The Big APN problem !

initiation à la recherche,
Toulon,
13 décembre 2018.

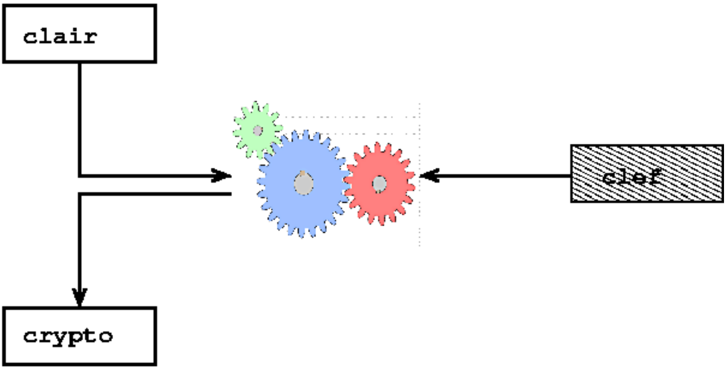
Philippe Langevin

IMATH, université de Toulon

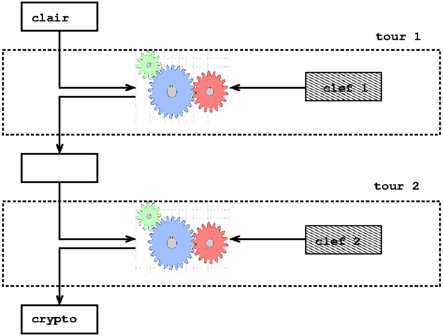
last revision December 20, 2018.

Sommaire

chiffrement par bloc



tours



Algorithme Rijndael

Le Rijndael a été conçu par Joan Daemen et Vincent Rijmen, dans le but de devenir un candidat à AES du NIST.

Après avoir réussi à se classer dans les six premiers, Rijndael a été choisi comme standard en 2000, prenant la place du premier véritable standard de la cryptographie : le DES.

Le chiffrement utilise une longueur de blocs variable, une longueur de clé variable et un nombre de rondes variable. En revanche, Rijndael version AES est restreint à des longueurs de clé de 128, 192 et 256 bits avec une longueur de bloc fixée à 128 bits.

NIST National Institute of Standards and Technology

AES Advanced Encryption Standard

DES Data Encryption System

Description de l'AES

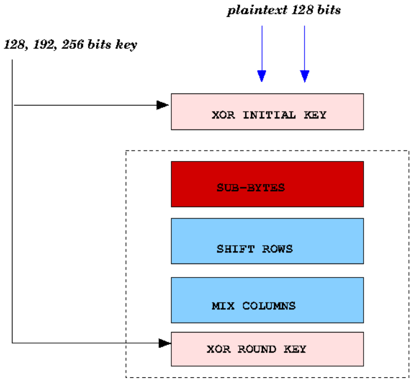
L'AES applique un réseau de substitutions et permutations, sur un bloc de 16 octets (128 bits) représenté par un tableau 4x4:

$$\begin{bmatrix} b_0 & b_4 & b_8 & b_{12} \\ b_1 & b_5 & b_9 & b_{13} \\ b_2 & b_6 & b_{10} & b_{14} \\ b_3 & b_7 & b_{11} & b_{15} \end{bmatrix}$$

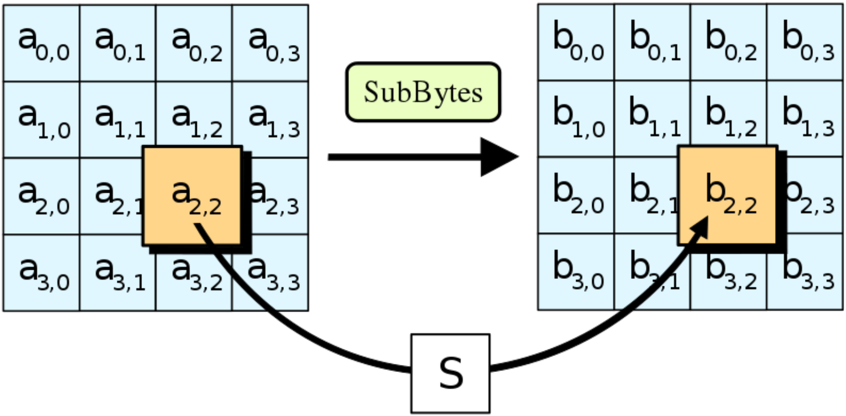
Le nombre de tours dépend de la taille des clés de chiffrements :

- 10 tours pour une clé de 128-bits.
- 12 tours pour une clé de 192-bits.
- 14 tours une clé de 256-bits.

Description de l'AES

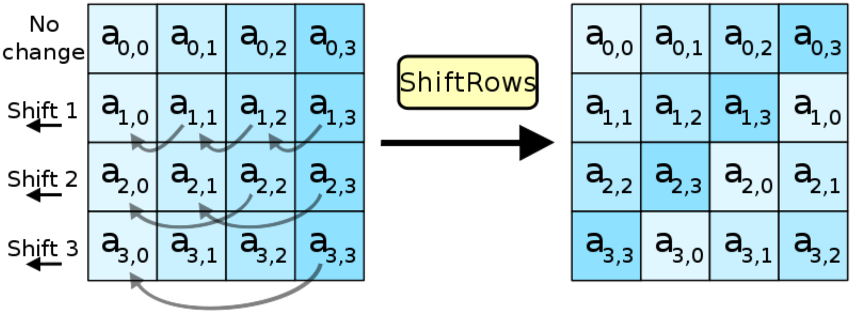


sub-bytes

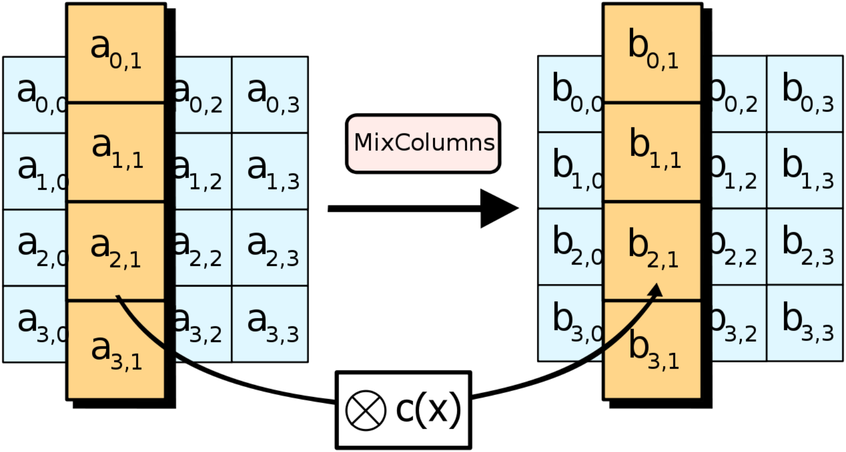


$a_{0,0}$ $a_{0,1}$ $a_{0,2}$ $a_{0,3}$
$a_{1,0}$ $a_{1,1}$ $a_{1,2}$ $a_{1,3}$
$a_{2,0}$ $a_{2,1}$ $a_{2,2}$ $a_{2,3}$
$a_{3,0}$ $a_{3,1}$ $a_{3,2}$ $a_{3,3}$
SubBytes
$b_{0,0}$ $b_{0,1}$ $b_{0,2}$ $b_{0,3}$
$b_{1,0}$ $b_{1,1}$ $b_{1,2}$ $b_{1,3}$
$b_{2,0}$ $b_{2,1}$ $b_{2,2}$ $b_{2,3}$
$b_{3,0}$ $b_{3,1}$ $b_{3,2}$ $b_{3,3}$
S

shift-rows


No change
Shift 1
Shift 2
Shift 3
a0,0 a0,1 a0,2 a0,3
a1,0 a1,1 a1,2 a1,3
a2,0 a2,1 a2,2 a2,3
a3,0 a3,1 a3,2 a3,3
ShiftRows
a0,0 a0,1 a0,2 a0,3
a1,1 a1,2 a1,3 a1,0
a2,2 a2,3 a2,0 a2,1
a3,3 a3,0 a3,1 a3,2

mix-colums



$a_{0,0}$ $a_{0,1}$ $a_{0,2}$ $a_{0,3}$
$a_{1,0}$ $a_{1,1}$ $a_{1,2}$ $a_{1,3}$
$a_{2,0}$ $a_{2,1}$ $a_{2,2}$ $a_{2,3}$
$a_{3,0}$ $a_{3,1}$ $a_{3,2}$ $a_{3,3}$
MixColumns
$\otimes c(x)$
$b_{0,0}$ $b_{0,1}$ $b_{0,2}$ $b_{0,3}$
$b_{1,0}$ $b_{1,1}$ $b_{1,2}$ $b_{1,3}$
$b_{2,0}$ $b_{2,1}$ $b_{2,2}$ $b_{2,3}$
$b_{3,0}$ $b_{3,1}$ $b_{3,2}$ $b_{3,3}$

confusion diffusion

Dans son analyse des méthodes de chiffrement, Claude Shanon introduit les notions :

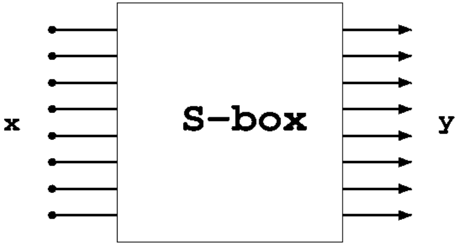
- confusion
- diffusion

Dans l'AES, les opérations MixColumn et ShiftRow travaillent de concert pour apporter de la diffusion dans le système de chiffrement et l'opération SubByte introduit de la confusion.

choix des S-box La transformation S ne peut pas être aléatoire ! Les attaques différentielles, linéaires, algébrique ... imposent des conditions difficiles à satisfaire.

1. complexité algébrique
2. nonlinéarité
3. uniformité différentielle

S-box



D'un point de vue mathématique, une S-box (à n-entrées) est modélisée par une permutation F de l'ensemble des mots binaires de n bits. Les images d'une S-box sont stockées dans une table.

Application vectorielle

Une S-box de dimension n transforme un mot de n bits en un mot de n bits, on parle d'*application vectorielle* :

$$F\colon \{0,1\}^n \longrightarrow \{0,1\}^n \qquad f\colon \{0,1\}^n \longrightarrow \{0,1\}$$

complètement défnie par n fonctions booléennes.

$$x \mapsto F(x) = \big(f_1(x), f_2(x), \dots, f_n(x)\big)$$

dimension	dim. booléenne	card. vectorielle
n	2^n	2^{n2^n}
4	16	2^{64}
6	64	∞
8	256	?

Sommaire

addition

Pour un entier n fixé, on note $\{0,1\}^n$ l'ensemble des mots de n bits.

$$x = (x_n, \ldots, x_2, x_1) = \frac{1}{2}\sum_{i=1}^{n} x_i 2^i$$

L'addition bit-à-bit sur ces mots :

$$(x \oplus y)_i = x_i \text{ xor } y_i$$

- associative
- commutative
- 0 est neutre
- un mot est son propre opposé !

composante

On introduit le *produit scalaire*:

$$x.y = x_1y_1 \oplus x_2y_2 \oplus \cdots \oplus x_ny_n$$

Les composantes d'une application vectorielle F sont les fonctions booléennes de la forme :

$$x \mapsto y.F(x)$$

Une fonction vectorielle possède 2^n composantes.

degré algébrique

Une fonction *booléenne* envoie les mots de n bits dans $\{0,1\}$. Elle possède plusieurs représentation polynomiales, le degré minimal est le degré de la fonction.
Par exemple,

$$x \mapsto \delta_0(x) = \begin{cases} 1, & x = 0; \\ 0, & x \neq 0. \end{cases}$$

On vérifie que :

$$\delta_0(x) = (x_1 \oplus 1)(x_2 \oplus 1) \cdots (x_n \oplus 1)$$

le développement fait apparaître un polynôme de degré n.

$$\deg(\delta_0) = n$$

non-linéarité

La *distance de Hamming* entre deux fonctions booléennes f et g compte le nombre de différences entre f et g

$$d(f,g) = \sharp\{x \in \{0,1\}^n \mid f(x) \neq g(x)\}$$

La distance de Hamming minimale entre f et une fonction de degré 1 est la *non-linéarité* de f.

critère cryptographique

Les composantes booléenne d'une S-box doivent maximiser

- degré
- non-linéarité

et minimiser

- défaut différentiel

Sommaire

défaut différentiel

On note $\delta(u, v)$ le nombre de solutions :

$$F(x \oplus u) \oplus F(x) = v$$

On définit le *défaut différentiel* de l'application vectorielle F par :

$$\Delta(F) = \max_{u \neq 0, v} \delta(u, v)$$

critère cryptographique

Une S-box doit minimiser le défaut différentiel.

Almost Perfect Nonlinearity

Pour $u \neq 0$ et $v\{0,1\}^n$ fixés, l'ensemble des solutions de l'équation

$$F(x \oplus u) \oplus F(x) = v$$

est invariant par translation de u, et donc :

$$2 \leq \Delta(f)$$

Définition

APN Une application vectorielle de défaut différentiel optimal 2.

Propriété

Théorème (flat property)

Une application vectorielle F est APN si et seulement si pour tout $x < y < z < t$:

$$x \oplus y \oplus z \oplus t = 0 \Longrightarrow F(x) \oplus F(y) \oplus F(z) \oplus F(t) \neq 0$$

Un exemple de fonction APN en dimension 2:

$$0 \mapsto 0 \quad 1 \mapsto 1 \quad 2 \mapsto 2 \quad 3 \mapsto 0$$

En effet,

$$0 \oplus 1 \oplus 2 \oplus 3 = 0$$

Sommaire

corps des complexes

Un polynôme de degré n à coefficients dans le corps des nombres complexes possède n racines.

- associativité, commutativité
- distributivité
- opposé, inverse.

Définition (corps fini)

Un ensemble fini muni d'une addition et d'une multiplication vérifiant les proriétés usuelles.

Remarque

En particulier, dans un corps fini, un polynôme de degré 2 possède au plus 2 racines !

corps fini

Théorème (corps fini)

Il existe une structure de corps fini à q éléments ssi q est primaire i.e.

$$q = p^n$$

où p est un nombre premier et n un entier non nul.

Corollaire (corps fini)

L'ensemble des mots de n bits peut être muni d'une structure de corps !

Remarque

Un corps fini $\{0,1\}^n$ possède trois racines cubique ssi n est pair.

cube

La fonction $f : x \mapsto x^3$ est APN !

$$\begin{aligned} f(x \oplus u) \oplus f(x) &= x^3 \oplus 3x^2u \oplus 3xu^2 \oplus u^3 \oplus x^3 \\ &= 3x^2u \oplus 3xu^2 \oplus u^3 \\ &= ux^2 \oplus u^2x \oplus u^3 \end{aligned}$$

$$\Delta(x^3) = 2$$

Théorème

Le cube est une permutation APN de $\{0,1\}^n$ si et ssi n est impair.

inversion

Observons les propriétés différentielles de l'inversion

$$f(x) = \begin{cases} 1/x, & x \neq 0; \\ 0 & x = 0. \end{cases}$$

$$f(x \oplus u) \oplus f(x) = \frac{1}{x \oplus u} \oplus \frac{1}{x} = \frac{u}{x(x \oplus u)}$$

Theorem

L'inversion est une permutation APN de $\{0,1\}^n$ ssi n est impair.

On remarque que $\delta(u, v) \leq 2$ sauf $\delta(1, 1) = 4$ en dimension pair.

Sommaire

Équivalence affine

Ψ un *automorphisme* de $\{0,1\}^n$:

$$\forall x, y \in \{0,1\}^n, \quad \Psi(x \oplus y) = \Psi(x) \oplus \Psi(y)$$

Une *transformation affine* est la composition d'une *translation* et d'un automorphisme. Le caractère APN est invariant par ces transformations.

Définition

Deux applications vectorielles F et G sont affines équivalentes quand il existe deux transformations affines Ψ et Ψ' tel que

$$G(x) = \Psi' \circ F \circ \Psi$$

- 534 classes de cubiques APN en dimension 6.

Équivalence CCZ

Rappelons que le *graphe* d'une application vectorielle F est

$$\Gamma(F) = \{(x, y) \mid y = F(x)\}$$

La meilleure notion d'équivalence introduite par Claude Carlet, Pascale Charpin et Victor Zinoviev:

Définition

Deux applications vectorielles F et G sont CCZ-équivalentes si leurs graphes sont affine équivalent.

Pas facile à comprendre sans notion sur les codes correcteurs d'erreurs !

- Gregor Leander, Marcus Brinkmann (2008) seulement 3 classe APN en dimenion 5
- 534 classes de cubiques APN en dimension 6.
- 13 CCZ-classes de cubiques APN en dimension 6.

Sommaire

choix pour l'AES

En 2000, Daemen et Rijmen, influencés par un article de Gilles Lachaud et Jacques Wolfmann sur la non-linéarité de l'inversion dans un corps fini, ont considéré que le meilleur choix pour la S-box du Rinjael serait :

$$x \mapsto \frac{1}{x}$$

Question Y-a-t-il un meilleur candidat ?
Depuis, le prolème est toujours ouvert !

The big APN problem

Problème

Exist-il une permutation APN des mots de 8 bits ?

2002 l'existence des permutations APN en dimension paire est posée.

2006 Xiang-Dong Hou : pas de permutation APN en dimension 4.

2009 Adam Wolfe et John Dillon découvre une permutation APN en dimension 6.

2012 PL classifie des cubiques APN en dimension 6.

Problème (APN permutation problem)

Exhiber une permutation APN en dimension paire supérieure à 6 !

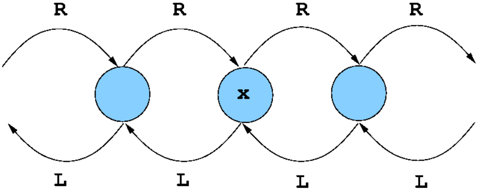
Sommaire

Liens dansants

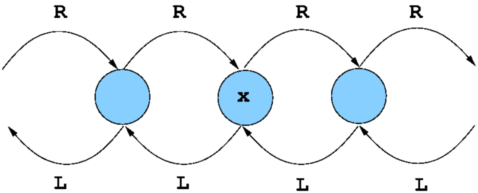
Les Liens dansants (Dancing links) sont une technique suggérée par Knuth pour implémenter de façon efficace l'algorithme X basée sur des réseaux de listes doublement chaînées.

backtracking Knuth utilise DLX pour la recherche de solution du problème de couverture exacte. L'introduction des liens dansants dans une implantation de backtracking accélére les temps de calcul.

délier

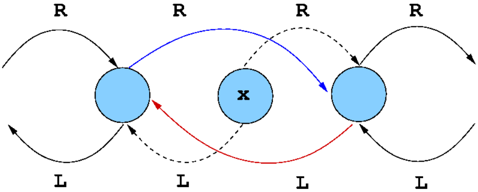


délier

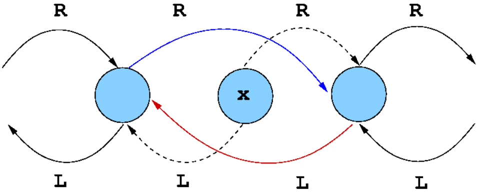


$$L[R[x]] \leftarrow L[x] \quad R[L[x]] \leftarrow R[x]$$

relier



relier

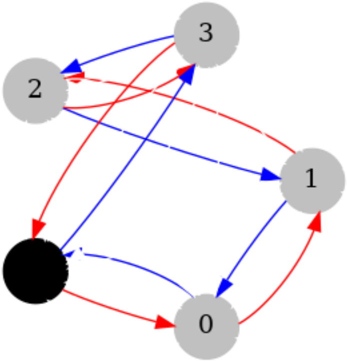


$$L[R[x]] \leftarrow x \quad R[L[x]] \leftarrow x$$

Approche DL des pour les permutations

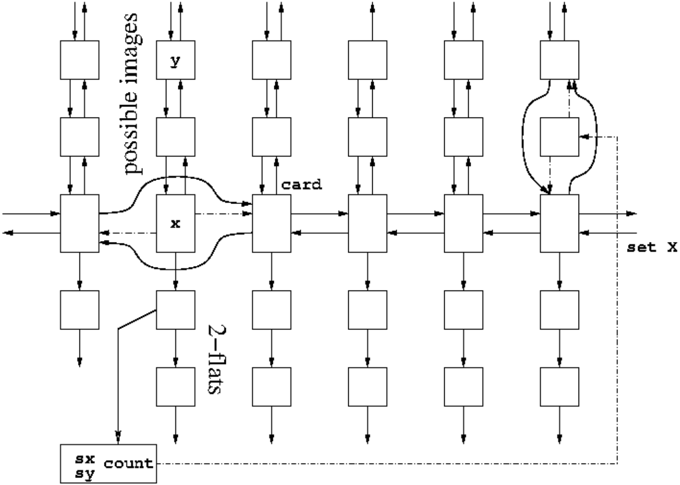
```
void gen( int  r  ,  int  f[]  ,  int  n )
{
int  s  = R[n];                show( n )  ;

if  (  s  == n  ) { count++; return; }

while (  s  < n ) {
            f [ r ]  =  s;
            L[R[s]]  = L[s];
            R[L[s]]  = R[s];
                        gen( r+1, f,  n);
            L[R[s]]  = s;
            R[L[s]]  = s;
            s  = R[s];
      }
}
```

réseau DL-permutation



Voir les liens danser !?

réseau DL-APN



Sommaire

mex

Dans les années 70, John Conway découvre une surprenante structure algébrique sur les nombres entiers, en relation avec la théorie des jeux.

Définition (minimum exclu)

Soit X une partie de l'ensemble des entiers naturels.

$$\mathrm{mex}(X) := \min_{x \notin X} x$$

Comment définir une structure algébrique à partir de la relation d'ordre sur les entiers ?

addition

On souhaite définir une addition $x \boxplus y$ pour obtenir un groupe sur les entiers.
Supposons définies les sommes :

$$a \boxplus y, \quad x \boxplus \qquad a < x \quad b < y$$

$x \boxplus y$ doit être différent de ces sommes !

$$a \boxplus y = x \boxplus y \Longrightarrow a = x$$

Un bon candidat ?

addition

On souhaite définir une addition $x \boxplus y$ pour obtenir un groupe sur les entiers.
Supposons définies les sommes :

$$a \boxplus y, \quad x \boxplus \qquad a < x \quad b < y$$

$x \boxplus y$ doit être différent de ces sommes !

$$a \boxplus y = x \boxplus y \Longrightarrow a = x$$

Un bon candidat ?

$$x \boxplus y = \mathrm{mex}\{a \boxplus y, x \boxplus b \mid a < x, b < y\}$$

Théorème

L'ensemble $\mathbb{N}$ muni de $\boxplus$ est un groupe !

addition

On souhaite définir une addition $x \boxplus y$ pour obtenir un groupe sur les entiers.
Supposons définies les sommes :

$$a \boxplus y, \quad x \boxplus \qquad a < x \quad b < y$$

$x \boxplus y$ doit être différent de ces sommes !

$$a \boxplus y = x \boxplus y \Longrightarrow a = x$$

Un bon candidat ?

$$x \boxplus y = \operatorname{mex}\{a \boxplus y, x \boxplus b \mid a < x, b < y\}$$

Théorème

L'ensemble $\mathbb{N}$ muni de $\boxplus$ est un groupe !

En fait, $x \boxplus y = x \oplus y$!!

multiplication

Essayons de construire un corps ! Il s'agit de définir le produit $x \otimes y$.
Les produits $(x \oplus a) \otimes (y \oplus b)$ ne peuvent pas être nul :

$$(x \oplus a) \otimes (b \oplus y) \neq 0,$$

Le produit $x \otimes y$ doit être différent des éléments :

$$x \otimes b \oplus a \otimes b \oplus a \otimes y$$

Un bon candidat ?

multiplication

Essayons de construire un corps ! Il s'agit de définir le produit $x \otimes y$.
Les produits $(x \oplus a) \otimes (y \oplus b)$ ne peuvent pas être nul :

$$(x \oplus a) \otimes (b \oplus y) \neq 0,$$

Le produit $x \otimes y$ doit être différent des éléments :

$$x \otimes b \oplus a \otimes b \oplus a \otimes y$$

Un bon candidat ?

$$x \otimes y = \mathrm{mex}\{ \quad x \otimes b \oplus a \otimes b \oplus a \otimes y \mid a < x, \quad b < y \quad \}$$

Théorème (Conway)

L'ensemble des entiers naturels muni de l'addition $\oplus$ et de la multiplication $\otimes$ forme un corps !