau.

Réseau SPN - Exemple de chiffrement



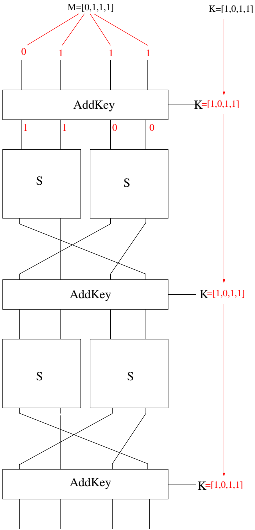
X	00	01	10	11
$S(X)$	10	11	00	01

Réseau SPN - Exemple de chiffrement



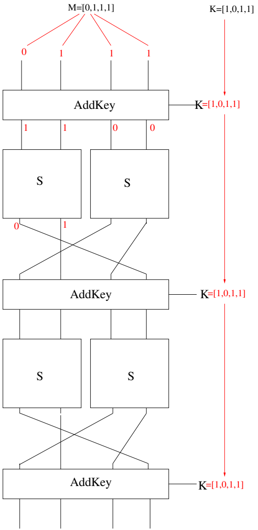
X	00	01	10	11
$S(X)$	10	11	00	01

Réseau SPN - Exemple de chiffrement



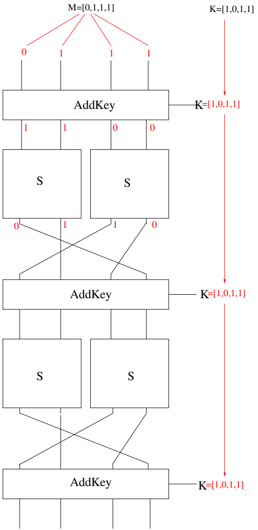
X	00	01	10	11
$S(X)$	10	11	00	01

Réseau SPN - Exemple de chiffrement



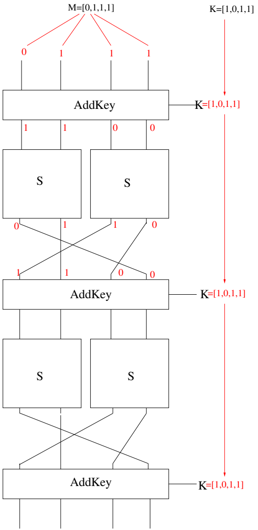
X	00	01	10	11
$S(X)$	10	11	00	01

Réseau SPN - Exemple de chiffrement



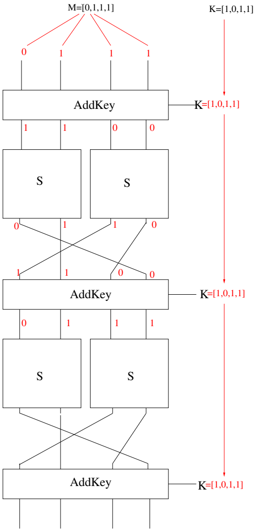
X	00	01	10	11
$S(X)$	10	11	00	01

Réseau SPN - Exemple de chiffrement



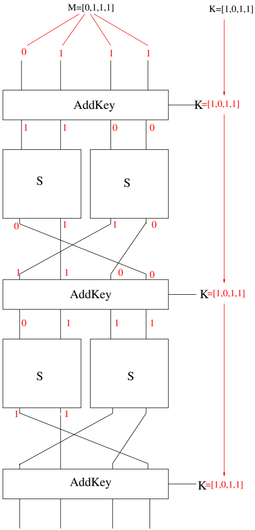
X	00	01	10	11
$S(X)$	10	11	00	01

Réseau SPN - Exemple de chiffrement



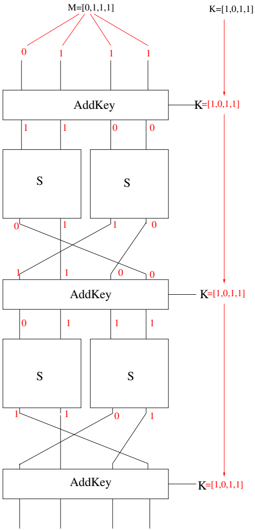
X	00	01	10	11
$S(X)$	10	11	00	01

Réseau SPN - Exemple de chiffrement



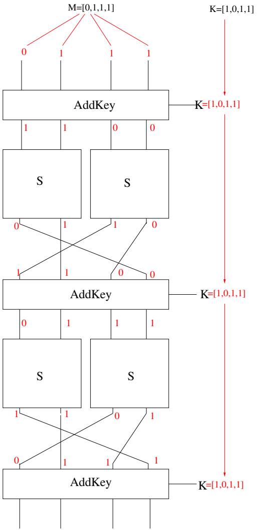
X	00	01	10	11
$S(X)$	10	11	00	01

Réseau SPN - Exemple de chiffrement



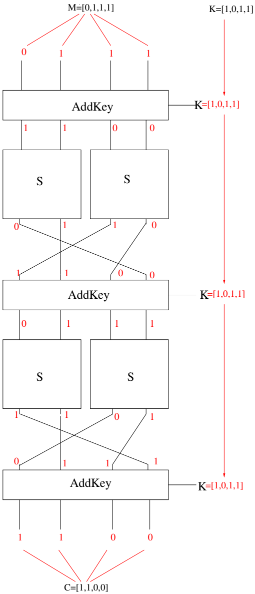
X	00	01	10	11
$S(X)$	10	11	00	01

Réseau SPN - Exemple de chiffrement



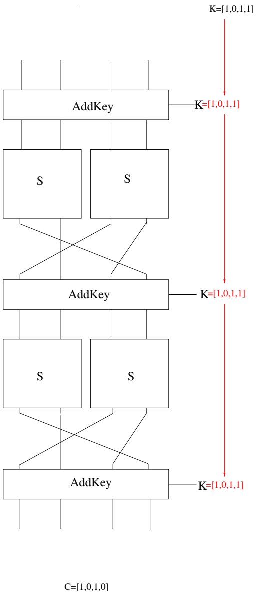
X	00	01	10	11
$S(X)$	10	11	00	01

Réseau SPN - Exemple de chiffrement



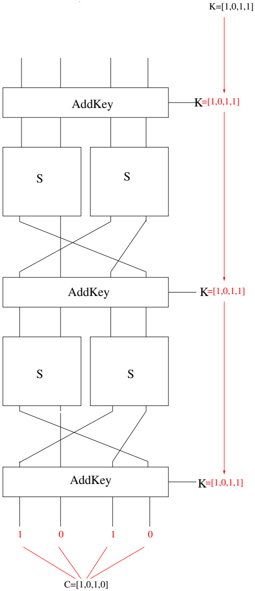
X	00	01	10	11
$S(X)$	10	11	00	01

Réseau SPN - Exemple de déchiffrement



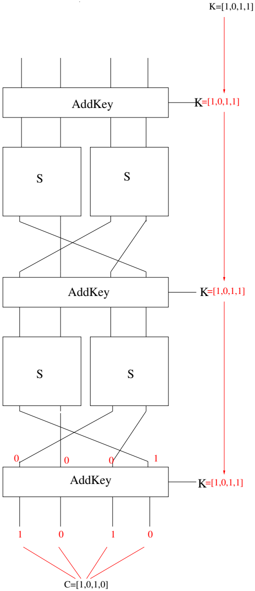
X	00	01	10	11
$S(X)$	10	11	00	01

Réseau SPN - Exemple de déchiffrement



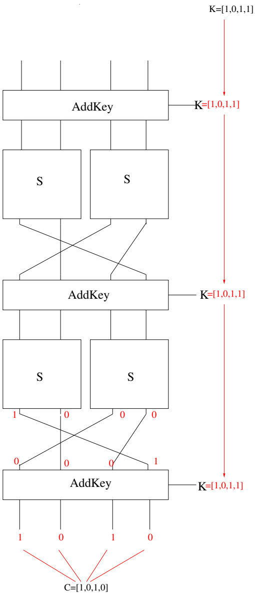
X	00	01	10	11
$S(X)$	10	11	00	01

Réseau SPN - Exemple de déchiffrement



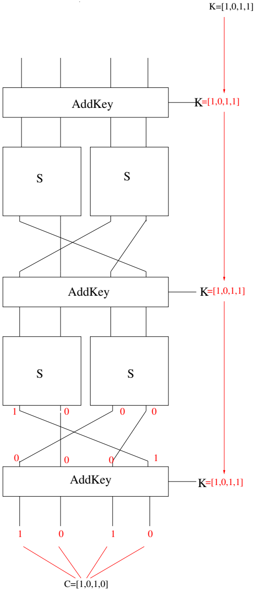
X	00	01	10	11
$S(X)$	10	11	00	01

Réseau SPN - Exemple de déchiffrement



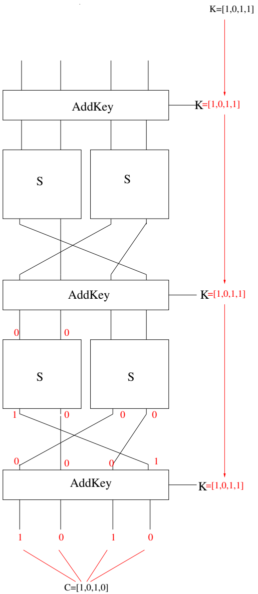
X	00	01	10	11
$S(X)$	10	11	00	01

Réseau SPN - Exemple de déchiffrement



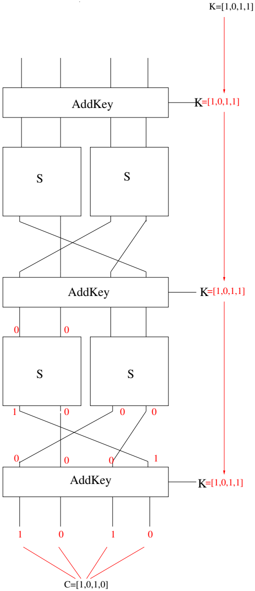
X	00	01	10	11
$S(X)$	10	11	00	01

Réseau SPN - Exemple de déchiffrement



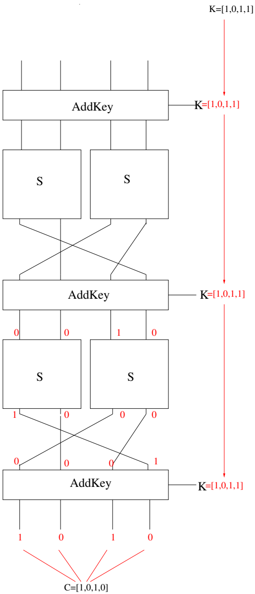
X	00	01	10	11
$S(X)$	10	11	00	01

Réseau SPN - Exemple de déchiffrement



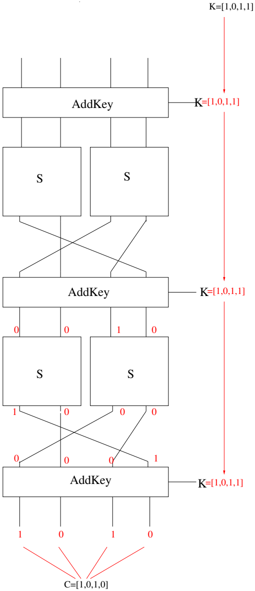
X	00	01	10	11
$S(X)$	10	11	00	01

Réseau SPN - Exemple de déchiffrement



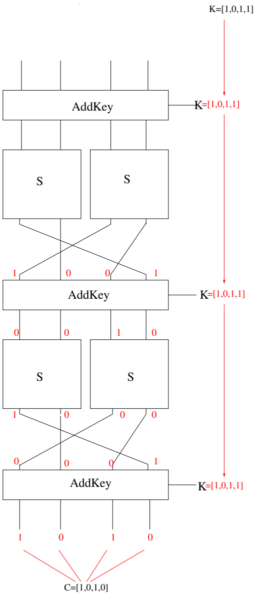
X	00	01	10	11
$S(X)$	10	11	00	01

Réseau SPN - Exemple de déchiffrement



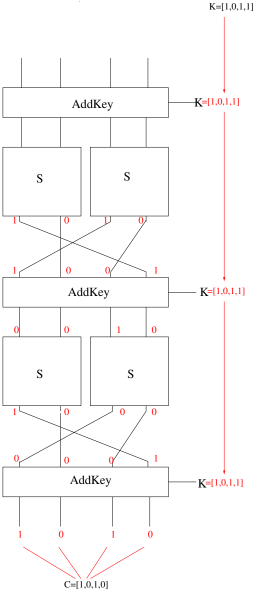
X	00	01	10	11
$S(X)$	10	11	00	01

Réseau SPN - Exemple de déchiffrement



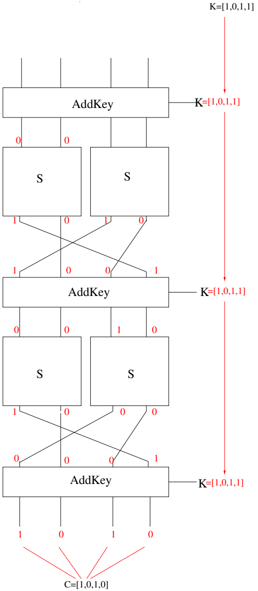
X	00	01	10	11
$S(X)$	10	11	00	01

Réseau SPN - Exemple de déchiffrement



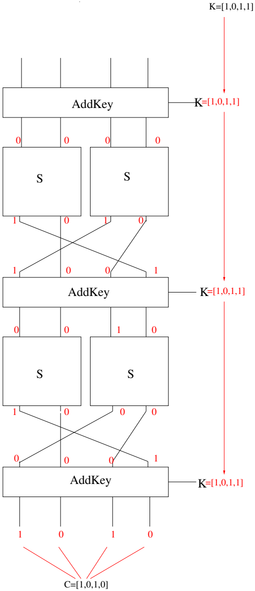
X	00	01	10	11
$S(X)$	10	11	00	01

Réseau SPN - Exemple de déchiffrement



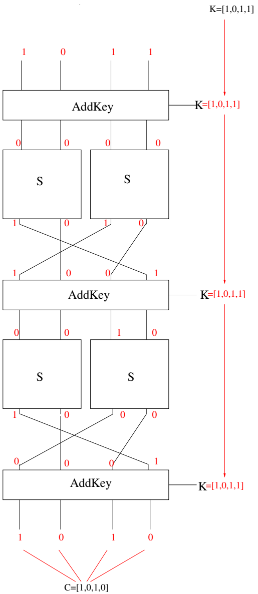
X	00	01	10	11
$S(X)$	10	11	00	01

Réseau SPN - Exemple de déchiffrement



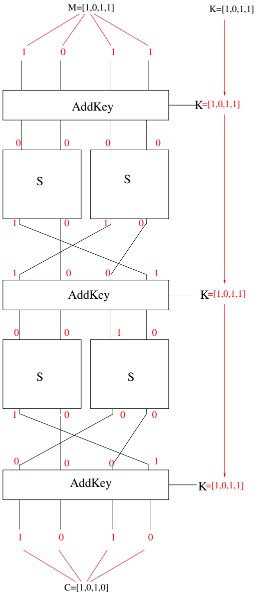
X	00	01	10	11
$S(X)$	10	11	00	01

Réseau SPN - Exemple de déchiffrement



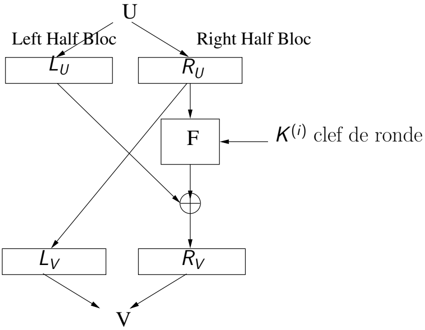
X	00	01	10	11
$S(X)$	10	11	00	01

Réseau SPN - Exemple de déchiffrement



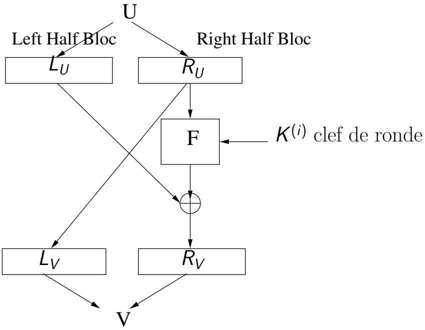
X	00	01	10	11
$S(X)$	10	11	00	01

- Une fonction de ronde suivant un schema de Feistel est comme suit



Shéma de Feistel

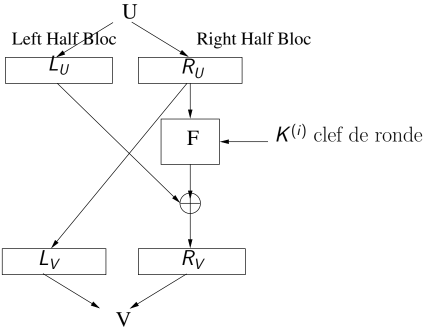
- Une fonction de ronde suivant un schema de Feistel est comme suit



- La fonction F est en général constituée de substitution permutation et d'XOR bit à bit avec la clef.

Shéma de Feistel

- Une fonction de ronde suivant un schema de Feistel est comme suit



- La fonction F est en général constituée de substitution permutation et d'XOR bit à bit avec la clef.
- C'est le cas des fonctions de ronde de la plupart des Block Cipher (DES,Twofish, Serpent,. . .).

DES = Data Encryption Standard.

$$DES\colon \{0,1\}^{64} \times \underbrace{\{0,1\}^{56}}_{\mathcal{K}} \to\colon \{0,1\}^{64}$$

Data Encryption Standard (DES)

- Il est basé sur Lucifer, conçu en 1971 par Horst Feistel et modifié par la NSA.

Data Encryption Standard (DES)

- Il est basé sur Lucifer, conçu en 1971 par Horst Feistel et modifié par la NSA.
- Il est adopté comme standard en 1977.

Data Encryption Standard (DES)

- Il est basé sur Lucifer, conçu en 1971 par Horst Feistel et modifié par la NSA.
- Il est adopté comme standard en 1977.
- Ses caractéristiques

Taille du bloc :	64 bits
Longueur de la clé :	56 bits
Structure :	schéma de Feistel
Nombre de rondes :	16 rondes

Data Encryption Standard (DES)

- Il est basé sur Lucifer, conçu en 1971 par Horst Feistel et modifié par la NSA.
- Il est adopté comme standard en 1977.
- Ses caractéristiques

Taille du bloc :	64 bits
Longueur de la clé :	56 bits
Structure :	schéma de Feistel
Nombre de rondes :	16 rondes

- Cassé par recherche exhaustive en 1998.

Data Encryption Standard (DES)

- Il est basé sur Lucifer, conçu en 1971 par Horst Feistel et modifié par la NSA.
- Il est adopté comme standard en 1977.
- Ses caractéristiques

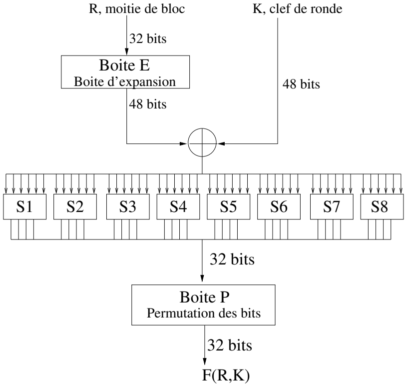
Taille du bloc :	64 bits
Longueur de la clé :	56 bits
Structure :	schéma de Feistel
Nombre de rondes :	16 rondes

- Cassé par recherche exhaustive en 1998.
- Utilisé aujourd'hui sous la forme de TripleDES :

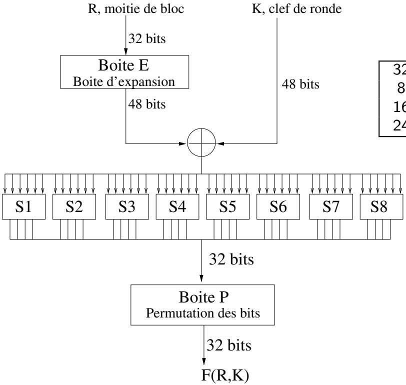
$$TripleDES_{[K_1,K_2]}(M) = DES_{K_1}^{-1} \circ DES_{K_2} \circ DES_{K_1}(M)$$

avec une clef de $K = [K_1, K_2]$ de 112 bits et M de 64 bits.

Exemple : fonction de ronde F du DES.



Exemple : fonction de ronde F du DES.



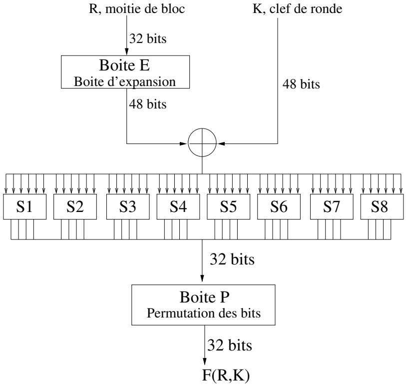
La boite d'expansion E permute et répète certains bits

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

Elle transforme un bloc de 32 bits $[b_1, \ldots, b_{32}]$ en un bloc de 48 bits

$$[b_{32}, b_1, b_2 \ldots, b_{31}, b_{32}, b_1]$$

Exemple : fonction de ronde F du DES.



La permutation P

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

En sortie les 32 bits $[b_1, \ldots, b_{32}]$ sont dans l'ordre

$$[b_{16}, b_7, \ldots, b_{25}]$$

Les boîtes S_i de la fonction F du DES

- Les boites S_i sont de la forme

S_1	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

Les boîtes S_i de la fonction F du DES

- Les boites S_i sont de la forme

S_1	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

- La valeur renvoyée $S_1([b_1, b_2, b_3, b_4, b_5, b_6])$ est la valeur se trouvant à l'intersection

$$[b_1, b_2, b_3, b_4, b_5, b_6] \rightarrow \begin{cases} \text{ligne d'indice } b_1 b_6 \\ \text{colonne d'indice } b_2 b_3 b_4 b_5 \end{cases}$$

Les boîtes S_i de la fonction F du DES

- Les boites S_i sont de la forme

S_1	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

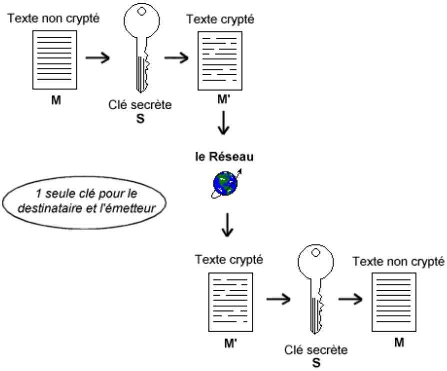
- La valeur renvoyée $S_1([b_1, b_2, b_3, b_4, b_5, b_6])$ est la valeur se trouvant à l'intersection

$$[b_1, b_2, b_3, b_4, b_5, b_6] \rightarrow \begin{cases} \text{ligne d'indice } b_1 b_6 \\ \text{colonne d'indice } b_2 b_3 b_4 b_5 \end{cases}$$

- Par exemple $S_1([0, 1, 0, 1, 0, 1]) = 10$, c'est la valeur à l'intersection de la ligne $[01]_2 = [1]_{10}$ et de la colonne $[1010]_2 = [10]_{10}$

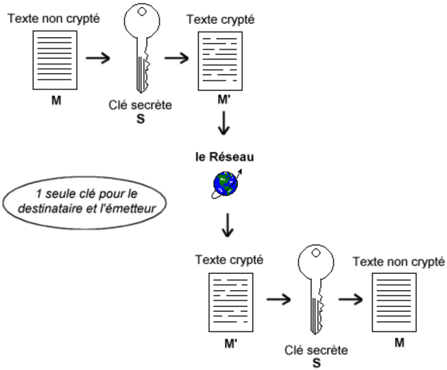
Plan

Chiffrement de message de longueur variable



- Pour l'instant on ne sait chiffrer et déchiffrer que des bloc de longueur ℓ bits.

Chiffrement de message de longueur variable



- Pour l'instant on ne sait chiffrer et déchiffrer que des bloc de longueur ℓ bits.
- On veut pouvoir chiffrer des messages de longueur arbitraire.

Bourrage et décomposition en bloc

- On a une fonction de chiffrement de bloc

$$E_K \colon \{0,1\}^\ell \to \{0,1\}^\ell$$

- Un message M de longueur arbitraire en bits $|M|$
- On *bourre* le message M afin qu'il ait une longueur multiple de ℓ

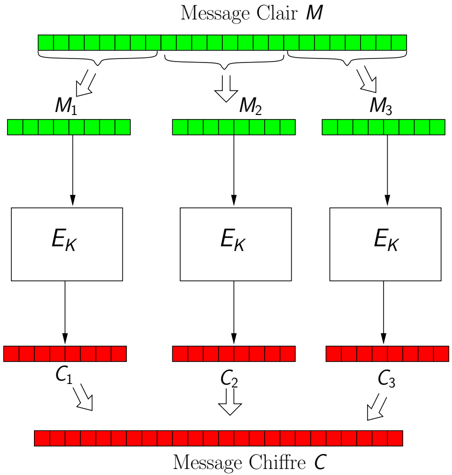
$$\bar{M} \leftarrow M || 10 \ldots 0$$

ou le nombre de zéro permet d'avoir ℓ divise $|\bar{M}|$.
- On décompose $\bar{M}$ en bloc de ℓ bits.

$$\bar{M} = [M_1, \ldots, M_i, \ldots].$$

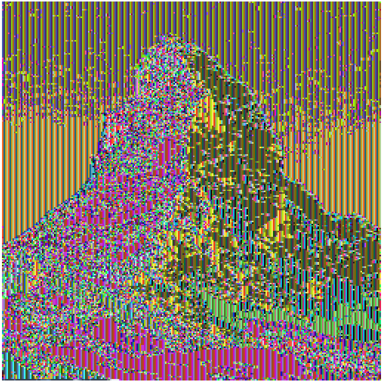
Chiffrement en Mode ECB (Electronic Code Book)

Ce mode consiste simplement à appliquer E_K à chaque bloc M_i.

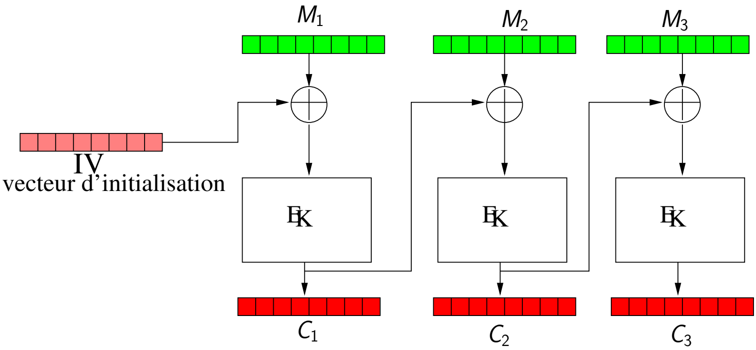


Défaut du mode ECB

- Le défaut majeur du mode ECB : il laisse passer de l'information.
- Un même bloc chiffré à deux instants différents sera toujours chiffré de la même manière.

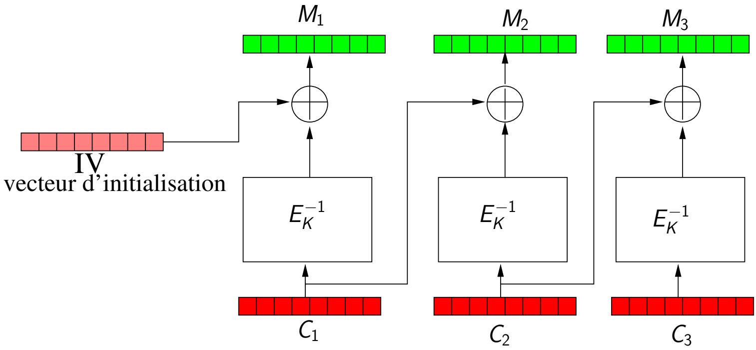


Mode CBC - Chiffrement



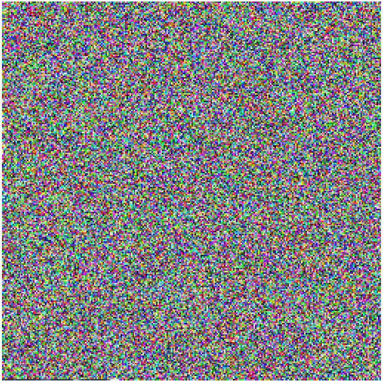
CBC = Cipher Block Chaining,

Mode CBC - Déchiffrement

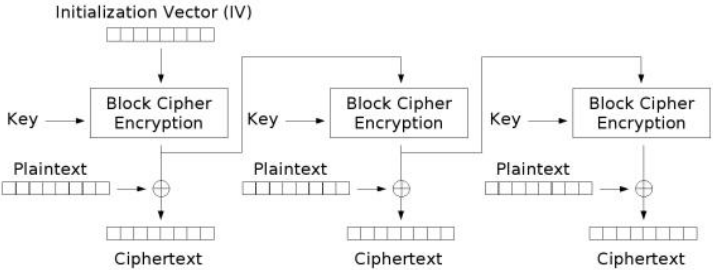


IV = Initial Value qui est choisie aléatoirement à chaque chiffrement.

Avantage de CBC

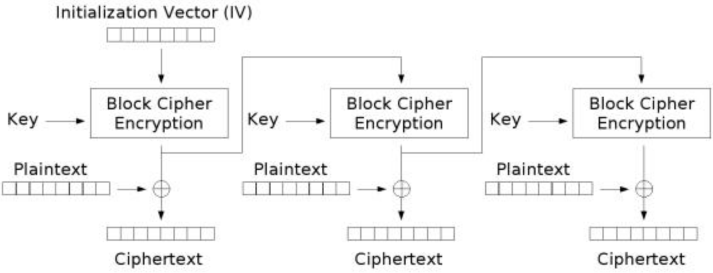


Autre mode de chiffrement : Mode OFB - chiffrement



Output Feedback (OFB) mode encryption

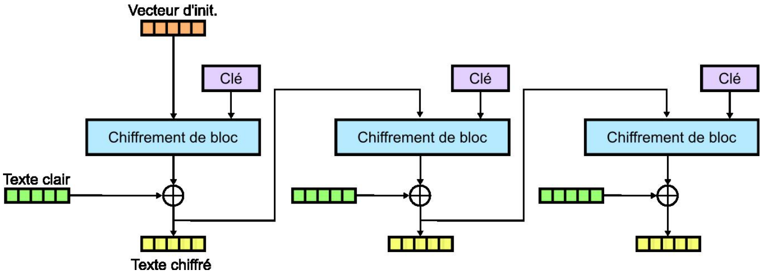
Autre mode de chiffrement : Mode OFB - chiffrement



Output Feedback (OFB) mode encryption

Un autre mode similaire : le mode CTR (mode CounTeuR). A partir d'un compteur on génère une suite de bloc pseudo aléatoire, et on fait un XOR avec le message.

Autre mode de chiffrement : Mode CFB (Cipher FeedBack mode) - chiffrement



Plan

Plan

Présentation.

- Le Block Cipher Rijndael (proposé par Rijmen et Daemen) a été accepté comme standard de chiffrement en 1999 par le NIST.
- Il existe trois version d'AES

	taille des blocs	taille de la clef
AES-128	128 bits	128 bits
AES-192	128 bits	192 bits
AES-256	128 bits	256 bits

- AES est un SPN (avec une permutation particulière).

Descriptions plus précise des composant d'AES

1. La boite S de substitution.
2. L'opération de Mix-Column.

Ces deux opérations utilisent des opérations sur les polynomes binaires.

Plan

Le corps des binaires

- Un corps c'est un ensemble dans lequel on sait
    - multiplier,
    - addition/soustraire des éléments
    - Et tout élément non nul est inversible.

Le corps des binaires

- Un corps c'est un ensemble dans lequel on sait
  - multiplier,
  - addition/soustraire des éléments
  - Et tout élément non nul est inversible.
- Le corps binaire $\mathbb{F}_2$ est l'ensemble $\{0, 1\}$ muni des lois d'addition et de multiplication suivante

$+$	0	1
0	0	1
1	1	0

$\times$	0	1
0	0	1
1	1	0

Polynôme binaire.

- Un polynôme binaire $A(X)$ c'est une somme formelle en une indéterminée X avec des coefficients dans $\mathbb{F}_2$

$$A(X) = a_0 + a_1 X + a_2 X + a_3 X^2 + \ldots + a_n X^n.$$

avec $a_i \in \mathbb{F}_2$

Polynôme binaire.

- Un polynôme binaire $A(X)$ c'est une somme formelle en une indéterminée X avec des coefficients dans $\mathbb{F}_2$

$$A(X) = a_0 + a_1X + a_2X + a_3X^2 + \ldots + a_nX^n.$$

avec $a_i \in \mathbb{F}_2$

- Exemple. $A(X) = 1 + X^3 + X^7$ est un polynôme binaire.

Polynôme binaire.

- Un polynôme binaire $A(X)$ c'est une somme formelle en une indéterminée X avec des coefficients dans $\mathbb{F}_2$

$$A(X) = a_0 + a_1X + a_2X + a_3X^2 + \ldots + a_nX^n.$$

  avec $a_i \in \mathbb{F}_2$
- Exemple. $A(X) = 1 + X^3 + X^7$ est un polynôme binaire.
- Contre exemples : les polynômes suivant ne sont pas des polynômes binaires

$$\begin{aligned} A(X) &= 2 + X^3 + X^7 \\ B(X) &= 1 + 1,1X + X^8 \end{aligned}$$

Arithmétique des polynomes binaires

. Ce sont des opération classiques sur les polynômes : seule différences les opérations sur les coefficients sont fait dans $\mathbb{F}_2$

$$A(X) = \sum_{i=0}^{n} a_i X^i), \quad B = (\sum_{i=0}^{n} b_i X^i)$$

Arithmétique des polynomes binaires

. Ce sont des opération classiques sur les polynômes : seule différences les opérations sur les coefficients sont fait dans $\mathbb{F}_2$

$$A(X) = \sum_{i=0}^{n} a_i X^i), \quad B = (\sum_{i=0}^{n} b_i X^i)$$

- *Addition.*

$$\begin{array}{rcl} A(X) + B(X) & = & (\sum_{i=0}^{n} a_i X^i) + (\sum_{i=0}^{n} b_i X^i) \\ & = & \sum_{i=0}^{n} \underbrace{(a_i + b_i)}_{\text{addition dans } \mathbb{F}_2} X^i \end{array}$$

Arithmétique des polynomes binaires

. Ce sont des opération classiques sur les polynômes : seule différences les opérations sur les coefficients sont fait dans $\mathbb{F}_2$

$$A(X) = \sum_{i=0}^{n} a_i X^i), \quad B = (\sum_{i=0}^{n} b_i X^i)$$

- *Addition.*

$$\begin{array}{rcl} A(X) + B(X) & = & (\sum_{i=0}^{n} a_i X^i) + (\sum_{i=0}^{n} b_i X^i) \\ & = & \sum_{i=0}^{n} \quad \underbrace{(a_i + b_i)} \quad X^i \end{array}$$

Arithmétique des polynomes binaires

. Ce sont des opération classiques sur les polynômes : seule différences les opérations sur les coefficients sont fait dans $\mathbb{F}_2$

$$A(X) = \sum_{i=0}^{n} a_i X^i), \quad B = (\sum_{i=0}^{n} b_i X^i)$$

- *Addition.*

$$\begin{array}{rcl} A(X) + B(X) & = & (\sum_{i=0}^{n} a_i X^i) + (\sum_{i=0}^{n} b_i X^i) \\ & = & \sum_{i=0}^{n} \quad \underbrace{(a_i + b_i)} \quad X^i \end{array}$$

Exemple :

$$(1 + X + X^4 + X^5) + (X + X^2 + X^5 + X^6)$$

Arithmétique des polynomes binaires

. Ce sont des opération classiques sur les polynômes : seule différences les opérations sur les coefficients sont fait dans $\mathbb{F}_2$

$$A(X) = \sum_{i=0}^{n} a_i X^i), \quad B = (\sum_{i=0}^{n} b_i X^i)$$

- *Addition.*

$$\begin{array}{rcl} A(X) + B(X) & = & (\sum_{i=0}^{n} a_i X^i) + (\sum_{i=0}^{n} b_i X^i) \\ & = & \sum_{i=0}^{n} \quad \underbrace{(a_i + b_i)} \quad X^i \end{array}$$

Exemple :

$$\begin{array}{l} (1 + X + X^4 + X^5) + (X + X^2 + X^5 + X^6) \\ = (1+0) + (1+1)X + (0+1)X^2 + (0+0)X^3 + (0+0)X^4 \\ \qquad +(1+1)X^5 + (0+1)X^6 \end{array}$$

Arithmétique des polynomes binaires

. Ce sont des opération classiques sur les polynômes : seule différences les opérations sur les coefficients sont fait dans $\mathbb{F}_2$

$$A(X) = \sum_{i=0}^{n} a_i X^i), \quad B = (\sum_{i=0}^{n} b_i X^i)$$

- *Addition.*

$$\begin{array}{rcl} A(X) + B(X) & = & (\sum_{i=0}^{n} a_i X^i) + (\sum_{i=0}^{n} b_i X^i) \\ & = & \sum_{i=0}^{n} \quad \underbrace{(a_i + b_i)} \quad X^i \end{array}$$

Exemple :

$$\begin{array}{l} (1 + X + X^4 + X^5) + (X + X^2 + X^5 + X^6) \\ = (1+0) + (1+1)X + (0+1)X^2 + (0+0)X^3 + (0+0)X^4 \\ \qquad + (1+1)X^5 + (0+1)X^6 \\ = 1 + X^2 + X^6 \end{array}$$

Arithmétique des polynômes binaires

.

- *Multiplication.* Fonctionne comme une multiplication classique : développement et addition.

$$A(X) \times B(X) = \sum_{i,j=0}^{n} a_i b_j X^{i+j}$$

Arithmétique des polynômes binaires

.

- *Multiplication.* Fonctionne comme une multiplication classique : développement et addition.

$$A(X) \times B(X) = \sum_{i,j=0}^{n} a_i b_j X^{i+j}$$

Exemple en développant par rapport à $A(X)$:

$$(1 + X + X^5) \times (X + X^2 + X^5 + X^6)$$

Arithmétique des polynômes binaires

.

- *Multiplication.* Fonctionne comme une multiplication classique : développement et addition.

$$A(X) \times B(X) = \sum_{i,j=0}^{n} a_i b_j X^{i+j}$$

Exemple en développant par rapport à $A(X)$:

$$\begin{aligned}
&(1 + X + X^5) \times (X + X^2 + X^5 + X^6) \\
&= (X + X^2 + X^5 + X^6) + X(X + X^2 + X^5 + X^6) \\
&\qquad + X^5(X + X^2 + X^5 + X^6)
\end{aligned}$$

Arithmétique des polynômes binaires

.

- *Multiplication.* Fonctionne comme une multiplication classique : développement et addition.

$$A(X) \times B(X) = \sum_{i,j=0}^{n} a_i b_j X^{i+j}$$

Exemple en développant par rapport à $A(X)$:

$$\begin{aligned}
&(1 + X + X^5) \times (X + X^2 + X^5 + X^6) \\
&= (X + X^2 + X^5 + X^6) + X(X + X^2 + X^5 + X^6) \\
&\qquad + X^5(X + X^2 + X^5 + X^6) \\
&= (X + X^2 + X^5 + X^6) + (X^2 + X^3 + X^6 + X^7) \\
&\qquad + (X^6 + X^7 + X^{10} + X^{11})
\end{aligned}$$

Arithmétique des polynômes binaires

.

- *Multiplication.* Fonctionne comme une multiplication classique : développement et addition.

$$A(X) \times B(X) = \sum_{i,j=0}^{n} a_i b_j X^{i+j}$$

Exemple en développant par rapport à $A(X)$:

$$\begin{aligned}
&(1 + X + X^5) \times (X + X^2 + X^5 + X^6) \\
&= (X + X^2 + X^5 + X^6) + X(X + X^2 + X^5 + X^6) \\
&\qquad\qquad\qquad\qquad + X^5(X + X^2 + X^5 + X^6) \\
&= (X + X^2 + X^5 + X^6) + (X^2 + X^3 + X^6 + X^7) \\
&\qquad\qquad\qquad\qquad + (X^6 + X^7 + X^{10} + X^{11}) \\
&= X + X^3 + X^5 + X^6 + X^{10} + X^{11}
\end{aligned}$$

Division Euclidienne

- Soit $A(X)$ et $B(X)$ deux polynômes binaires.
- Effectuer la division euclidienne de A par B consiste à trouver deux polynômes Q et R tel que

$$A = Q \times B + R \quad \text{où } \deg R < \deg B$$

- Exemple :

$$\begin{array}{l|l} X^8 + X^5 + X^2 + X & X^4 + X^2 + 1 \\ \hline & \end{array}$$

Division Euclidienne

- Soit $A(X)$ et $B(X)$ deux polynômes binaires.
- Effectuer la division euclidienne de A par B consiste à trouver deux polynômes Q et R tel que

$$A = Q \times B + R \quad \text{où } \deg R < \deg B$$

- Exemple :

$$\begin{array}{l|l} X^8 + X^5 + X^2 + X & X^4 + X^2 + 1 \\ \hline & X^4 \end{array}$$

Division Euclidienne

- Soit $A(X)$ et $B(X)$ deux polynômes binaires.
- Effectuer la division euclidienne de A par B consiste à trouver deux polynômes Q et R tel que

$$A = Q \times B + R \quad \text{où } \deg R < \deg B$$

- Exemple :

$$\begin{array}{rl|l}
 & X^8 + X^5 + X^2 + X & X^4 + X^2 + 1 \\ \cline{3-3}
+ & X^8 + X^6 + X^4 & X^4
\end{array}$$

Division Euclidienne

- Soit $A(X)$ et $B(X)$ deux polynômes binaires.
- Effectuer la division euclidienne de A par B consiste à trouver deux polynômes Q et R tel que

$$A = Q \times B + R \quad \text{où } \deg R < \deg B$$

- Exemple :

$$\begin{array}{rl|l}
 & X^8 + X^5 + X^2 + X & X^4 + X^2 + 1 \\ \cline{3-3}
+ & X^8 + X^6 + X^4 & X^4 \\ \cline{2-2}
 & X^6 + X^5 + X^4 + X^2 + X &
\end{array}$$

Division Euclidienne

- Soit $A(X)$ et $B(X)$ deux polynômes binaires.
- Effectuer la division euclidienne de A par B consiste à trouver deux polynômes Q et R tel que

$$A = Q \times B + R \quad \text{où } \deg R < \deg B$$

- Exemple :

$$\begin{array}{rl|l}
 & X^8 + X^5 + X^2 + X & X^4 + X^2 + 1 \\ \cline{3-3}
+ & \underline{X^8 + X^6 + X^4} & X^4 + X^2 \\
 & X^6 + X^5 + X^4 + X^2 + X &
\end{array}$$

Division Euclidienne

- Soit $A(X)$ et $B(X)$ deux polynômes binaires.
- Effectuer la division euclidienne de A par B consiste à trouver deux polynômes Q et R tel que

$$A = Q \times B + R \quad \text{où } \deg R < \deg B$$

- Exemple :

$$\begin{array}{rl|l}
 & X^8 + X^5 + X^2 + X & X^4 + X^2 + 1 \\ \cline{3-3}
+ & \underline{X^8 + X^6 + X^4} & X^4 + X^2 \\
 & \quad X^6 + X^5 + X^4 + X^2 + X & \\
+ & \quad X^6 + X^4 + X^2 &
\end{array}$$

- Soit $A(X)$ et $B(X)$ deux polynômes binaires.
- Effectuer la division euclidienne de A par B consiste à trouver deux polynômes Q et R tel que

$$A = Q \times B + R \quad \text{où } \deg R < \deg B$$

- Exemple :

$$\begin{array}{rl|l}
 & X^8 + X^5 + X^2 + X & X^4 + X^2 + 1 \\ \cline{3-3}
+ & X^8 + X^6 + X^4 & X^4 + X^2 \\ \cline{2-2}
 & X^6 + X^5 + X^4 + X^2 + X & \\
+ & X^6 + X^4 + X^2 & \\ \cline{2-2}
 & X^5 + X &
\end{array}$$

Division Euclidienne

- Soit $A(X)$ et $B(X)$ deux polynômes binaires.
- Effectuer la division euclidienne de A par B consiste à trouver deux polynômes Q et R tel que

$$A = Q \times B + R \quad \text{où } \deg R < \deg B$$

- Exemple :

$$
\begin{array}{rl|l}
 & X^8 + X^5 + X^2 + X & X^4 + X^2 + 1 \\
\cline{3-3}
+ & X^8 + X^6 + X^4 & X^4 + X^2 + X \\
\cline{2-2}
 & \quad X^6 + X^5 + X^4 + X^2 + X & \\
+ & \quad X^6 + X^4 + X^2 & \\
\cline{2-2}
 & \qquad X^5 + X & \\
\end{array}
$$

Division Euclidienne

- Soit $A(X)$ et $B(X)$ deux polynômes binaires.
- Effectuer la division euclidienne de A par B consiste à trouver deux polynômes Q et R tel que

$$A = Q \times B + R \quad \text{où } \deg R < \deg B$$

- Exemple :

$$\begin{array}{rl|l}
 & X^8 + X^5 + X^2 + X & X^4 + X^2 + 1 \\ \cline{3-3}
+ & \underline{X^8 + X^6 + X^4} & X^4 + X^2 + X \\
 & X^6 + X^5 + X^4 + X^2 + X & \\
+ & \underline{X^6 + X^4 + X^2} & \\
 & X^5 + X & \\
+ & X^5 + X^3 + X & \\
\end{array}$$

Division Euclidienne

- Soit $A(X)$ et $B(X)$ deux polynômes binaires.
- Effectuer la division euclidienne de A par B consiste à trouver deux polynômes Q et R tel que

$$A = Q \times B + R \quad \text{où } \deg R < \deg B$$

- Exemple :

$$
\begin{array}{rl|l}
 & X^8 + X^5 + X^2 + X & X^4 + X^2 + 1 \\
\cline{3-3}
+ & \underline{X^8 + X^6 + X^4} & X^4 + X^2 + X \\
 & X^6 + X^5 + X^4 + X^2 + X & \\
+ & \underline{X^6 + X^4 + X^2} & \\
 & X^5 + X & \\
+ & \underline{X^5 + X^3 + X} & \\
 & X^3 &
\end{array}
$$

Division Euclidienne

- Soit $A(X)$ et $B(X)$ deux polynômes binaires.
- Effectuer la division euclidienne de A par B consiste à trouver deux polynômes Q et R tel que

$$A = Q \times B + R \quad \text{où } \deg R < \deg B$$

- Exemple :

$$
\begin{array}{l|l}
\ X^8 + X^5 + X^2 + X & X^4 + X^2 + 1 \\ \cline{2-2}
+\ \underline{X^8 + X^6 + X^4} & X^4 + X^2 + X \\
\ X^6 + X^5 + X^4 + X^2 + X & \\
+\ \underline{X^6 + X^4 + X^2} & \\
\ X^5 + X & \\
+\ \underline{X^5 + X^3 + X} & \\
\ X^3 &
\end{array}
$$

Le quotient $Q(X)$

Le reste $R(X)$

Arithmétique modulaire des polynômes binaires

Soit $P(X)$ un polynôme binaire de degré n.
Soit $A(X), B(X)$ de degré ¡ $\deg P = n$.

Arithmétique modulaire des polynômes binaires

Soit $P(X)$ un polynôme binaire de degré n.
Soit $A(X), B(X)$ de degré ¡ $\deg P = n$.

- Addition modulo P, consiste simplement à calculer $A(X) + B(X)$.

Arithmétique modulaire des polynômes binaires

Soit $P(X)$ un polynôme binaire de degré n.
Soit $A(X), B(X)$ de degré ¡ $\deg P = n$.

- Addition modulo P, consiste simplement à calculer $A(X) + B(X)$.
- Multiplication modulo P, consiste à calculer $R(X)$ tels que

$$\begin{aligned} C(X) &= A(X) \times B(X) \\ C(X) &= Q(X)P(X) + R(X) \text{ tels que } \deg R < \deg P \end{aligned}$$

On note $R(X) = A(X) \times B(X) \mod P(X)$.

Arithmétique modulaire des polynômes binaires

Soit $P(X)$ un polynôme binaire de degré n.
Soit $A(X), B(X)$ de degré ¡ $\deg P = n$.

- Addition modulo P, consiste simplement à calculer $A(X) + B(X)$.
- Multiplication modulo P, consiste à calculer $R(X)$ tels que

$$\begin{array}{rcl} C(X) & = & A(X) \times B(X) \\ C(X) & = & Q(X)P(X) + R(X) \text{ tels que } \deg R < \deg P \end{array}$$

  On note $R(X) = A(X) \times B(X) \mod P(X)$.
- Inversion modulo P. Consiste à trouver $A'(X)$ tel que

$$A(X) \times A'(X) = 1 \mod P$$

  On note souvent A^{-1} l'inverse de A modulo P.

Plan

La boite de substitution S d'AES

- Soit $A = [a_7, \ldots, a_1, a_0]$ un mot de 8 bits.
- Soit $P = X^8 + X^4 + X^3 + X + 1$.
- On définit $A(X) = a_0 + a_1 X + \ldots + a_7 X^7$.
- On calcule $A' = [a'0, a'_1, \ldots, a'_8]$ les coefficients de $A'(X)$ l'inverse de $A(X)$ modulo P.
- On cacule enfin (opération matricielle dans $\mathbb{F}_2$)

$$S(A) = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} a'_0 \\ a'_1 \\ a'_2 \\ a'_3 \\ a'_4 \\ a'_5 \\ a'_6 \\ a'_7 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

L'opération de MixColumn d'AES

- On trasforme une colonne d'octet en transformant chaque coefficient en polynome binaire.

$$\begin{bmatrix} A_0 \\ A_1 \\ A_2 \\ A_3 \end{bmatrix} \longrightarrow \begin{bmatrix} A_0(X) \\ A_1(X) \\ A_2(X) \\ A_3(X) \end{bmatrix}$$

- On effectue ensuite le produit matriciel suivant

$$MC(\begin{bmatrix} A_0 \\ A_1 \\ A_2 \\ A_3 \end{bmatrix}) = \begin{bmatrix} X & X+1 & 1 & 1 \\ 1 & X & X+1 & 1 \\ 1 & 1 & X & X+1 \\ X+1 & 1 & 1 & X \end{bmatrix} \cdot \begin{bmatrix} A_0(X) \\ A_1(X) \\ A_2(X) \\ A_3(X) \end{bmatrix} \mod F$$

Quelques références

- Le document décrivant Rijndael soumis au concours de NIST par Rijmen et Daemen

  http : //www.daimi.au.dk/~ivan/rijndael.pdf

- L'animation détaillant l'execution d'AES.

  http : //www.formaestudio.com/rijndaelinspector/