Théorie de l'information 3^e année - Cryptographie

Vincent Migliore

vincent.migliore@insa-toulouse.fr

INSA-TOULOUSE / LAAS-CNRS

Definition

La cryptographie est la science relative à l'écriture des secret. Vient des mots grecs kruptós (couvert, caché) et graphein (écrire, peindre).

Cette science est corrélée à l'évolution des civilisations, organisées en tribus, royaumes, pays, souhaitant protéger des écrits.

Cryptologie $\neq$ Cryptographie $\neq$ Cryptanalyse

- Cryptanalyse est la science pour casser les secrets.
- La cryptographie et la cryptanalyse font partie de la cryptologie.

Definition

La cryptographie est la science relative à l'écriture des secret. Vient des mots grecs kruptós (couvert, caché) et graphein (écrire, peindre).

Cette science est corrélée à l'évolution des civilisations, organisées en tribus, royaumes, pays, souhaitant protéger des écrits.

Cryptologie $\neq$ Cryptographie $\neq$ Cryptanalyse

- Cryptanalyse est la science pour casser les secrets.
- La cryptographie et la cryptanalyse font partie de la cryptologie.

Première trace connue de la cryptographie

La première trace connue de la cryptographie remonte à l'égypte antique, avec l'utilisation des hiéroglyphes. A cette époque, les scribes, fonctionnaires du monarque, étaient pratiquement les seuls à connaitre leur signification (même au sein de la population, fortement analphabète) et étaient responsables de l?ecritures de documents administratifs notamment.

Comment les hiéroglyphes ont été décryptés au XIX^e siècle ?

Comment les hiéroglyphes ont été décryptés au XIX^e siècle ?

Pierre de Rosette

Grace à la pierre de rosette, découverte en 1799, qui est n décret promulgué en 196 avant Jesus-Christ et rédigé à la fois en égyptien ancien et grec ancien.

Comment les hiéroglyphes ont été décryptés au XIX^e siècle ?

Pierre de Rosette

Grace à la pierre de rosette, découverte en 1799, qui est n décret promulgué en 196 avant Jesus-Christ et rédigé à la fois en égyptien ancien et grec ancien.

Question

La théorie de l'information aurait-elle aidé à étudier les hiéroglyphes aujourd'hui ? Autrement dit, quelle information est contenue dans un mot ?

Comment les hiéroglyphes ont été décryptés au XIX^e siècle ?

Pierre de Rosette

Grace à la pierre de rosette, découverte en 1799, qui est n décret promulgué en 196 avant Jesus-Christ et rédigé à la fois en égyptien ancien et grec ancien.

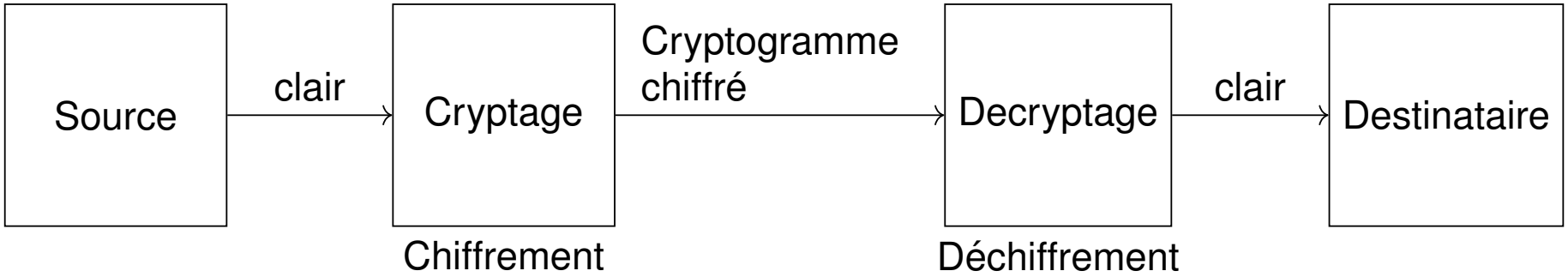
Question

La théorie de l'information aurait-elle aidé à étudier les hiéroglyphes aujourd'hui ? Autrement dit, quelle information est contenue dans un mot ?

Reponse

Le langage est malheureusement difficile à étudier en théorie de l'information, notamment par le fait qu'il fait l'objet d'interprétations. De même, le fait qu'un mot soit défini à partir d'autre mots fait que l'on à une régression à l'infini de leur signification.

Modélisation d'un échange



Vocabulaire

- Le message provenant de la source s'appelle le clair.
- Le message transitant sur le canal non sûr (potentiellement écouté) est appelé le chiffré.
- La fonction exécutée par l'émetteur s'appelle le chiffrement, elle convertie un clair en chiffré.
- La fonction exécutée par le récepteur s'appelle le déchiffrement, elle convertie un chiffré en clair, normalement égal au clair produit par la source.

Propriété de sécurité classiques

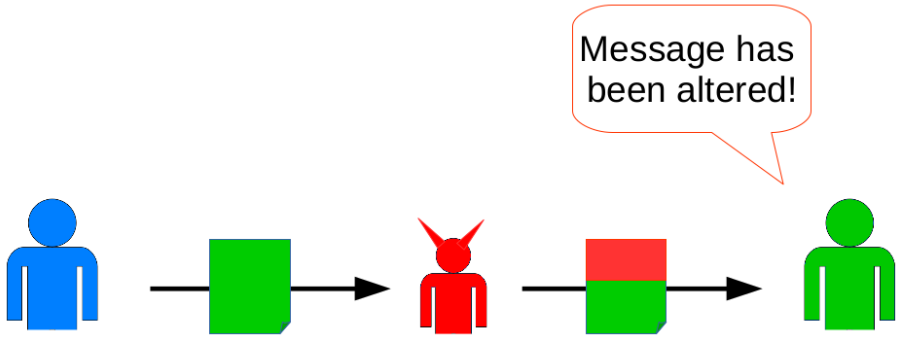
- Confidentialité
- Integrité
- Authentification
- Non-repudiation

Confidentialité

Si un attaquant intercepte un chiffré, il ne peut pas en déduire d'information sur le message.

Propriété de sécurité classiques

- Confidentialité
- Integrité
- Authentification
- Non-repudiation

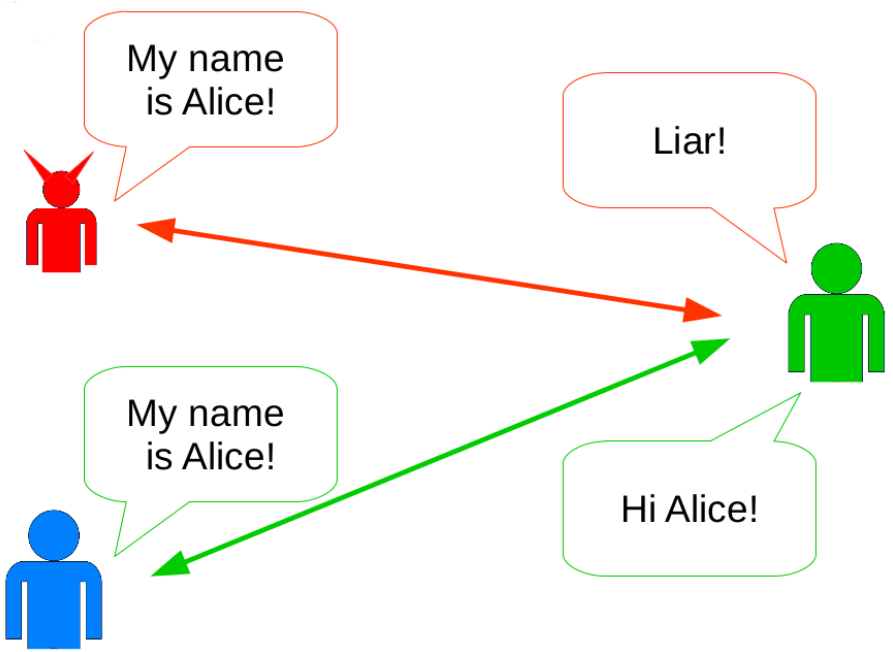


Integrité

Si un attaquant modifie un chiffré, le destinataire peut le détecter.

Propriété de sécurité classiques

- Confidentialité
- Integrité
- Authentification
- Non-repudiation

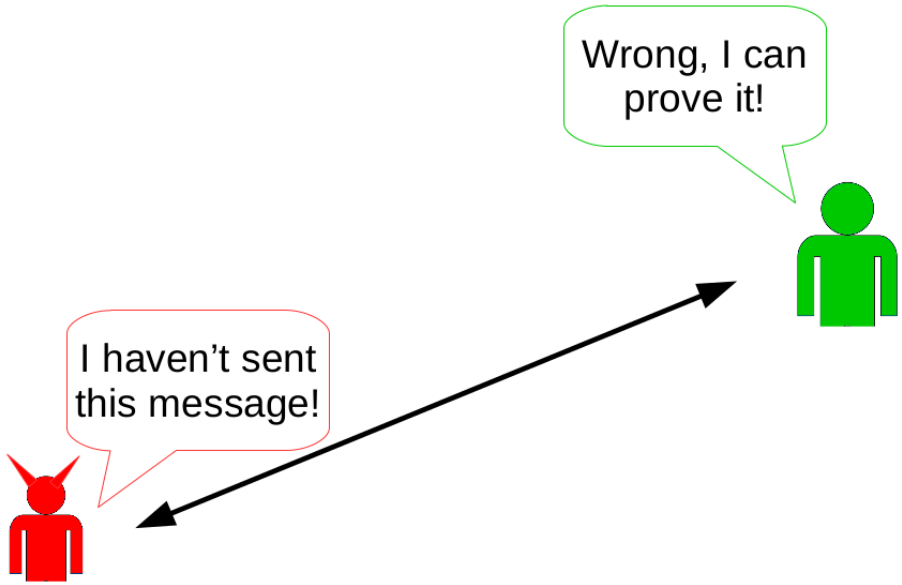


Authentification

Si un attaquant tente d'usurper l'identité de quelqu'un, il peut être détecté.

Propriété de sécurité classiques

- Confidentialité
- Integrité
- Authentification
- Non-repudiation



Non-repudiation

Si quelqu'un réfute être l'émetteur d'un message, on peut prouver le contraire.

Propriété de sécurité classiques

- Confidentialité
- Integrité
- Authentification
- Non-repudiation

Définition du type de canal

En fonction des proriétés de sécurité garanties, on définit plusieurs types de canal :

Canal non sûr	:	Ecoute et modification du message possible
Canal sûr	:	Ecoute et modification du message impossible
Canal confidentiel	:	Ecoute du message impossible
Canal authentique	:	Modification du message impossible

Chiffrements reposant sur la substitution

Rappel

Une substitution est un changement de symboles par d'autres symboles.

Exemple

En prenant $A \to B$, $B \to D$, $C \to A$, $D \to C$

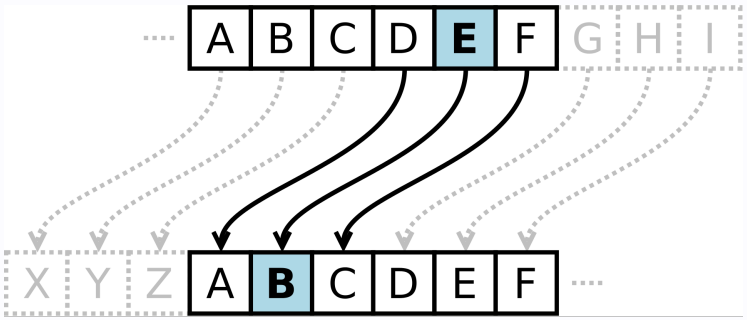
- ABCD
- BDAC

Chiffre de Cesar: -100 AV J-C

Type

Chiffrement par substitution mono-alphabetique.

Principe



On applique une opération de substitution des symboles en se décalant sur l'alphabet d'une valeur donnée. On dit que la substitution est monoalphabétique car chaque lettre est toujours remplacée par la même lettre.

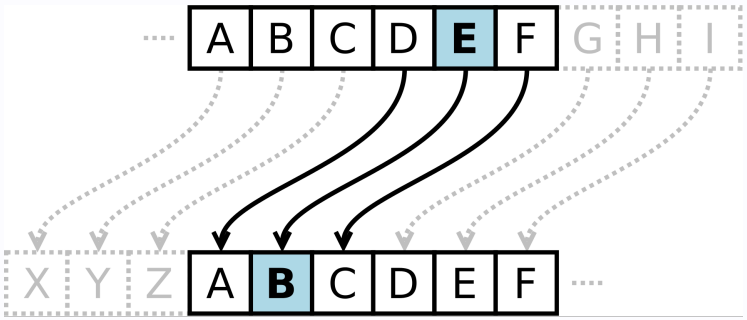
Exemple

On veut chiffrer le mot CANARD, on obtient le mot :

Type

Chiffrement par substitution mono-alphabetique.

Principe



On applique une opération de substitution des symboles en se décalant sur l'alphabet d'une valeur donnée. On dit que la substitution est monoalphabétique car chaque lettre est toujours remplacée par la même lettre.

Exemple

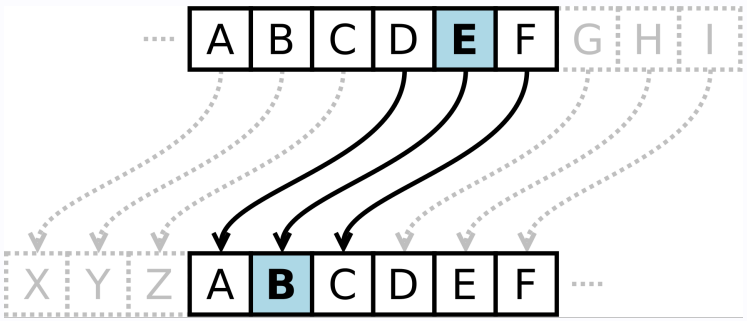
On veut chiffrer le mot CANARD, on obtient le mot : ZXKXOA

Chiffre de Cesar: -100 AV J-C

Type

Chiffrement par substitution mono-alphabetique.

Principe



On applique une opération de substitution des symboles en se décalant sur l'alphabet d'une valeur donnée. On dit que la substitution est monoalphabétique car chaque lettre est toujours remplacée par la même lettre.

Exemple

On veut chiffrer le mot CANARD, on obtient le mot : ZXKXOA

Limitations

L'analyse des fréquences des symboles permet de déterminer sans difficulté la valeur du décalage.

Type

Chiffrement par substitution poly-alphabetique.

Principe

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

L'idée de Vigenère est de faire en sorte que la substitution des lettres ne soient plus unique mais dépende de la valeur d'une clé. Si la clé est plus petite que le message, elle est alors répétée suffisamment de fois pour couvrir tout le message.

Exemple

Toujours avec l'exemple précendent, on souhaite chiffrer le mot CANARD avec la clé KEY.

CANARD
KEYKEY

Chiffre de Vigenère: XVI siècle

Type

Chiffrement par substitution poly-alphabetique.

Principe

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

L'idée de Vigenère est de faire en sorte que la substitution des lettres ne soient plus unique mais dépende de la valeur d'une clé. Si la clé est plus petite que le message, elle est alors répétée suffisamment de fois pour couvrir tout le message.

Exemple

Toujours avec l'exemple précendent, on souhaite chiffrer le mot CANARD avec la clé KEY.

CANARD
KEYKEY MELKVB

Type

Chiffrement par substitution poly-alphabetique.

Principe

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

L'idée de Vigenère est de faire en sorte que la substitution des lettres ne soient plus unique mais dépende de la valeur d'une clé. Si la clé est plus petite que le message, elle est alors répétée suffisamment de fois pour couvrir tout le message.

Limitations

Si la clé est répétée trop de fois (message long), le cryptanalyste peut, après avoir fait une hypothèse sur la taille de la clé, découper le message en regroupant par indice de clé identique, puis mener une attaque fréquentielle.

Type

Chiffrement par substitution polygraghique.

Principe

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

Chiffrement inventé par Charles Wheatstone, puis popularisé par Lord Playfair. La clé de chiffrement est un tableau de 5x5. On chiffre par bigrammes (2 lettres par 2 lettres), en construisant un rectangle en prenant comme extrémités de sa diagonale les deux lettres, et en les substituant par les deux lettres de l'autre diagonale.

Cas particuliers

- Si les 2 lettres identiques se suivent, insérer la lettre X entre les deux ;
- Si les 2 lettres sont sur la même ligne, les remplacer par celles immédiatement à leur droite ;
- Si les 2 lettres sont sur la même colonne, les remplacer par celles immédiatement en dessous de chaque lettre.

Type

Chiffrement par substitution polygraghique.

Principe

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

Chiffrement inventé par Charles Wheatstone, puis popularisé par Lord Playfair. La clé de chiffrement est un tableau de 5x5. On chiffre par bigrammes (2 lettres par 2 lettres), en construisant un rectangle en prenant comme extrémités de sa diagonale les deux lettres, et en les substituant par les deux lettres de l'autre diagonale.

Exemple

CANNE La lettre N étant répétée 2 fois, on chiffre CANXNE, puis après chiffrement :
OH PN AI

Type

Chiffrement par substitution polygraghique.

Principe

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

Chiffrement inventé par Charles Wheatstone, puis popularisé par Lord Playfair. La clé de chiffrement est un tableau de 5x5. On chiffre par bigrammes (2 lettres par 2 lettres), en construisant un rectangle en prenant comme extrémités de sa diagonale les deux lettres, et en les substituant par les deux lettres de l'autre diagonale.

Exemple

CANNE La lettre N étant répétée 2 fois, on chiffre CANXNE, puis après chiffrement :
OH PN AI

Type

Chiffrement par substitution polygraghique.

Principe

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

Chiffrement inventé par Charles Wheatstone, puis popularisé par Lord Playfair. La clé de chiffrement est un tableau de 5x5. On chiffre par bigrammes (2 lettres par 2 lettres), en construisant un rectangle en prenant comme extrémités de sa diagonale les deux lettres, et en les substituant par les deux lettres de l'autre diagonale.

Limitations

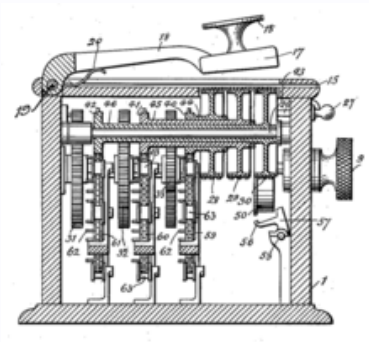
Attaque par analyse des fréquences de bigrammes. De plus, si vous connaissez un clair et un chiffré, vous obtenez beaucoup d'information sur la clé.

Chiffre de Hill: XX siècle (1929)

Type

Chiffrement par substitution polygraphique. Fait partie également d'une des transformations appliquées lors du chiffrement standard AES.

Principe



Dans le cas où l'on chiffre par paquets de deux caractères, l'algorithme de chiffrement est le suivant:

$$\begin{pmatrix} C_k \\ C_{k+1} \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \cdot \begin{pmatrix} P_k \\ P_{k+1} \end{pmatrix} \text{ [26]}$$

Avec P_k le k^{eme} caractère du message, C_k le k^{eme} caractère du chiffré, et (a, b, c, d) la clé de chiffrement associée.

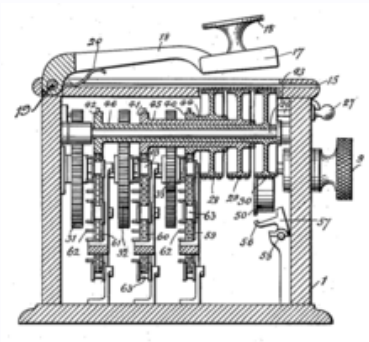
Propriétés

- Pour une matrice 2x2, sur l'alphabet (26 caractères), le nombre de transformations est donnée par le nombre de matrices inversibles modulo 26 ;
- Attaque sur les bigrammes possible ;
- Connaissant le clair, on peut récupérer facilement une partie de la clé.

Type

Chiffrement par substitution polygraphique. Fait partie également d'une des transformations appliquées lors du chiffrement standard AES.

Principe



Dans le cas où l'on chiffre par paquets de deux caractères, l'algorithme de chiffrement est le suivant:

$$\begin{pmatrix} C_k \\ C_{k+1} \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \cdot \begin{pmatrix} P_k \\ P_{k+1} \end{pmatrix} \text{ [26]}$$

Avec P_k le k^{eme} caractère du message, C_k le k^{eme} caractère du chiffré, et (a, b, c, d) la clé de chiffrement associée.

Nombre de clés possibles

- Pour déchiffrer correctement, il faut que la matrice soit inversible.
- Autrement dit, il faut que sont déterminant (ad -bc) ait un inverse dans $\mathbb{Z}_{26}$, autrement dit que ad - bc soit premier avec 26 ;
- On prouve que le nombre de solutions est $157248 \approx 2^{17.26}$.

Hypothèses et notation

- La fonction de chiffrement est bijective et déterministe.
- On note K l'ensemble des clés, S_K la source émettant des clés, et $k_i \in K$ les messages émis pas la source S_K. On applique la même notation pour les clairs et les chiffrés.
- On note $n_k = \text{Card}(K)$
- On note $n_c = \text{Card}(C) = \text{Card}(M) = n_m$

Par définition :

$$H(K, C) = \sum_k \sum_c p(k, c) \log_2 \frac{1}{p(k, c)}$$

$$H(K, M) = \sum_k \sum_m p(k, m) \log_2 \frac{1}{p(k, m)}$$

Comme la fonction de chiffrement est bijective, alors en fixant une clé particulière :

$$\forall i \in \mathbb{Z}_{n_m} \exists j \in \mathbb{Z}_{n_c} p(m_i) = p(c_j)$$

Ceci étant vrai pour toutes les clés possibles, nous avons que :

$$\forall k \in K \forall i \in \mathbb{Z}_{n_m} \exists j \in \mathbb{Z}_{n_c} p(k, m_i) = p(k, c_j)$$

Hypothèses et notation

- La fonction de chiffrement est bijective et déterministe.
- On note K l'ensemble des clés, S_K la source émettant des clés, et $k_i \in K$ les messages émis pas la source S_K. On applique la même notation pour les clairs et les chiffrés.
- On note $n_k = \mathrm{Card}(K)$
- On note $n_c = \mathrm{Card}(C) = \mathrm{Card}(M) = n_m$

Par définition :

$$H(K, C) = \sum_k \sum_c p(k, c) \log_2 \frac{1}{p(k, c)}$$

$$H(K, M) = \sum_k \sum_m p(k, m) \log_2 \frac{1}{p(k, m)}$$

Comme la fonction de chiffrement est bijective, alors en fixant une clé particulière :

$$\forall i \in \mathbb{Z}_{n_m} \exists j \in \mathbb{Z}_{n_c} p(m_i) = p(c_j)$$

Ceci étant vrai pour toutes les clés possibles, nous avons que :

$$\forall k \in K \forall i \in \mathbb{Z}_{n_m} \exists j \in \mathbb{Z}_{n_c} p(k, m_i) = p(k, c_j)$$

Hypothèses et notation

- La fonction de chiffrement est bijective et déterministe.
- On note K l'ensemble des clés, S_K la source émettant des clés, et $k_i \in K$ les messages émis pas la source S_K. On applique la même notation pour les clairs et les chiffrés.
- On note $n_k = \mathrm{Card}(K)$
- On note $n_c = \mathrm{Card}(C) = \mathrm{Card}(M) = n_m$

Par définition :

$$H(K, C) = \sum_k \sum_c p(k, c) \log_2 \frac{1}{p(k, c)}$$

$$H(K, M) = \sum_k \sum_m p(k, m) \log_2 \frac{1}{p(k, m)}$$

Comme la fonction de chiffrement est bijective, alors en fixant une clé particulière :

$$\forall i \in \mathbb{Z}_{n_m} \exists j \in \mathbb{Z}_{n_c} p(m_i) = p(c_j)$$

Ceci étant vrai pour toutes les clés possibles, nous avons que :

$$\forall k \in K \forall i \in \mathbb{Z}_{n_m} \exists j \in \mathbb{Z}_{n_c} p(k, m_i) = p(k, c_j)$$

Hypothèses et notation

- La fonction de chiffrement est bijective et déterministe.
- On note K l'ensemble des clés, S_K la source émettant des clés, et $k_i \in K$ les messages émis pas la source S_K. On applique la même notation pour les clairs et les chiffrés.
- On note $n_k = \text{Card}(K)$
- On note $n_c = \text{Card}(C) = \text{Card}(M) = n_m$

Par définition :

$$H(K, C) = \sum_k \sum_c p(k, c) \log_2 \frac{1}{p(k, c)}$$

$$H(K, M) = \sum_k \sum_m p(k, m) \log_2 \frac{1}{p(k, m)}$$

Comme la fonction de chiffrement est bijective, alors en fixant une clé particulière :

$$\forall i \in \mathbb{Z}_{n_m} \exists j \in \mathbb{Z}_{n_c} p(m_i) = p(c_j)$$

Ceci étant vrai pour toutes les clés possibles, nous avons que :

$$\forall k \in K \forall i \in \mathbb{Z}_{n_m} \exists j \in \mathbb{Z}_{n_c} p(k, m_i) = p(k, c_j)$$

Hypothèses et notation

- La fonction de chiffrement est bijective et déterministe.
- On note K l'ensemble des clés, S_K la source émettant des clés, et $k_i \in K$ les messages émis pas la source S_K. On applique la même notation pour les clairs et les chiffrés.
- On note $n_k = \mathrm{Card}(K)$
- On note $n_c = \mathrm{Card}(C) = \mathrm{Card}(M) = n_m$

En opérant un simple changement de variable, on obtient bien que :

$$\sum_k \sum_c p(k,c) \log_2 \frac{1}{p(k,c)} = \sum_k \sum_m p(k,m) \log_2 \frac{1}{p(k,m)}$$

Définition

Un chiffrement est dit parfaitement sûr (ou a secret parfait) si le chiffré ne donne aucune information sur le message. De manière équivalente, on a la propriété suivante :

$$p(c|m) = p(c) \tag{1}$$

Définition

Un chiffrement est dit parfaitement sûr (ou a secret parfait) si le chiffré ne donne aucune information sur le message. De manière équivalente, on a la propriété suivante :

$$p(c|m) = p(c) \tag{1}$$

Théorème

Un chiffrement est parfaitement sûr si et seulement si tous les chiffrés sont indépendants sans connaissance de la clé.

Définition

Un chiffrement est dit parfaitement sûr (ou a secret parfait) si le chiffré ne donne aucune information sur le message. De manière équivalente, on a la propriété suivante :

$$p(c|m) = p(c) \tag{1}$$

Théorème

Un chiffrement est parfaitement sûr si et seulement si tous les chiffrés sont indépendants sans connaissance de la clé.

Propriété

Si on considère un chiffrement parfaitement sûr, alors nécessairement :

$$Card(K) \geq Card(M) \tag{2}$$

Avec des clés choisies uniformément et aléatoirement.

Construction d'un chiffrement parfaitement sur

clair $\oplus$ clé = chiffré chiffré $\oplus$ clé = clair

```
message :  0 1 0 1 1 1 1 0 0 0 0 1 0 0 1
clé     :  1 1 0 0 0 1 0 1 0 0 0 1 1 1 0
=======================================
chiffré :  1 0 0 1 1 0 1 1 0 0 0 1 1 1 1
```

Propriété de sécurité 1

Pour une clé tirée de manière uniforme, la sortie suit une distribution uniforme.

Preuve

$$\begin{aligned} P(c_i = 1) &= P(k_i = 1 \cap m_i = 0) + P(k_i = 0 \cap m_i = 1) \\ &= P(k_i = 1) \times P(m_i = 0) + P(k_i = 0) \times P(m_i = 1) \\ &= \frac{1}{2} \times P(m_i = 0) + \frac{1}{2} \times P(m_i = 1) \\ &= \frac{1}{2} \times (P(m_i = 0) + P(m_i = 1)) \\ P(c_i = 1) &= \frac{1}{2} \end{aligned}$$

clair $\oplus$ clé = chiffré chiffré $\oplus$ clé = clair

```
message :  0 1 0 1 1 1 1 0 0 0 0 1 0 0 1
clé     :  1 1 0 0 0 1 0 1 0 0 0 1 1 1 0
=======================================
chiffré :  1 0 0 1 1 0 1 1 0 0 0 1 1 1 1
```

Propriété de sécurité 1

Pour une clé tirée de manière uniforme, la sortie suit une distribution uniforme.

Preuve

$$\begin{aligned} P(c_i = 1) &= P(k_i = 1 \cap m_i = 0) + P(k_i = 0 \cap m_i = 1) \\ &= P(k_i = 1) \times P(m_i = 0) + P(k_i = 0) \times P(m_i = 1) \\ &= \frac{1}{2} \times P(m_i = 0) + \frac{1}{2} \times P(m_i = 1) \\ &= \frac{1}{2} \times (P(m_i = 0) + P(m_i = 1)) \\ P(c_i = 1) &= \frac{1}{2} \end{aligned}$$

clair $\oplus$ clé = chiffré chiffré $\oplus$ clé = clair

```
message :  0 1 0 1 1 1 1 0 0 0 0 1 0 0 1
clé     :  1 1 0 0 0 1 0 1 0 0 0 1 1 1 0
=======================================
chiffré :  1 0 0 1 1 0 1 1 0 0 0 1 1 1 1
```

Propriété de sécurité 1

Pour une clé tirée de manière uniforme, la sortie suit une distribution uniforme.

Preuve

$$\begin{aligned} P(c_i = 1) &= P(k_i = 1 \cap m_i = 0) + P(k_i = 0 \cap m_i = 1) \\ &= P(k_i = 1) \times P(m_i = 0) + P(k_i = 0) \times P(m_i = 1) \\ &= \frac{1}{2} \times P(m_i = 0) + \frac{1}{2} \times P(m_i = 1) \\ &= \frac{1}{2} \times (P(m_i = 0) + P(m_i = 1)) \\ P(c_i = 1) &= \frac{1}{2} \end{aligned}$$

clair $\oplus$ clé = chiffré chiffré $\oplus$ clé = clair

```
message :  0 1 0 1 1 1 1 0 0 0 0 1 0 0 1
clé     :  1 1 0 0 0 1 0 1 0 0 0 1 1 1 0
=======================================
chiffré :  1 0 0 1 1 0 1 1 0 0 0 1 1 1 1
```

Propriété de sécurité 1

Pour une clé tirée de manière uniforme, la sortie suit une distribution uniforme.

Preuve

$$\begin{aligned} P(c_i = 1) &= P(k_i = 1 \cap m_i = 0) + P(k_i = 0 \cap m_i = 1) \\ &= P(k_i = 1) \times P(m_i = 0) + P(k_i = 0) \times P(m_i = 1) \\ &= \frac{1}{2} \times P(m_i = 0) + \frac{1}{2} \times P(m_i = 1) \\ &= \frac{1}{2} \times (P(m_i = 0) + P(m_i = 1)) \\ P(c_i = 1) &= \frac{1}{2} \end{aligned}$$

clair $\oplus$ clé = chiffré chiffré $\oplus$ clé = clair

```
message :  0 1 0 1 1 1 1 0 0 0 0 1 0 0 1
clé     :  1 1 0 0 0 1 0 1 0 0 0 1 1 1 0
=======================================
chiffré :  1 0 0 1 1 0 1 1 0 0 0 1 1 1 1
```

Propriété de sécurité 1

Pour une clé tirée de manière uniforme, la sortie suit une distribution uniforme.

Preuve

$$\begin{aligned} P(c_i = 1) &= P(k_i = 1 \cap m_i = 0) + P(k_i = 0 \cap m_i = 1) \\ &= P(k_i = 1) \times P(m_i = 0) + P(k_i = 0) \times P(m_i = 1) \\ &= \frac{1}{2} \times P(m_i = 0) + \frac{1}{2} \times P(m_i = 1) \\ &= \frac{1}{2} \times (P(m_i = 0) + P(m_i = 1)) \end{aligned}$$

$$P(c_i = 1) = \frac{1}{2}$$

clair $\oplus$ clé = chiffré chiffré $\oplus$ clé = clair

```
message :  0 1 0 1 1 1 1 0 0 0 0 1 0 0 1
clé     :  1 1 0 0 0 1 0 1 0 0 0 1 1 1 0
=======================================
chiffré :  1 0 0 1 1 0 1 1 0 0 0 1 1 1 1
```

Propriété de sécurité 1

Pour une clé tirée de manière uniforme, la sortie suit une distribution uniforme.

Preuve

$$
\begin{aligned}
P(c_i = 1) &= P(k_i = 1 \cap m_i = 0) + P(k_i = 0 \cap m_i = 1) \\
&= P(k_i = 1) \times P(m_i = 0) + P(k_i = 0) \times P(m_i = 1) \\
&= \frac{1}{2} \times P(m_i = 0) + \frac{1}{2} \times P(m_i = 1) \\
&= \frac{1}{2} \times (P(m_i = 0) + P(m_i = 1)) \\
P(c_i = 1) &= \frac{1}{2}
\end{aligned}
$$

Masque jetable (One Time Pad) / Vernam (1917)

clair $\oplus$ clé = chiffré chiffré $\oplus$ clé = clair

```
message :  0 1 0 1 1 1 1 0 0 0 0 1 0 0 1
clé     :  1 1 0 0 0 1 0 1 0 0 0 1 1 1 0
=======================================
chiffré :  1 0 0 1 1 0 1 1 0 0 0 1 1 1 1
```

Propriété de sécurité 2

Pour un chiffré donné, tout clair est possible.

clair $\oplus$ clé = chiffré chiffré $\oplus$ clé = clair

```
message :  0 1
clé     :  1 1
=======================================
chiffré :  1 0
```

Propriété de sécurité 2

Pour un chiffré donné, tout clair est possible.

clair $\oplus$ clé = chiffré chiffré $\oplus$ clé = clair

```
message :  X X
clé     :  X X
=======================================
chiffré :  1 0
```

Propriété de sécurité 2

Pour un chiffré donné, tout clair est possible.

Masque jetable (One Time Pad) / Vernam (1917)

clair $\oplus$ clé = chiffré			chiffré $\oplus$ clé = clair			
message :	X	X	0 0 |	0 1 |	1 0 |	1 1
clé :	X	X	1 0 |	1 1 |	0 0 |	0 1
chiffré :	1	0	1 0 |	1 0 |	1 0 |	1 0

Propriété de sécurité 2

Pour un chiffré donné, tout clair est possible.

Masque jetable (One Time Pad) / Vernam (1917)

INSA TOULOUSE

Question

Le chiffre de Vernam est-il parfaitement sûr ?

Masque jetable (One Time Pad) / Vernam (1917)

INSA TOULOUSE

Question

Le chiffre de Vernam est-il parfaitement sûr ?

Reponse

La sortie étant uniforme pour une clé choisie de manière purement aléatoire, le Chiffre de Vernam est parfaitement sûr.

Maléabilité

Si un attaquant intercepte un chiffré c et calcule $c' = c \oplus x$, alors le clair m est aussi modifié ($m' = m \oplus x$).

Gestion de la clé

Une clé ne peut pas être réutilisée :
$c_1 \oplus c_2 = (m_1 \oplus k) \oplus (m_2 \oplus k) = m_1 \oplus m_2$

Sans oublier que pour qu'il soit parfaitement sûr, il faut avoir une clé (générée avec un aléa pur) aussi grande que le clair.

Chiffrements reposant sur la permutation

Rappel

Une permutation est un changement de l'ordre des symboles.

Exemple

- ABCDEF
- BDFACE

Principe

On écrit un texte en suivant le gabarie suivant :

```
      1               5               9
  2       3   6       7   10              11   etc..
      4               8               12
```

Puis on encode le texte en émettant les symboles ligne par ligne.

Permutation à croix grecque

Principe

On écrit un texte en suivant le gabarie suivant :

```
      1           5           9
  2       3   6       7   10      11   etc..
      4           8          12
```

Puis on encode le texte en émettant les symboles ligne par ligne.

Exemple

On souhaite chiffrer le message suivant : MESSAGE TOP SECRET :

```
      M           A           O           C
  E       S   G       E   P       S   R       E
      S           T           E           T
```

Permutation à croix grecque

Principe

On écrit un texte en suivant le gabarie suivant :

```
      1           5           9
  2       3   6       7   10      11   etc..
      4           8          12
```

Puis on encode le texte en émettant les symboles ligne par ligne.

Exemple

On souhaite chiffrer le message suivant : MESSAGE TOP SECRET :

```
      M           A           O           C
  E       S   G       E   P       S   R       E
      S           T           E           T
```

Message chiffré : MAOCESGEPSRESTET

Principe

On écrit un texte en suivant le gabarie suivant :

```
      1           5           9
  2       3   6       7   10      11   etc..
      4           8          12
```

Puis on encode le texte en émettant les symboles ligne par ligne.

Exemple

On souhaite chiffrer le message suivant : MESSAGE TOP SECRET :

```
      M           A           O           C
  E       S   G       E   P       S   R       E
      S           T           E           T
```

Message chiffré : MAOCESGEPSRESTET

Limitations

- Pas de clé, le secret est contenu dans la méthode chiffrement $\rightarrow$ Très mauvaise pratique (l'encodage est trouvé par l'ennemi, tous les chiffrés précédents tombent).
- Pas de modification de la fréquence des caractères.

Principe

Le principe est de paramétrer la permutation à partir d'une clé que l'on appelle Mot Clé. Nous allons tout d'abord remplir un tableau ligne par ligne avec notre message, chaque ligne faisant la taille du Mot Clé, puis on va envoyer le message en émettant chaque colonne dans l'ordre alphabétique du Mot Clé. Si des lettres se répètent dans le Mot Clé, on prend les colonnes de gauche à droite.

Exemple

On souhaite chiffrer LA VIE EST BELLE LES OISEAUX CHANTENT avec le Mot Clé TOULOUSE :

T	O	U	L	O	U	S	E
L	A	V	I	E	E	S	T
B	E	L	L	E	L	E	S
O	I	S	E	A	U	X	C
H	A	N	T	E	N	T	X

Principe

Le principe est de paramétrer la permutation à partir d'une clé que l'on appelle Mot Clé. Nous allons tout d'abord remplir un tableau ligne par ligne avec notre message, chaque ligne faisant la taille du Mot Clé, puis on va envoyer le message en émettant chaque colonne dans l'ordre alphabétique du Mot Clé. Si des lettres se répètent dans le Mot Clé, on prend les colonnes de gauche à droite.

Exemple

On souhaite chiffrer LA VIE EST BELLE LES OISEAUX CHANTENT avec le Mot Clé TOULOUSE :

T	O	U	L	O	U	S	E
L	A	V	I	E	E	S	T
B	E	L	L	E	L	E	S
O	I	S	E	A	U	X	C
H	A	N	T	E	N	T	X

Permutation des colonnes en rangeant par ordre alphabétique le Mot Clé :

E	L	O	O	S	T	U	U
T	I	A	E	S	L	V	E
S	L	E	E	E	B	L	L
C	E	I	A	X	O	S	U
X	R	A	E	T	H	N	N

Le chiffré est donc : TSCX ILER AEIA EEAE SEXT LBOH VLSN ELUN.

Cryptographie moderne

Auguste Kerckhoffs: XIX siècle

Kerckhoffs, cryptographe militaire, a proposé différents principes qui restent aujourd'hui la référence de la cryptographie moderne :

- Le cryptosystème doit être, s'il n'est pas théoriquement inviolable, inviolable en pratique ;
- La conception d'un cryptosystème ne doit pas contenir de secret, et compromettre le cryptosystème de doit pas amener l'effondrement de la sécurité.

Aujourd'hui, quasiment l'intégralité des cryptosystèmes sont publiques. Certains sont sous brevet mais cela reste marginal et non bloquant grâce à la présence d'alternatives. Le niveau de sécurité d'un cryptosystème est évalué en fonction des performances des machines informatiques.

Fritz Nebel

Georges Painvin

Inventé par le le lieutenant Allemand Fritz Nebel et fut utilisé pendant la première guerre mondiale pour sécuriser les communications radiophoniques des allemands lors de l'offensive sur Paris. Il fu néanmoins cassé début 1918 par Georges Painvin.

Phase 1 : Substitution

Cette première partie reprend le principe du carré de polype (150 AV JC), en remplaçant chaque symboles par un bigramme à partir d'une table :

	A	D	F	G	V	X
A	8	T	B	W	R	Q
D	P	4	C	G	2	9
F	3	O	5	M	X	E
G	D	A	Z	J	S	Y
V	L	H	7	U	V	0
X	N	1	K	6	I	F

Pourquoi ADFGVX ? Car leurs code morse diffèrent fortement.

Fritz Nebel

Georges Painvin

Inventé par le le lieutenant Allemand Fritz Nebel et fut utilisé pendant la première guerre mondiale pour sécuriser les communications radiophoniques des allemands lors de l'offensive sur Paris. Il fu néanmoins cassé début 1918 par Georges Painvin.

Phase 2 : Permutation

Cette deuxième partie reprend le même principe que la permutation à mot complet.

Fritz Nebel

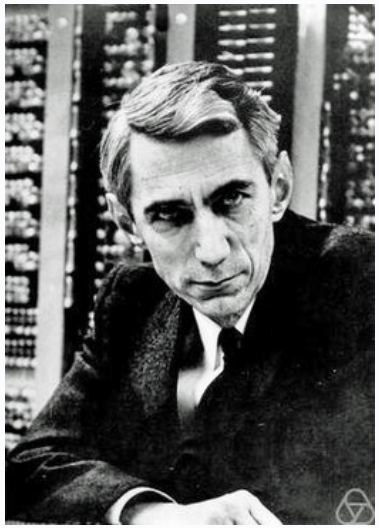
Georges Painvin

Inventé par le le lieutenant Allemand Fritz Nebel et fut utilisé pendant la première guerre mondiale pour sécuriser les communications radiophoniques des allemands lors de l'offensive sur Paris. Il fu néanmoins cassé début 1918 par Georges Painvin.

Cryptanalyse

Sa cryptanalyse est loin d'être évidente à cause de la phase de permutation qui casse les fréquences des bigrammes. Cependant, compte tenu du nombre limité de bigrammes, pour des messages longs, des répétitions apparaissent permettant à partir de la connaissance du langage source, de récupérer la correspondance des lettres fréquentes (e, a, s, n).

Claude Shannon: XX siècle



Claude Shannon est un ingénieur en génie électrique et cryptographie. Expert en télécommunications, il a travaillé pendant la seconde guerre mondiale à extraire dans les codes ennemis des informations pertinentes cachés dans les brouillages (travaux classifiés jusqu'en 1980). En 1948, une fois la Guerre terminée, il publie dans un article dans le journal *Mathematical Theory of Communication* qui pose les outils mathématiques pour modéliser la transmission de l'information.

La théorie de l'information et des codes fait partie intégrante de la cryptographie d'aujourd'hui. Elle donne des outils de compréhension des limites à la sécurité, et des outils pour la cryptanalyse.

Alan Turing: XX siècle

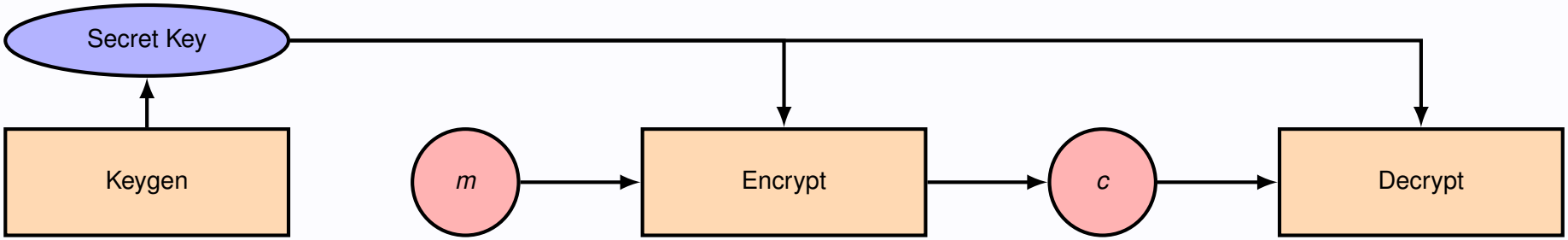
Contemporain de Claude Shannon, c'est un mathématicien et cryptographe Britannique ayant fortement contribué à la théorie des fonctions calculables. Ces travaux ont permis pendant la seconde Guerre Mondiale à construire une machine permettant de décryptage de cryptogrammes produits à partir de la machine Enigma.

- Les primitives cryptographiques modernes reposent sur des problèmes mathématiques réputés difficiles en terme de complexité calculatoire ;
- Son utilisation est très standardisée et très majoritairement publique ;
- La conception de bibliothèques logicielles pour la cryptographie demande une grand expertise. Il existe de nombreuses bibliothèques réputées et maintenues (openSSL, libreSSL, mbedTLS, ...).

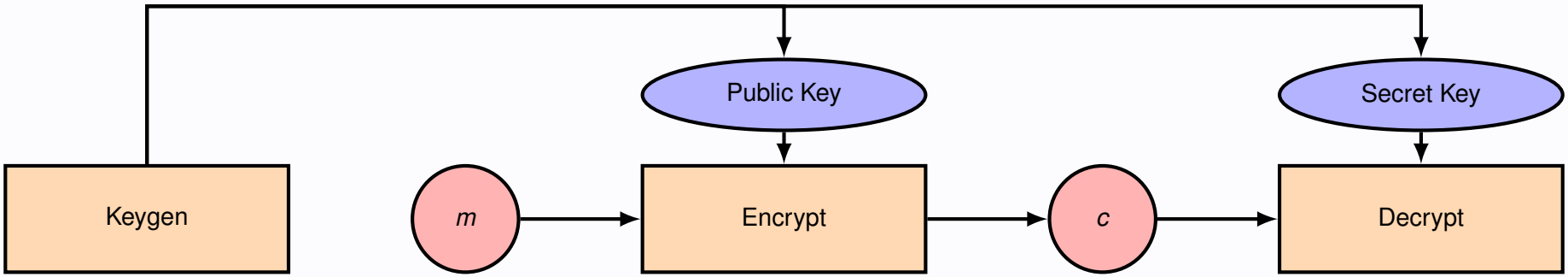
Elle reste néanmoins vulnérable à de nouvelles menaces

- Attaques physiques de type injection de faute :
  - glitches d'alimentations, impulsion laser, attaques sur la mémoire, ...
- Canaux de fuite :
  - Attaque temporelle sur les caches, écoute de la consommation énergétique, ...
- L'ordinateur quantique :
  - Casse les cryptosystèmes reposant sur le logarithme discret (Shor) et le problème de factorisation (variante), soit toute la cryptographie dite asymétrique actuelle ;
  - Divise par deux le niveau de sécurité des attaques reposant sur la recherche exhaustive, donc divise par 2 le niveau de sécurité de la cryptographie dite symétrique.

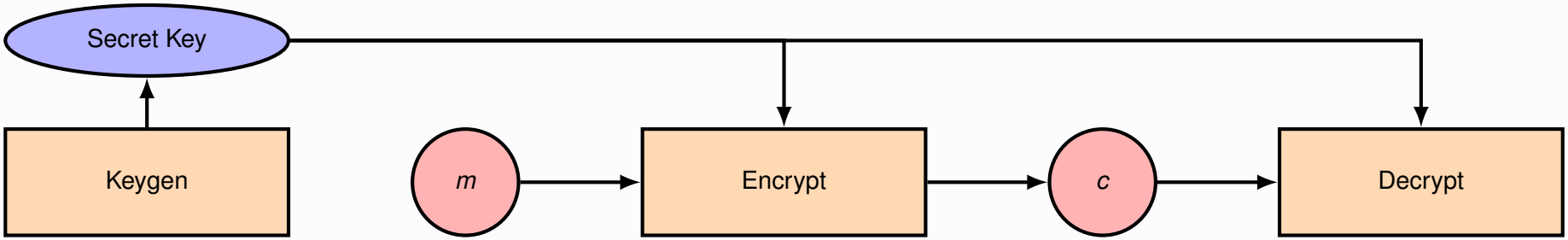
Chiffrement symétrique



Chiffrement asymétrique



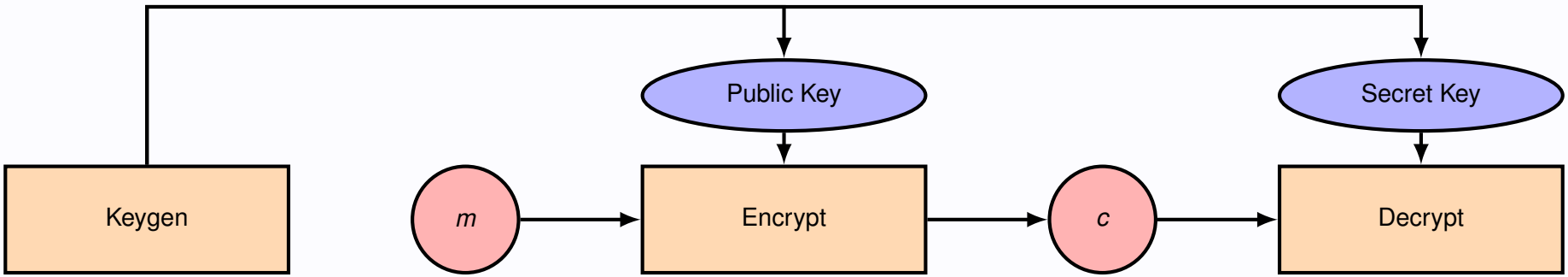
Chiffrement symétrique



Usage du chiffrement symétrique

- Chiffrement symétrique est utilisé classiquement pour l'échange de données.
- Le chiffrement et déchiffrement sont en général rapides.
- La taille du chiffré est proche de la taille du clair.

Chiffrement asymétrique



Usage du chiffrement asymétrique

- Le chiffrement asymétrique est utilisé classiquement pour l'échange de clés.
- C'est un chiffrement de type tous-vers-un, c'est-à-dire plein de sources, un seul destinataire.

Cryptographie symétrique moderne

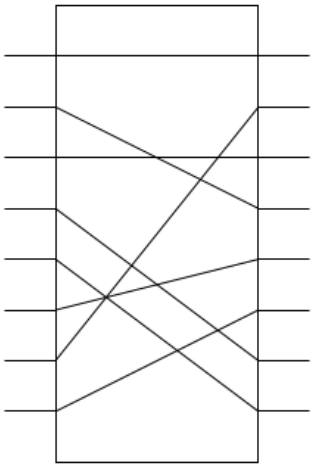


Figure: P-BOX

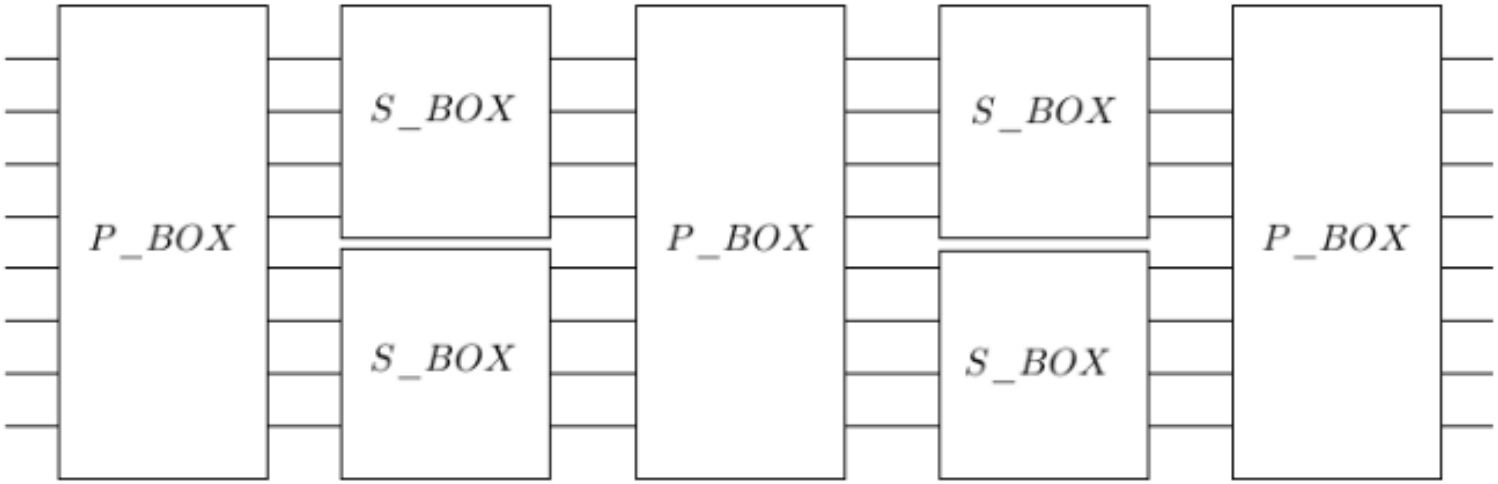


Figure: Machine de Lucifer

Principe

Circuit électronique constitué d'une succession d'opérations de substitution (S-BOX) et de permutation (P-BOX). Approche très générique que l'on va retrouver (adapté) sur les algorithmes utilisés dans les standards (typiquement AES).

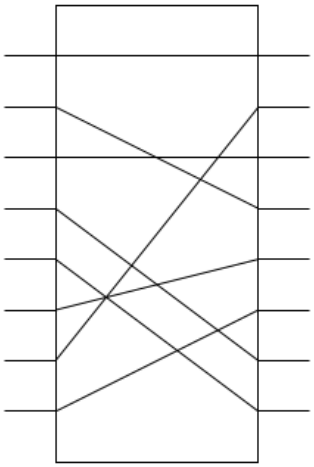


Figure: P-BOX

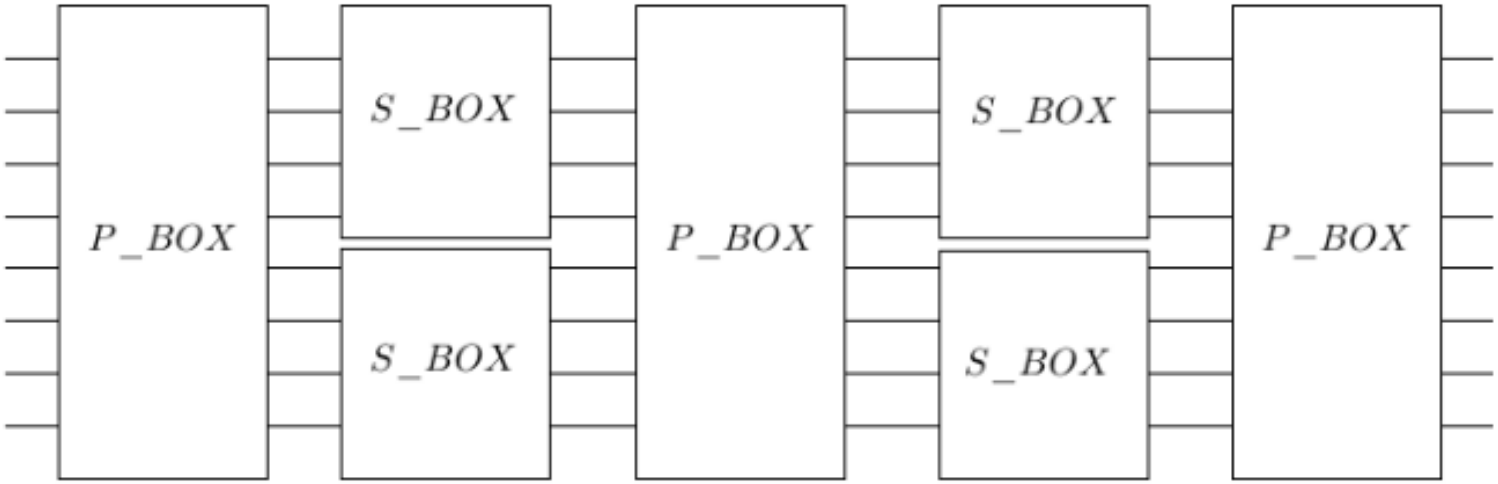


Figure: Machine de Lucifer

Construction d'une P-BOX

Connexion des entrées/sorties au niveau bit. La complexité de fabrication augmente fortement avec le nombre n de broches. Facile à rétro-ingénierie en n essais, en activant une broche à la fois.

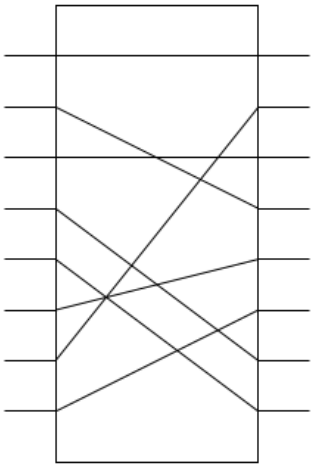


Figure: P-BOX

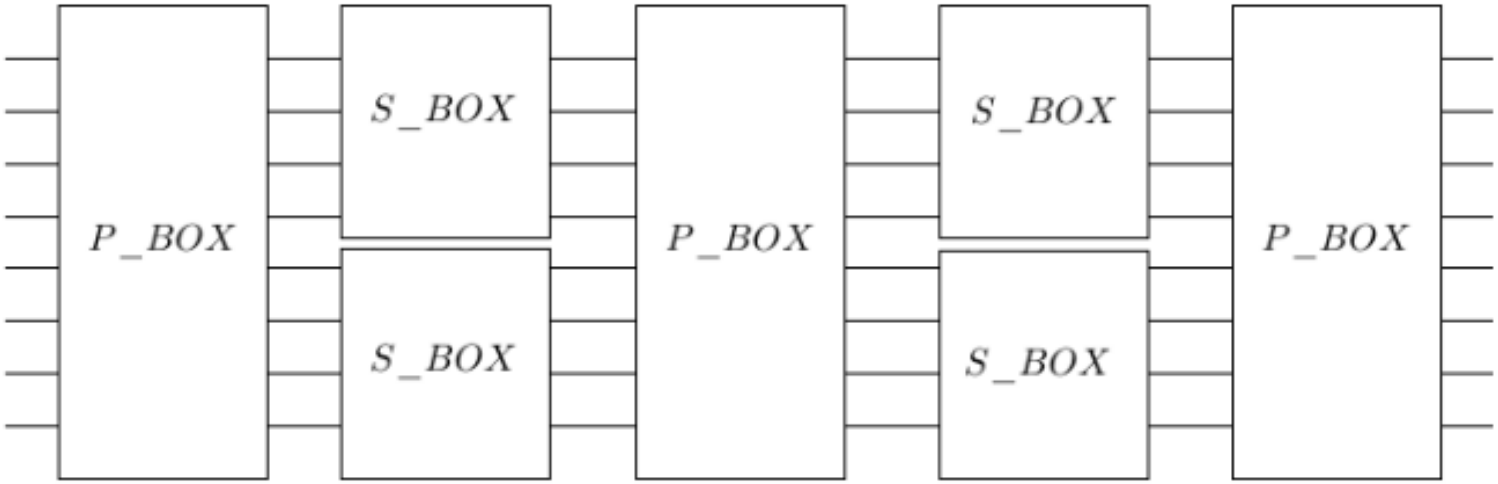


Figure: Machine de Lucifer

Construction d'une S-BOX

Stockage des correspondances dans une mémoire ROM (ou RAM), en considérant l'adresse comme le symbole d'entrée, et la valeur stockée à cette adresse le symbole correspondant.

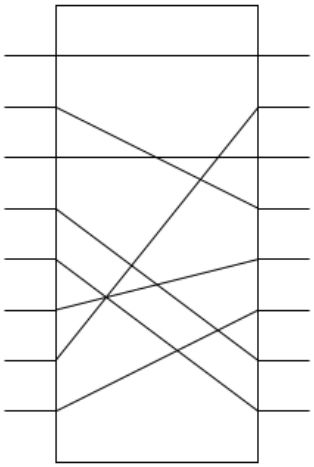


Figure: P-BOX

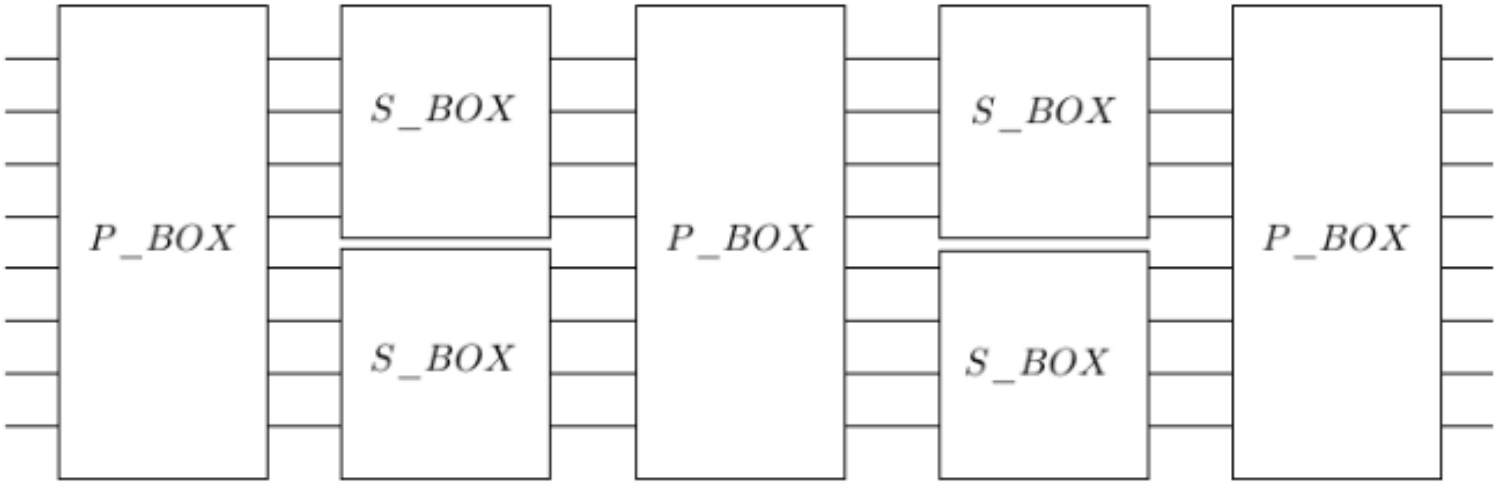


Figure: Machine de Lucifer

Sécurité

Pas de clé implique que si une machine tombe dans les mains de l'ennemi, compromission de tous les messages émis avec une machine similaire.

La propriété de confidentialité d'un chiffrement repose sur deux grands principes établis par Shannon dans *Communication Theory of Secrecy Systems* en 1949.

La confusion

La confusion correspond historiquement à une volonté de rendre la relation entre la clé de chiffrement et le texte chiffré la plus complexe possible. Aujourd'hui, la confusion est liée à la non linéarité de la transformation.

La diffusion

La diffusion correspond historiquement à la propriété où la redondance statistique dans un texte en clair est dissipée dans les statistiques du texte chiffré. On parle aujourd'hui plutôt de *l'effet d'avalanche* : Dans un chiffrement avec une bonne diffusion, l'inversion d'un seul bit en entrée doit changer chaque bit en sortie avec une probabilité de 0,5 (critère d'avalanche strict).

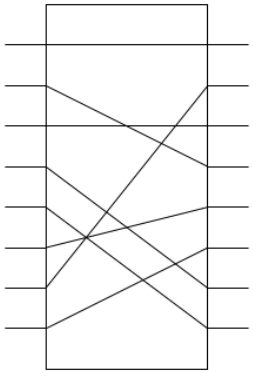


Figure: P-BOX

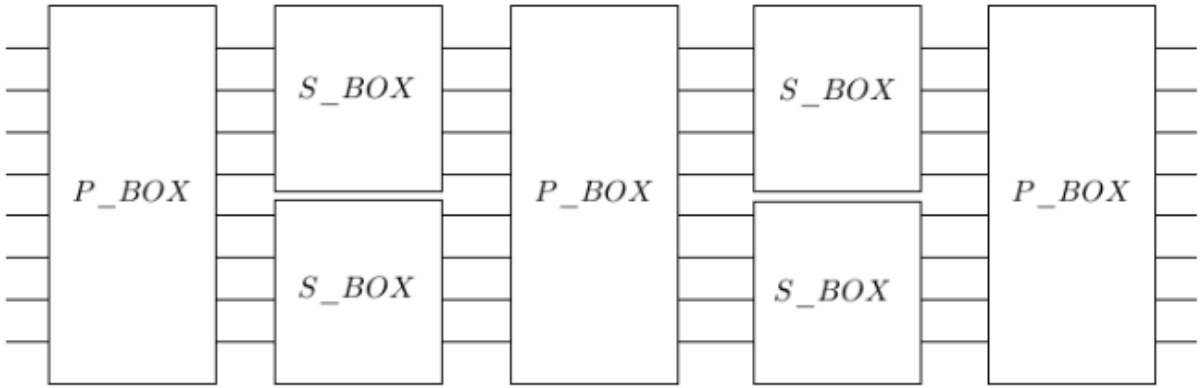


Figure: Machine de Lucifer

Confusion

Renforcement de la non linéarité de la transformation.

La diffusion

Inversion d'un bit en entrée provoque une inversion de chaque bit en sortie avec une probabilité de 0,5.

Question

La fonction S-BOX est responsable principalement de :

A. La confusion
B. La diffusion

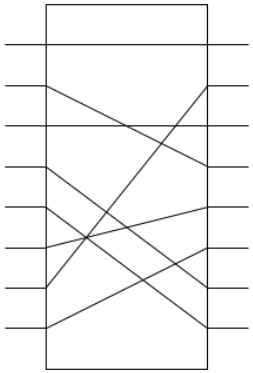


Figure: P-BOX

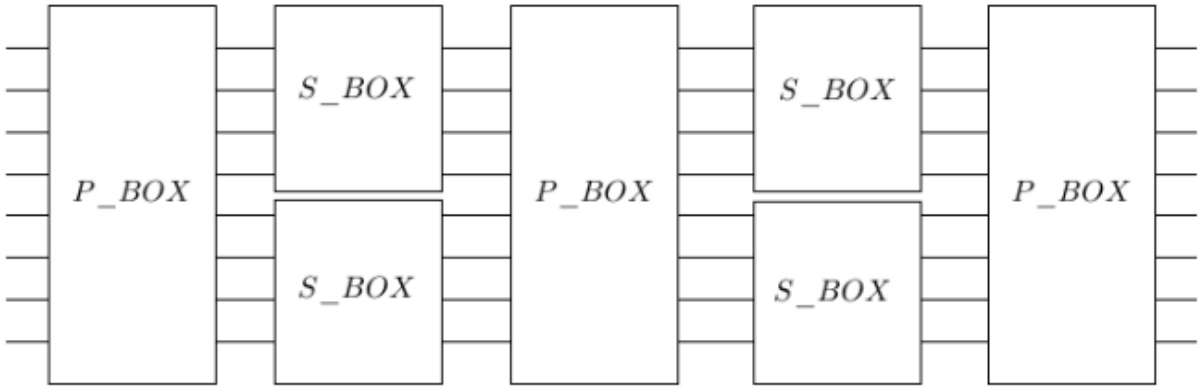


Figure: Machine de Lucifer

Confusion

Renforcement de la non linéarité de la transformation.

La diffusion

Inversion d'un bit en entrée provoque une inversion de chaque bit en sortie avec une probabilité de 0,5.

Question

La fonction S-BOX est responsable principalement de :

A. La confusion
B. La diffusion

Reponse

La confusion.

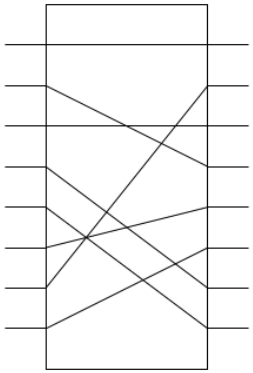


Figure: P-BOX

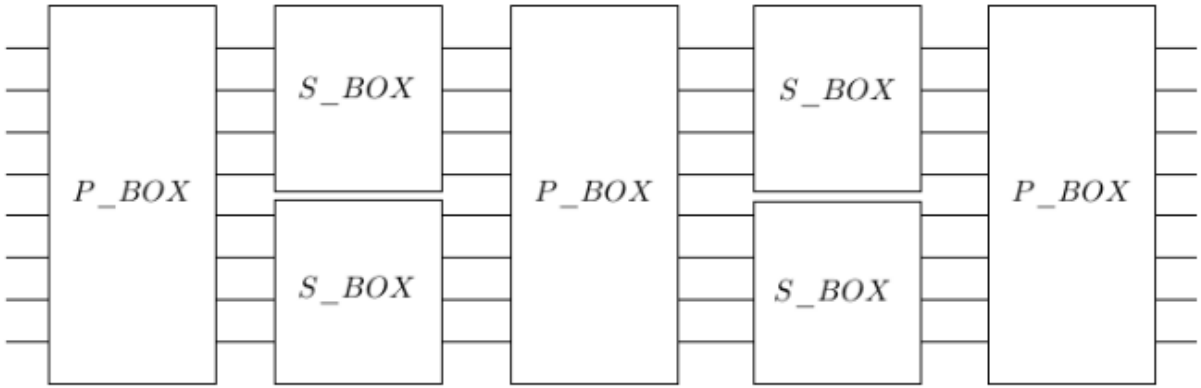


Figure: Machine de Lucifer

Confusion

Renforcement de la non linéarité de la transformation.

La diffusion

Inversion d'un bit en entrée provoque une inversion de chaque bit en sortie avec une probabilité de 0,5.

Question

La fonction P-BOX est responsable principalement de :

A. La confusion
B. La diffusion

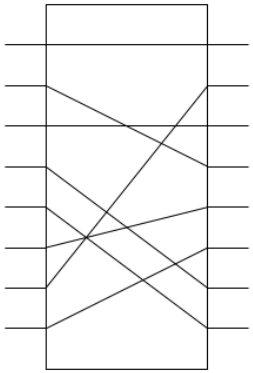


Figure: P-BOX

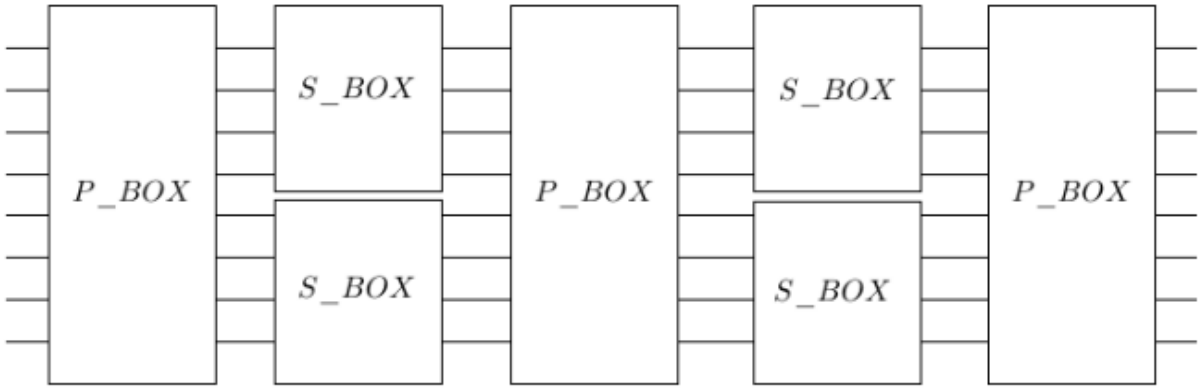


Figure: Machine de Lucifer

Confusion

Renforcement de la non linéarité de la transformation.

La diffusion

Inversion d'un bit en entrée provoque une inversion de chaque bit en sortie avec une probabilité de 0,5.

Question

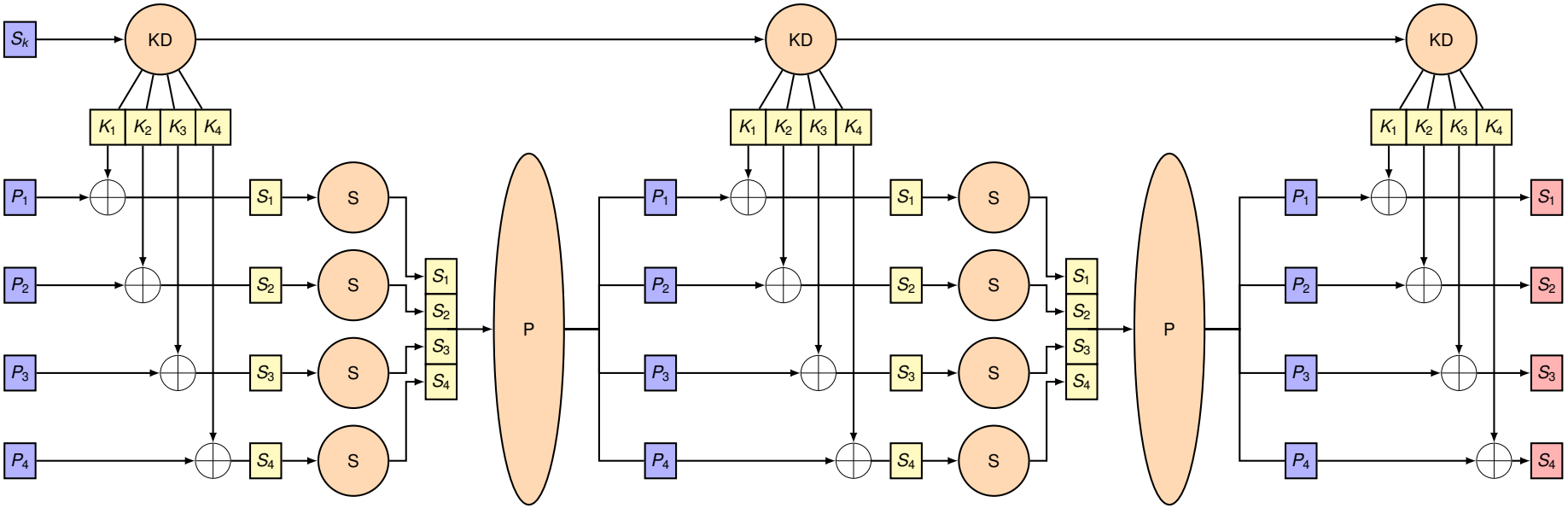
La fonction P-BOX est responsable principalement de :

A. La confusion
B. La diffusion

Reponse

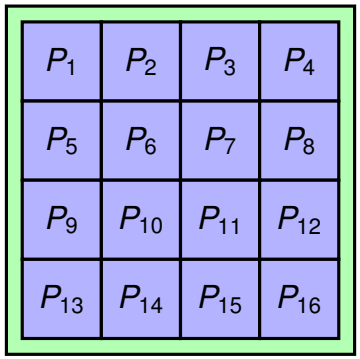
De manière unitaire, rien de particulier (voir exemple sur AES), mais lorsqu'elle est combinée avec une S-BOX et chaînée, contribue à la diffusion.

Chiffrement symétrique moderne : Réseau SP



Exemple d'un réseau SP avec un bloc de 4 symboles (rappel, AES 16 symboles par blocs).

- S_k clé de chiffrement
- *KD* : Algorithme de dérivation de clé qui extrait de l'entropie de la clé pour générer une clé de tour dont la taille est égale à celle du bloc.
- *S* : Opération de substitution, *P* : Opération de permutation.



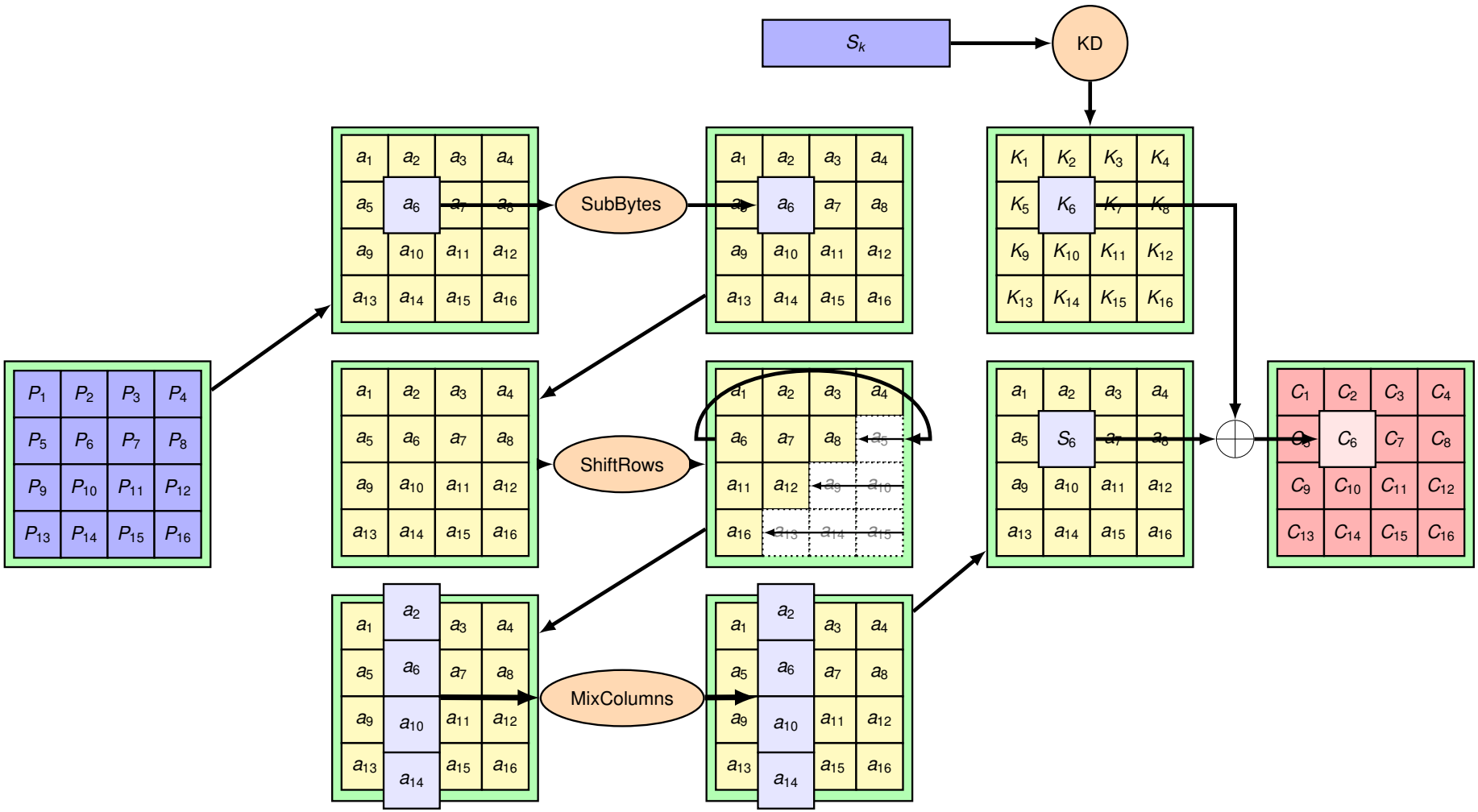
- Développé par Joan Daemen et Vincent Rijmen (Belgique).
- Vainqueurs en 2000 de la compétition “AES” du NIST.
- Construction moderne : La sécurité et les contraintes d’implémentations étudiées de manière simultanée pour chaque fonctions internes.

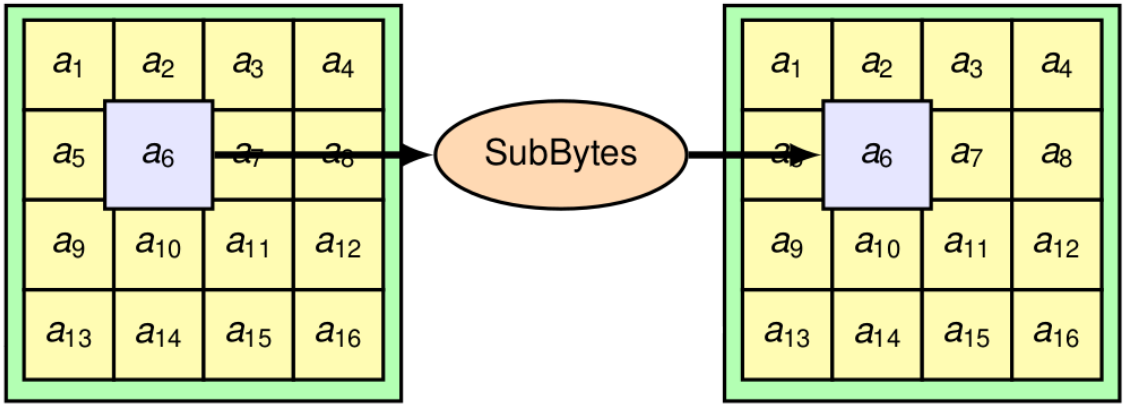
Caractéristiques

- Le clair est découpé en blocs de 16 octets.
- On chiffre bloc par bloc pour obtenir de chiffré.
- Le bloc va subir un certain nombre de transformations appelé réseau SP.
- Opérations dans le corps fini $\mathbb{Z}_2[X]/m(x)$, où $m(x)$ vaut :

$$m(x) = x^8 + x^4 + x^3 + x + 1 \tag{3}$$

AES :Description détaillée d'un tour



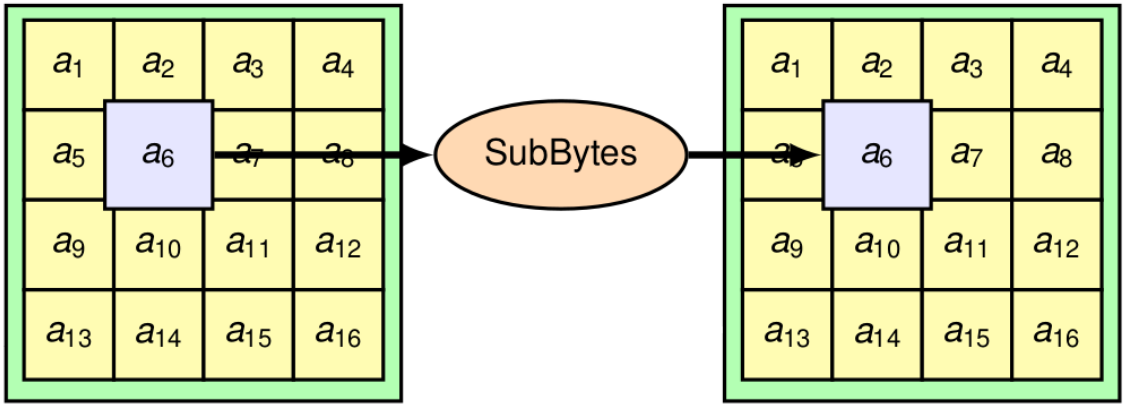


- SubByte fonction de substitution (responsable de la confusion).
- La transformation est appliquée sur chaque octet du bloc.
- L'entrée va subir 2 transformations successives :
  1. Calcul de l'inverse :

$$F(x) = \begin{cases} x^{-1} & , si \quad x \neq 0 \\ 0 & , si \quad x = 0 \end{cases} \tag{4}$$

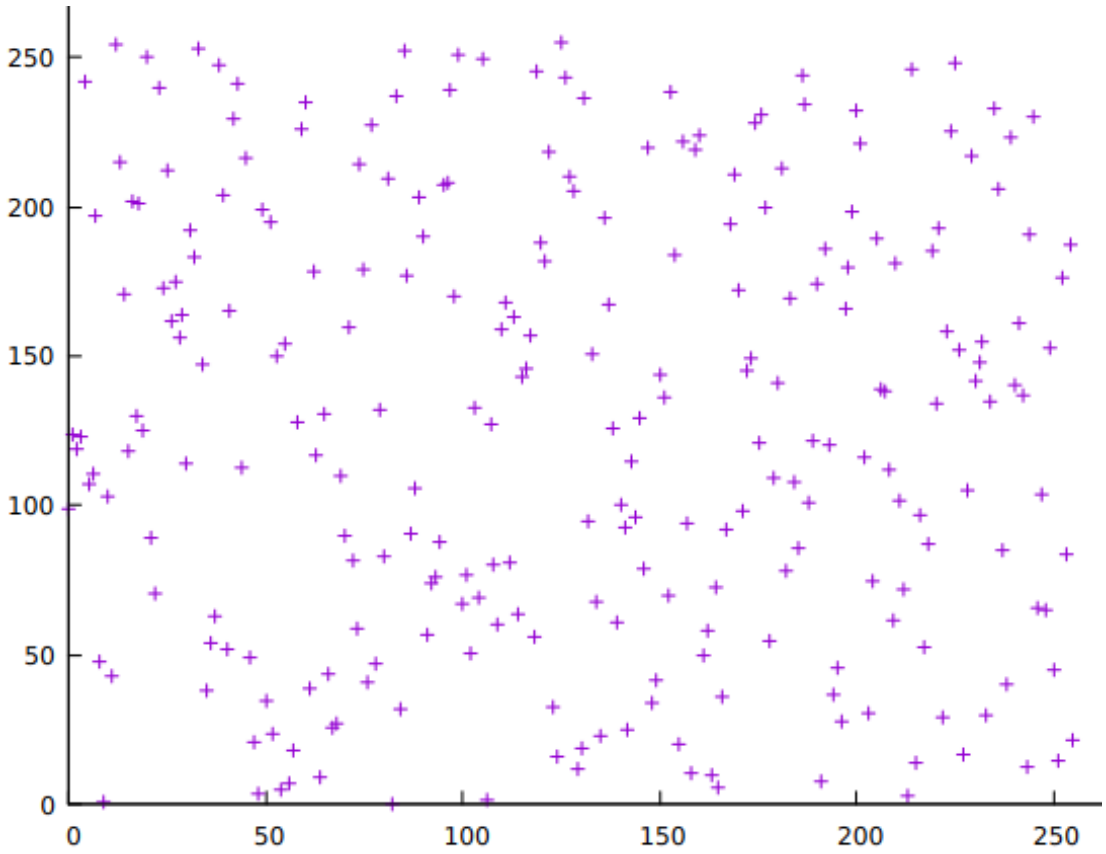
  2. Transformation affine :

$$b(x) = (x^6 + x^5 + x + 1) + a(x)(x^7 + x^6 + x^5 + x^4 + 1) \mod x^8 + 1 \tag{5}$$



Caractéristiques

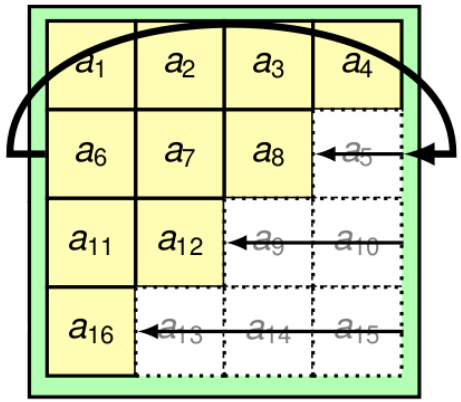
1. inversibilité ;
2. minimisation de la plus grande corrélation entre une combinaison des entrée et une combinaison des sorties ;
3. minimisation de la plus grande valeur dans la table EXOR ;
4. complexité de représentation dans GF(2^8) ;
5. description simple.



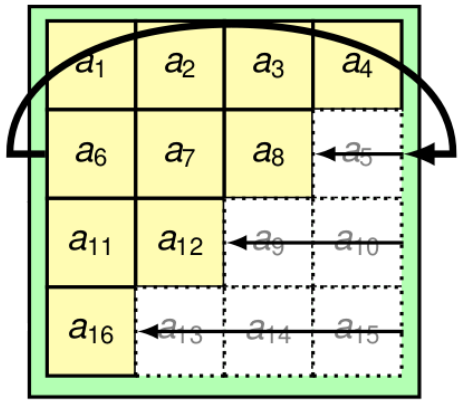
y = SubByte(x)

Coefficient de corrélation : -0.04

AES : ShiftRows



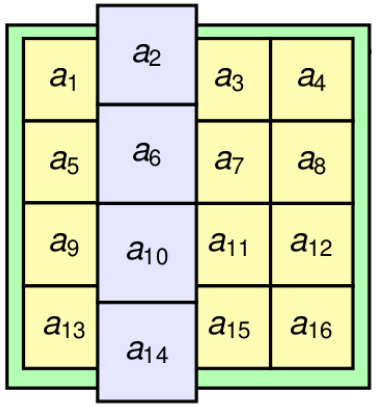
- ShiftRows fonction de permutation (responsable de la diffusion).
- Rotation des lignes vers la gauche de 8 bits.



- ShiftRows fonction de permutation (responsable de la diffusion).
- Rotation des lignes vers la gauche de 8 bits.

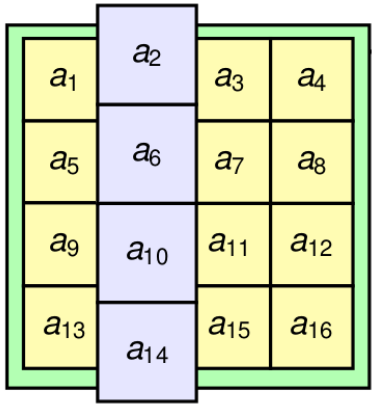
Caractéristiques

1. inversibilité ;
2. linéarité dans GF(2^8) ;
3. propriété de diffusion suffisante ;
4. rapidité de calcul sur processeur 8 bits ;
5. symétrie ;
6. description simple.



- MixColumn fonction de permutation.
- Rappel : La permutation doit favoriser la diffusion, i.e. réduire la redondance statistique que l'on peut trouver sur le clair.
- Multiplication de chaque colonne dans GF$(2^8)/(x^4+1)$ par la matrice :

$$\begin{bmatrix} b_1 \\ b_5 \\ b_9 \\ b_{13} \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \times \begin{bmatrix} a_1 \\ a_5 \\ a_9 \\ a_{13} \end{bmatrix} \quad (4)$$



Caractéristiques

1. inversibilité ;
2. linéarité dans GF(2^8) ;
3. propriété de diffusion suffisante ;
4. rapidité de calcul sur processeur 8 bits ;
5. symétrie ;
6. description simple.

Méthodologie

1. Génération d'un état initial aléatoire.
2. Inversion d'un bit.
3. Comparaison des bits de sortie entre l'état initial et l'état avec 1 bit inversé.

Expérience 1 : ShiftRows + MixColumns

- État initial :

10000100	01001000	11010101	10111100
00100000	10000100	11001011	01011101
11011100	01101000	01010010	00010000
01001011	00110101	01001000	00010010

- Inversion du 4^{eme} bit du premier octet.
- Après 1 tour :

00001000	00010000	00010000	00011000	
00000000	00000000	00000000	00000000	(5 différences / 128)
00000000	00000000	00000000	00000000	
00000000	00000000	00000000	00000000	

- Après 10 tours :

00000100	00001000	00001000	00001100	
00011000	00011000	00010100	00001100	(23 différences / 128)
00010000	00011000	00001000	00010000	
00011000	00001000	00010000	00010000	

Méthodologie

1. Génération d'un état initial aléatoire.
2. Inversion d'un bit.
3. Comparaison des bits de sortie entre l'état initial et l'état avec 1 bit inversé.

Expérience 2 : SubByte + ShiftRows + MixColumns

- État initial :

10000100	01001000	11010101	10111100
00100000	10000100	11001011	01011101
11011100	01101000	01010010	00010000
01001011	00110101	01001000	00010010

- Inversion du 4^{eme} bit du premier octet.
- Après 1 tour :

00001101	00011010	00011010	00010111	
00000000	00000000	00000000	00000000	(13 différences / 128)
00000000	00000000	00000000	00000000	
00000000	00000000	00000000	00000000	

- Après 10 tours :

01001101	11010101	01110010	01000000	
10011001	10101000	10011100	11001110	(60 différences / 128)
00100000	01101111	00101101	01100001	
01101110	01101001	11001010	00110001	

Méthodologie

1. Génération d'un état initial aléatoire.
2. Inversion d'un bit.
3. Comparaison des bits de sortie entre l'état initial et l'état avec 1 bit inversé.

Expérience 3 : SubByte

- État initial :

10000100	01001000	11010101	10111100
00100000	10000100	11001011	01011101
11011100	01101000	01010010	00010000
01001011	00110101	01001000	00010010

- Inversion du 4^{eme} bit du premier octet.
- Après 1 tour :

00011010	00000000	00000000	00000000
00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000

(3 différences / 128)

- Après 10 tours :

00001110	00000000	00000000	00000000
00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000
00000000	00000000	00000000	00000000

(3 différences / 128)

Sécurité

- A ce jour, pas de vulnérabilité connue sur AES complet, la meilleur attaque étant l'attaque par biclique divisant par 4 la recherche de la clé (2 bits de sécurité) $\rightarrow$ Taille de la clé $\approx$ niveau de sécurité ;
- Des attaques existent sur des versions réduites (moins de tour) ;
- Concernant le post-quantique, niveau de sécurité divisé par 2 à cause de Grover $\rightarrow$ choisir une clé de minimum 192 bits.
- Dans le standard actuel, il existe 3 niveaux de sécurité :

Nom	Longueur de clé (bits)	Niveau de sécurité	tours
AES-128	128	$\approx$126	10
AES-192	192	$\approx$190	12
AES-256	256	$\approx$254	14

Nom	Longueur de clé (bits)	Niveau de sécurité	tours
AES-128	128	$\approx$126	10
AES-192	192	$\approx$190	12
AES-256	256	$\approx$254	14

Question

On considère AES-256. Combien de bit de message je peux chiffrer ?

Nom	Longueur de clé (bits)	Niveau de sécurité	tours
AES-128	128	≈126	10
AES-192	192	≈190	12
AES-256	256	≈254	14

Question

On considère AES-256. Combien de bit de message je peux chiffrer ? 128 bits.

Question

On considère AES-128. Comparé au OTP, j'ai une taille de clé :

- Plus petite ;
- Plus grande ;
- Identique.

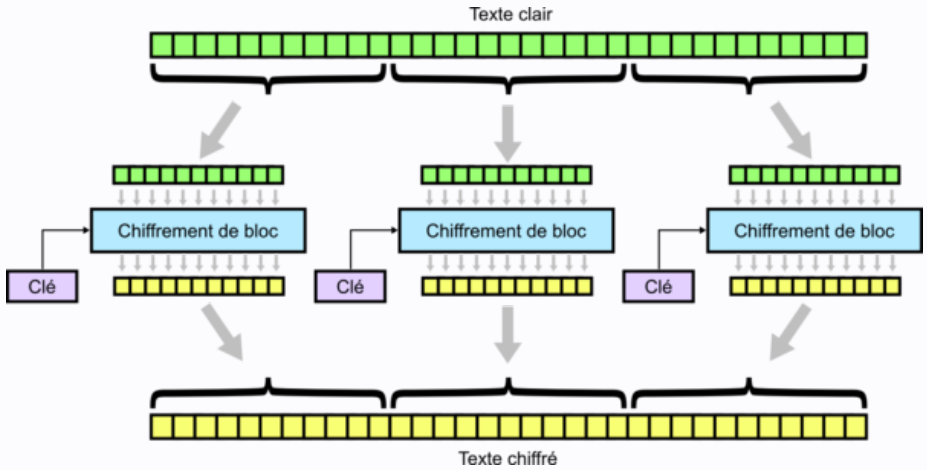
Question

On considère AES-128. Comparé au OTP, j'ai une taille de clé :

- Plus petite ;
- Plus grande ;
- Identique.

Comment chiffrer un message plus grand que la taille du bloc ?

Mode d'Opération Electronic Code Book (ECB)

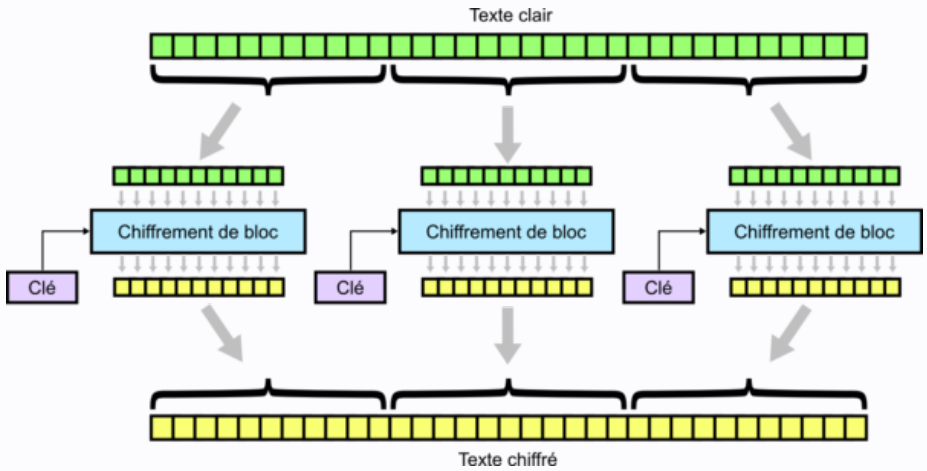


Construction

Construction très simple reposant par la découpe du message en bloc correspondant à la taille du bloc de l'algorithme de chiffrement, puis chiffrement bloc par bloc.

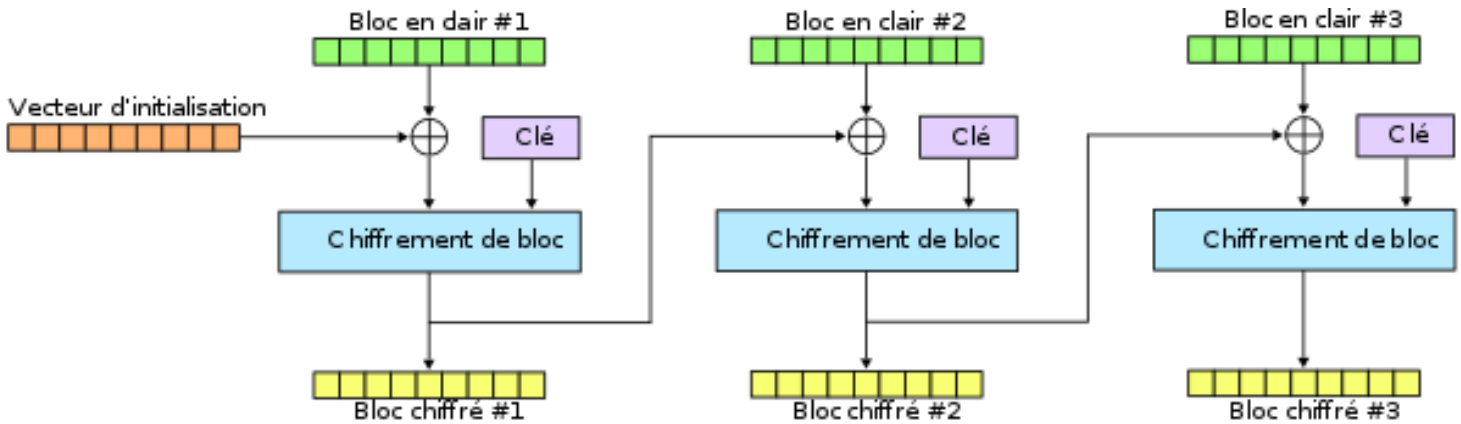
Mode d'Opération Electronic Code Book (ECB)

Vulnerability HIGH, use at your own risk



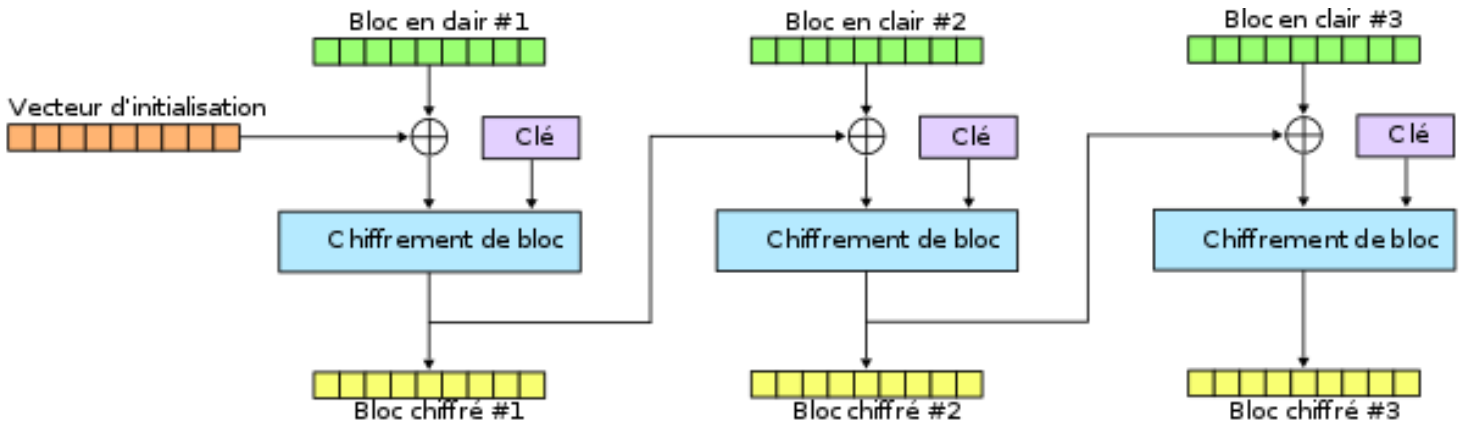
Caractéristiques

- Avantages :
  - Simple
  - Parallélisable
  - Déchiffrement du bloc souhaité indépendamment des autres.
- Désavantages :
  - Un gruyère en terme de sécurité. Notamment, à votre avis que se passe-t-il si deux blocs de messages sont égaux ?



Construction

Construction reprenant le principe de ECB avec l'ajout d'une rétroaction en ajoutant au bloc de message suivant le chiffré du bloc précédent.

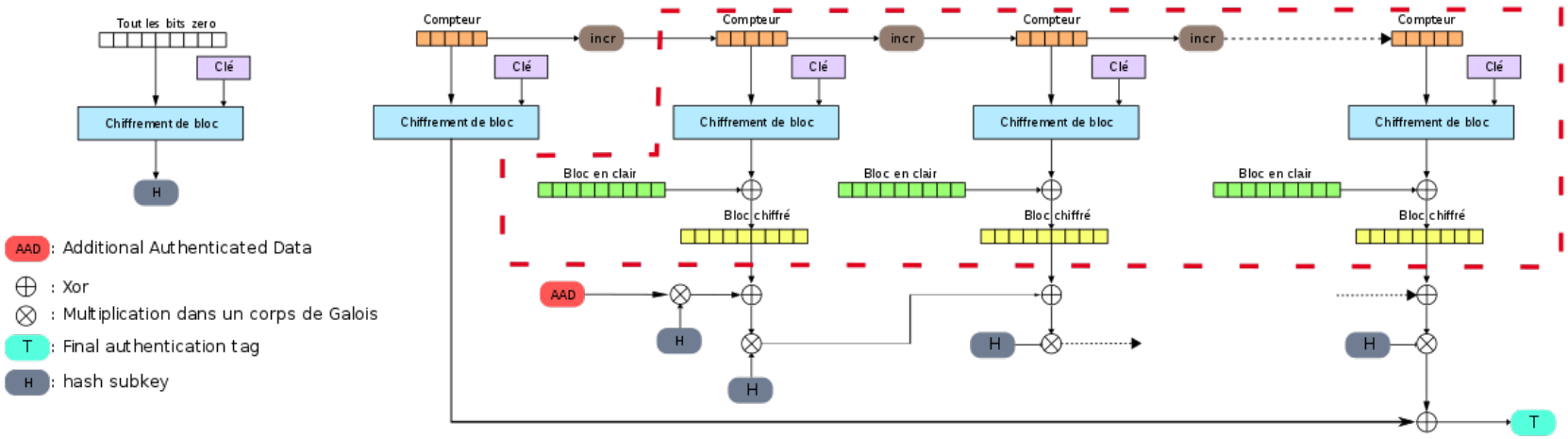


Caractéristiques

- Avantages :
  - Simple
  - Déchiffrement du bloc souhaité possible sans avoir à déchiffrer les autres.
- Désavantages :
  - Non parallélisable (pour le chiffrement).
  - Utilisable en pratique (a fait partie du standard pendant de nombreuses années, et est encore utilisé) à condition de faire bien attention au vecteur d'initialisation (initial, mais aussi celui pour le prochain message envoyé).

Warning Warning Warning Warning Warning Warning Warning Warning

Vulnerability LOW, refer to the documentation for best practice



Construction

Standard actuel (TLSv1.3). AES n'est plus utilisé pour chiffrer le message directement, il est utilisé pour générer une clé de grande taille (à partir d'un compteur), puis cette clé est ajouté au message (comme le Chiffre de Vernam).

Configuration minimale des standards actuels

TLSv1.2 (la majorité des machines actuelles ?)

AES-128 en mode CBC.

TLSv1.3

AES-128 en mode GCM.

Note vis-à-vis de la machine quantique

Ni TLSv1.2 et TLSv1.3 ne sont sécurisés face à une machine quantique.

- **Chiffrement symétrique :** algorithme de Grover permet de faire une recherche exhaustive sur N éléments en $\sqrt{N}$ opérations $\rightarrow$ division par deux du niveau de sécurité d'AES actuel.
- **Mécanisme asymétrique :** Les standards actuels reposent (ou peuvent se réduire) au problème du logarithme discret, solvable en temps polynomial sur une machine quantique avec l'algorithme de Shor $\rightarrow$ Ils sont donc considérés cassés.

Configuration minimale des standards actuels

TLSv1.2 (la majorité des machines actuelles ?)

AES-128 en mode CBC.

TLSv1.3

AES-128 en mode GCM.

Note vis-à-vis de la machine quantique

Ni TLSv1.2 et TLSv1.3 ne sont sécurisés face à une machine quantique.

- **Chiffrement symétrique :** algorithme de Grover permet de faire une recherche exhaustive sur N éléments en $\sqrt{N}$ opérations $\rightarrow$ division par deux du niveau de sécurité d'AES actuel.
- **Mécanisme asymétrique :** Les standards actuels reposent (ou peuvent se réduire) au problème du logarithme discret, solvable en temps polynomial sur une machine quantique avec l'algorithme de Shor $\rightarrow$ Ils sont donc considérés cassés.

TLSv1.2 (la majorité des machines actuelles ?)

AES-128 en mode CBC.

TLSv1.3

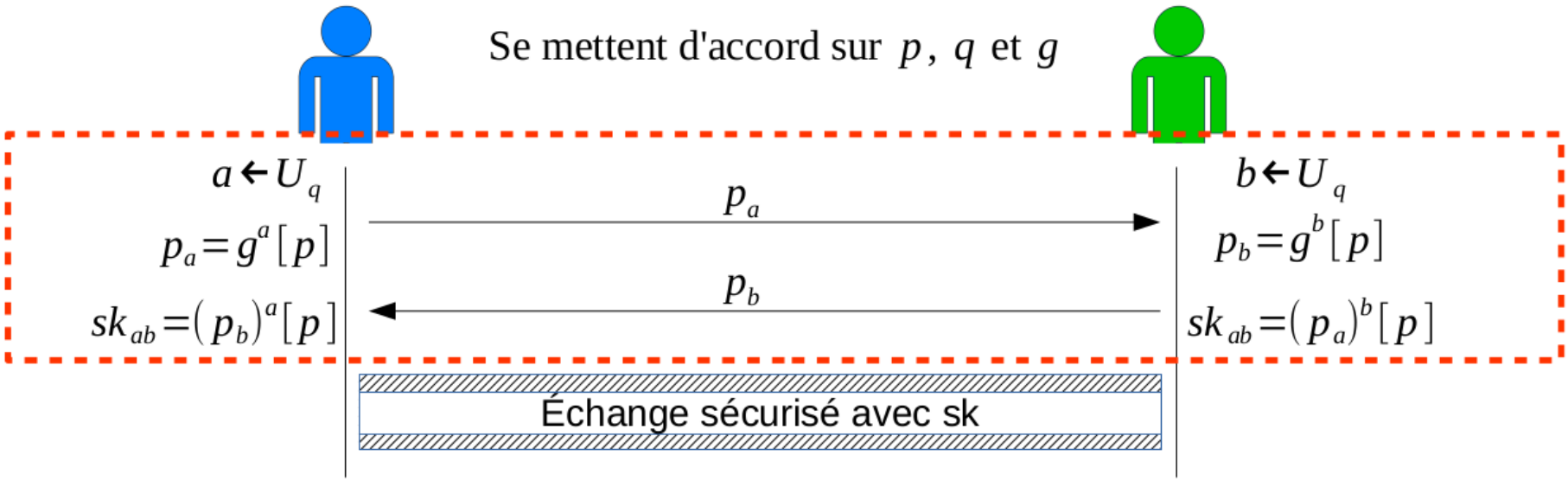
AES-128 en mode GCM.

Note vis-à-vis de la machine quantique

Ni TLSv1.2 et TLSv1.3 ne sont sécurisés face à une machine quantique.

- **Chiffrement symétrique :** algorithme de Grover permet de faire une recherche exhaustive sur N éléments en $\sqrt{N}$ opérations $\rightarrow$ division par deux du niveau de sécurité d'AES actuel.
- **Mécanisme asymétrique :** Les standards actuels reposent (ou peuvent se réduire) au problème du logarithme discret, solvable en temps polynomial sur une machine quantique avec l'algorithme de Shor $\rightarrow$ Ils sont donc considérés cassés.

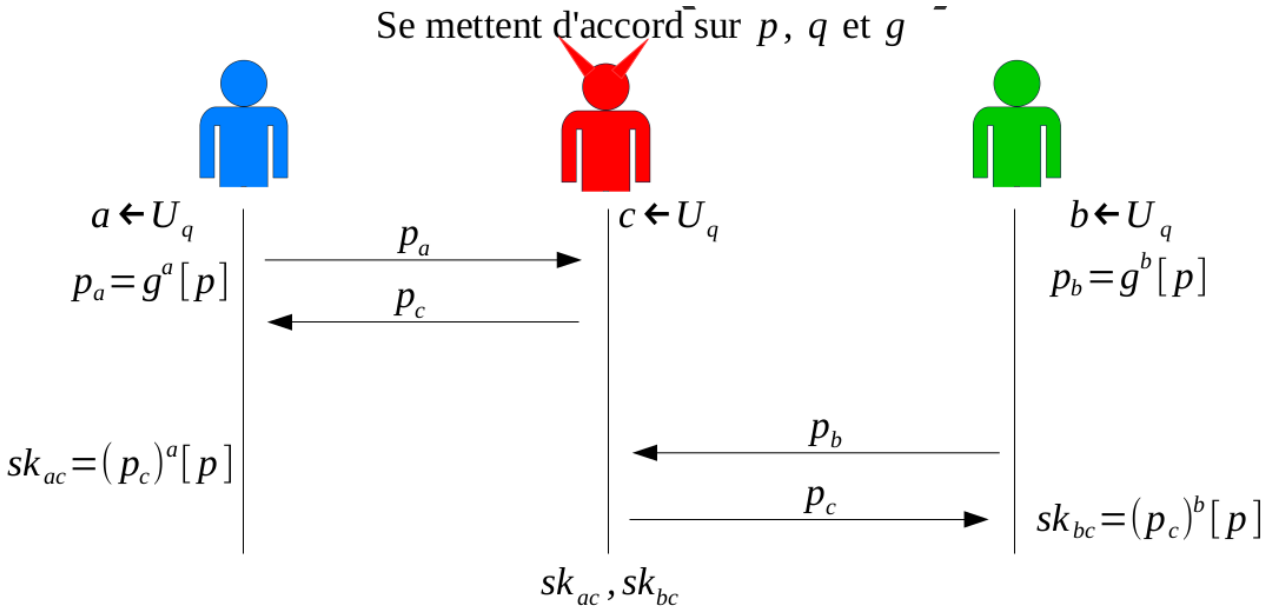
Mécanismes asymétriques (pour l'échange de clé)



Construction

- Soit q un nombre premier tel que $p = 2.q + 1$ est aussi premier (par exemple un nombre premier de Sophie Germain) ;
- Alors on peut trouver $g \in \mathbb{Z}_p$ tel que :

$$\forall (i, j) \in \mathbb{Z}_q,\ g^i \neq g^j [p]$$
$$g^q = 1 [p]$$



Sécurité : Attaque de l'homme au milieu

Un attaquant interceptant les paramètres publiques peut construire 2 échanges Diffie-Hellman avec son propre paramètre publique $\rightarrow$ Attaque de l'homme au milieu.

En pratique, les clés sont certifiées par un tier de confiance qui permet d'éviter l'usurpation d'identité (si le tier est réellement de confiance).

RSA (1977)

- RSA est un algorithme de chiffrement ayant une clé différente pour la fonction de chiffrement et de déchiffrement ;
- Il repose sur une fonction à trappe, c'est-à-dire une fonction bijective facile à appliquer, mais dont la recherche de l'antécédent est complexe dans le cas général mais simple en connaissant un paramètre particulier ;
- La sécurité de RSA repose sur la complexité de la factorisation d'entiers de très grande taille (plusieurs milliers de bits).

- RSA est un algorithme de chiffrement ayant une clé différente pour la fonction de chiffrement et de déchiffrement ;
- Il repose sur une fonction à trappe, c'est-à-dire une fonction bijective facile à appliquer, mais dont la recherche de l'antécédent est complexe dans le cas général mais simple en connaissant un paramètre particulier ;
- La sécurité de RSA repose sur la complexité de la factorisation d'entiers de très grande taille (plusieurs milliers de bits).

Fonctionnement

- p et q deux premiers distincts ;
- $N = p \cdot q$ définissant l'anneau $\mathbb{Z}_N$;
- $\phi(N)$ l'indicatrice d'Euler évaluée en N ;
- e et d deux entiers tels que $e \cdot d \equiv 1[\phi(N)]$.
- $KeyGen(1^\lambda)$: pour un niveau de sécurité λ, renvoie $P_k = (e, N)$ et $S_k = d$;
- $Encrypt(P_k, m) = m^e[N] = C$;
- $Decrypt(S_k, C) = C^d[N] = m$.

Sécurité

- Pas d'attaque critique connu, mais des instances faibles existent (premiers p et q trop proches, réutilisation de N, etc...)
- Ne repose pas directement sur le log discret, mais vulnérable également à Shor $\rightarrow$ pas post-quantique.

Ouverture : Couplage Code/Cryptographie dans le cadre de la cryptographie post-quantique

Cas particulier de $\mathbb{Z}_2$

- Soit $\mathbb{Z}_2$ le corps à 2 éléments $\{0, 1\}$. Les mots de longueur n sont définis dans $\mathbb{Z}_2^n$.
- Un code de longueur n est un sous-espace vectoriel $C \in \mathbb{Z}_2^n$ de dimension k.
- Pour d la distance minimale du code, le type du code C est donné par le triplet (n, k, d).

Matrice génératrice

Soit un code de type [n,k]. On dit que $G \in \mathbb{Z}_2^{k \times n}$ est une matrice génératrice de C si :

$$C = \{\mu G | \mu \in \mathbb{Z}_2^k\} \tag{4}$$

Matrice de parité

Soit un code de type [n,k]. On dit que $H \in \mathbb{Z}_2^{(n-k) \times n}$ est une matrice de parité si H est une matrice de génératrice du code dual. On a en particulier la propriété suivante :
Si $x \in C$, alors $Hx^T = 0$

Définition de l'addition

Pour deux éléments $(x, y, c) \in \nu^3$. On définit l'addition de x par y dont l'unique résultat est donné par c de la manière suivante :

$$c_k = x_k \oplus y_k, k \in \{0, 1, \cdots, n-1\} \tag{5}$$

Définition de la multiplication

Pour deux éléments $(x, y, c) \in \nu^3$. On définit la multiplication de x par y dont l'unique résultat est donné par c de la manière suivante :

$$c_k = \sum_{i+j \equiv k[n]} x_i y_j, k \in \{0, 1, \cdots, n-1\} \tag{6}$$

Matrice circulante

Soit $x = (x_0, x_1, \cdots, x_{n-1}) \in \mathbb{Z}_2^n$, on note la matrice circulante de x rot qui est définie de la manière suivante :

$$\mathrm{rot}(x) = \begin{pmatrix} x_0 & x_{n-1} & . & . & x_1 \\ x_1 & x_0 & . & . & x_2 \\ . & . & & & . \\ . & . & & & . \\ x_{n-1} & x_{n-2} & . & . & x_0 \end{pmatrix}$$

Propriétés de la matrice circulante

$$x \cdot y = x.\mathrm{rot}(y)^\top = y.\mathrm{rot}(x)^\top = y.x$$

$$x \cdot y = \begin{pmatrix} x_0 & x_1 & x_2 & \cdots & x_{n-1} \end{pmatrix} \begin{pmatrix} y_0 & y_1 & . & . & y_{n-1} \\ y_{n-1} & y_0 & . & . & y_{n-2} \\ . & . & & & . \\ . & . & & & . \\ y_1 & y_2 & . & . & y_0 \end{pmatrix}$$

Matrice circulante

Soit $x = (x_0, x_1, \cdots, x_{n-1}) \in \mathbb{Z}_2^n$, on note la matrice circulante de x rot qui est définie de la manière suivante :

$$\mathrm{rot}(x) = \begin{pmatrix} x_0 & x_{n-1} & . & . & x_1 \\ x_1 & x_0 & . & . & x_2 \\ . & . & & & . \\ . & . & & & . \\ x_{n-1} & x_{n-2} & . & . & x_0 \end{pmatrix}$$

Propriétés de la matrice circulante

$$x \cdot y = x.\mathrm{rot}(y)^\top = y.\mathrm{rot}(x)^\top = y.x$$

$$x \cdot y = \begin{pmatrix} x_0 & x_1 & x_2 & \cdots & x_{n-1} \end{pmatrix} \begin{pmatrix} y_0 & y_1 & . & . & y_{n-1} \\ y_{n-1} & y_0 & . & . & y_{n-2} \\ . & . & & & . \\ . & . & & & . \\ y_1 & y_2 & . & . & y_0 \end{pmatrix}$$

Matrice circulante

Soit $x = (x_0, x_1, \cdots, x_{n-1}) \in \mathbb{Z}_2^n$, on note la matrice circulante de x rot qui est définie de la manière suivante :

$$\mathrm{rot}(x) = \begin{pmatrix} x_0 & x_{n-1} & . & . & x_1 \\ x_1 & x_0 & . & . & x_2 \\ . & . & & & . \\ . & . & & & . \\ x_{n-1} & x_{n-2} & . & . & x_0 \end{pmatrix}$$

Propriétés de la matrice circulante

$$x \cdot y = x.\mathrm{rot}(y)^\top = y.\mathrm{rot}(x)^\top = y.x$$

$$x \cdot y = \begin{pmatrix} x_0 & x_1 & x_2 & \cdots & x_{n-1} \end{pmatrix} \begin{pmatrix} y_0 & y_1 & . & . & y_{n-1} \\ y_{n-1} & y_0 & . & . & y_{n-2} \\ . & . & & & . \\ . & . & & & . \\ y_1 & y_2 & . & . & y_0 \end{pmatrix}$$

Rappel sur l'équivalence des codes

Soit C un code linéaire quelconque de matrice génératrice G. Alors C est équivalent à un code systématique, i.e. :

$$\exists (P, S) \in \mathbb{Z}_2^{k \times k} \times \mathbb{Z}_2^{n \times n} \; \exists A \in \mathbb{Z}_2^{k \times n-k} \; S \cdot G \cdot P = \begin{bmatrix} I_d & A \end{bmatrix} \tag{7}$$

Remarque

- Il est difficile en pratique de décoder un code linéaire aléatoire.
- De plus, pour certains types de code (par exemple les codes de Goppa), il est difficile, en donnant à quelqu'un un code linéaire équivalent choisi aléatoirement, de retrouver les matrices S et P choisies.

Rappel sur l'équivalence des codes

Soit C un code linéaire quelconque de matrice génératrice G. Alors C est équivalent à un code systématique, i.e. :

$$\exists (P, S) \in \mathbb{Z}_2^{k \times k} \times \mathbb{Z}_2^{n \times n} \; \exists A \in \mathbb{Z}_2^{k \times n-k} \; S \cdot G \cdot P = \begin{bmatrix} I_d & A \end{bmatrix} \tag{7}$$

Remarque

- Il est difficile en pratique de décoder un code linéaire aléatoire.
- De plus, pour certains types de code (par exemple les codes de Goppa), il est difficile, en donnant à quelqu'un un code linéaire équivalent choisi aléatoirement, de retrouver les matrices S et P choisies.

Construction d'une fonction à trappe (chiffrement asymétrique)

On va se donner un type de code dont il est difficile de calculer S et P en pratique (notons le C), nous allons :

1. Calculer la matrice génératrice de C, que l'on note G ; (que l'on garde jalousement)
2. Générer une matrice S inversible et une matrice de permutation P ; (que l'on garde également jalousement)
3. Calculer le code équivalent que l'on note $\widehat{C}$, de matrice génératrice $\widehat{G}$. (que l'on donne à tout le monde)

Chiffrement

Partant d'un message m, on calcule l'unique mot de code de $\widehat{C}$ grâce à $\widehat{G}$. On introduit un certain nombre d'erreurs dans la limite des capacités de correction du code C.

Déchiffrement

Pour la personne possédant uniquement $\widehat{G}$, il faut être capable, en un temps raisonnable, de trouver les matrices S et P pour tomber sur le code C. Pour les personnes ayant (G, S , P). Ainsi, notre correspondant est capable de supprimer les erreurs volontairement introduites, et non un adversaire.

Construction d'une fonction à trappe (chiffrement asymétrique)

On va se donner un type de code dont il est difficile de calculer S et P en pratique (notons le C), nous allons :

1. Calculer la matrice génératrice de C, que l'on note G ; (que l'on garde jalousement)
2. Générer une matrice S inversible et une matrice de permutation P ; (que l'on garde également jalousement)
3. Calculer le code équivalent que l'on note $\widehat{C}$, de matrice génératrice $\widehat{G}$. (que l'on donne à tout le monde)

Chiffrement

Partant d'un message m, on calcule l'unique mot de code de $\widehat{C}$ grâce à $\widehat{G}$. On introduit un certain nombre d'erreurs dans la limite des capacités de correction du code C.

Déchiffrement

Pour la personne possédant uniquement $\widehat{G}$, il faut être capable, en un temps raisonnable, de trouver les matrices S et P pour tomber sur le code C. Pour les personnes ayant (G, S , P). Ainsi, notre correspondant est capable de supprimer les erreurs volontairement introduites, et non un adversaire.

Construction d'une fonction à trappe (chiffrement asymétrique)

On va se donner un type de code dont il est difficile de calculer S et P en pratique (notons le C), nous allons :

1. Calculer la matrice génératrice de C, que l'on note G ; (que l'on garde jalousement)
2. Générer une matrice S inversible et une matrice de permutation P ; (que l'on garde également jalousement)
3. Calculer le code équivalent que l'on note $\widehat{C}$, de matrice génératrice $\widehat{G}$. (que l'on donne à tout le monde)

Chiffrement

Partant d'un message m, on calcule l'unique mot de code de $\widehat{C}$ grâce à $\widehat{G}$. On introduit un certain nombre d'erreurs dans la limite des capacités de correction du code C.

Déchiffrement

Pour la personne possédant uniquement $\widehat{G}$, il faut être capable, en un temps raisonnable, de trouver les matrices S et P pour tomber sur le code C. Pour les personnes ayant (G, S , P). Ainsi, notre correspondant est capable de supprimer les erreurs volontairement introduites, et non un adversaire.

Instantiation pédagogique de McEliece

Choix d'un code de départ :

- Qui nous permet de décoder simplement ;
- Dont il est difficile, donnant un code équivalent choisi aléatoirement, de retrouver les transformations effectuées.

Proposition

Choix d'un code systématique :

- Qui nous permet de décoder simplement ?
- Dont il est difficile, donnant un code équivalent choisi aléatoirement, de retrouver les transformations effectuées ?

Choix d'un code de départ :

- Qui nous permet de décoder simplement ;
- Dont il est difficile, donnant un code équivalent choisi aléatoirement, de retrouver les transformations effectuées.

Proposition

Choix d'un code systématique :

- Qui nous permet de décoder simplement ?
- Dont il est difficile, donnant un code équivalent choisi aléatoirement, de retrouver les transformations effectuées ?

Choix d'un code de départ :

- Qui nous permet de décoder simplement ;
- Dont il est difficile, donnant un code équivalent choisi aléatoirement, de retrouver les transformations effectuées.

Proposition

Choix d'un code systématique :

- Qui nous permet de décoder simplement ? $\rightarrow$ OK
- Dont il est difficile, donnant un code équivalent choisi aléatoirement, de retrouver les transformations effectuées ? $\rightarrow$ FAUX

Paramètres

$$S = \begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 \end{pmatrix}, G = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix}, P = \begin{pmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Question

- Calculez la capacité de correction du code de départ.
- Calculez le chiffré de $m = \begin{pmatrix} 1 & 0 & 1 & 0 \end{pmatrix}$ en introduisant un maximum d'erreur.

Fin

Annexe (Hors programme)

Notations

On $C.Encode(x) = y$ la fonction qui permet d'encoder un message à partir du code C, et $C.Decode(y)$ la fonction qui permet de décoder un message.

Notations

On $C.Encode(x) = y$ la fonction qui permet d'encoder un message à partir du code C, et $C.Decode(y)$ la fonction qui permet de décoder un message.

Setup(1^λ)

Pour un niveau de sécurité λ, déterminer les paramètres du code $(n, k, d, w) = \text{param}$.

Notations

On $C.Encode(x) = y$ la fonction qui permet d'encoder un message à partir du code C, et $C.Decode(y)$ la fonction qui permet de décoder un message.

Setup(1^λ)

Pour un niveau de sécurité λ, déterminer les paramètres du code $(n, k, d, w) = \text{param}$.

Keygen(param)

- Générer $q_r \in \nu$;
- Calculer $Q = \begin{pmatrix} I_n & \text{rot}(q_r) \end{pmatrix}$;
- G génératrice du code C ;
- $(x, y) \in \nu^2$ tel que $\omega(x) = \omega(y) = w$;
- $S_k = (x, y)$, $P_k = (G, Q, s = S_k.Q^\top)$

Notations

On $C.Encode(x) = y$ la fonction qui permet d'encoder un message à partir du code C, et $C.Decode(y)$ la fonction qui permet de décoder un message.

Setup(1^λ)

Pour un niveau de sécurité λ, déterminer les paramètres du code $(n, k, d, w) = \text{param}$.

Keygen(param)

- Générer $q_r \in \nu$;
- Calculer $Q = \begin{pmatrix} I_n & \text{rot}(q_r) \end{pmatrix}$;
- G génératrice du code C ;
- $(x, y) \in \nu^2$ tel que $\omega(x) = \omega(y) = w$;
- $S_k = (x, y)$, $P_k = (G, Q, s = S_k.Q^\top)$

Remarque : Le paramètre s ne permet pas de retrouver la clé secrète d'après le problème de décodage de syndrome.

Construction d'un algorithme de chiffrement

$S_k = (x, y)$, $P_k = (G, Q, s = S_k.Q^\top)$

$S_k = (x, y)$, $P_k = (G, Q, s = S_k.Q^\top)$

Chiffrement : $\mathrm{Encrypt}(P_k, \mu, \theta)$

- Générer $\epsilon \in \nu$, $(r_1, r_2) \in \nu^2$ tels que $\omega(\epsilon) = \omega(r_1) = \omega(r_2) = w$;
- Calculer $v^\top = Q.r^\top$;
- Calculer $\rho = \mu.G + s.r_2 + \epsilon$;
- Émettre (ρ, v).

$S_k = (x, y)$, $P_k = (G, Q, s = S_k.Q^\top)$

Chiffrement : $\mathrm{Encrypt}(P_k, \mu, \theta)$

- Générer $\epsilon \in \nu$, $(r_1, r_2) \in \nu^2$ tels que $\omega(\epsilon) = \omega(r_1) = \omega(r_2) = w$;
- Calculer $v^\top = Q.r^\top$;
- Calculer $\rho = \mu.G + s.r_2 + \epsilon$;
- Émettre (ρ, v).

Question

Pourquoi ρ n'est pas décodable directement ?

$S_k = (x, y)$, $P_k = (G, Q, s = S_k.Q^\top)$

Chiffrement : $\mathrm{Encrypt}(P_k, \mu, \theta)$

- Générer $\epsilon \in \nu$, $(r_1, r_2) \in \nu^2$ tels que $\omega(\epsilon) = \omega(r_1) = \omega(r_2) = w$;
- Calculer $v^\top = Q.r^\top$;
- Calculer $\rho = \mu.G + s.r_2 + \epsilon$;
- Émettre (ρ, v).

Question

Pourquoi ρ n'est pas décodable directement ?

Reponse

$s.r_2$ génère trop de bruit pour être décodable en l'état. Pour rappel, $s = S_k.Q^\top = x + y.q_r$, avec x et y de poids w, mais pas q_r qui est quelconque.

Construction d'un algorithme de chiffrement

$S_k = (x, y)$, $P_k = (G, Q, s = S_k.Q^\top)$.
$v^\top = Q.r^\top$, $\rho = \mu.G + s.r_2 + \epsilon$.

$S_k = (x, y)$, $P_k = (G, Q, s = S_k.Q^\top)$.
$v^\top = Q.r^\top$, $\rho = \mu.G + s.r_2 + \epsilon$.

Déchiffrement : $\mathrm{Decrypt}(S_k, \rho, v)$

- Calculer $C.Decode(\rho - v.y)$

$S_k = (x, y)$, $P_k = (G, Q, s = S_k.Q^\top)$.
$v^\top = Q.r^\top$, $\rho = \mu.G + s.r_2 + \epsilon$.

Déchiffrement : $\mathrm{Decrypt}(S_k, \rho, v)$

- Calculer $C.Decode(\rho - v.y)$

Question

Sous quelle condition le décodage est correct ?

$S_k = (x, y)$, $P_k = (G, Q, s = S_k.Q^\top)$.
$v^\top = Q.r^\top$, $\rho = \mu.G + s.r_2 + \epsilon$.

Déchiffrement : $\mathrm{Decrypt}(S_k, \rho, v)$

- Calculer $C.Decode(\rho - v.y)$

Question

Sous quelle condition le décodage est correct ?

Reponse

$$\begin{aligned}\rho - v.y &= \mu.G + s.r_2 + \epsilon - r.Q^\top.y \\ &= \mu.G + s.r_2 + \epsilon - r_1.y - r_2.q_r.y \\ &= \mu.G + x.r_2 + y.q_r.r_2 + \epsilon - r_1.y - r_2.q_r.y \\ &= \mu.G + x.r_2 + \epsilon - r_1.y\end{aligned} \tag{8}$$

Il faut donc que $\rho - v.y$ soit plus petit que la capacité de décodage, sachant que x, y, r_1, r_2 et ϵ ont tous un poids de Hamming de w.