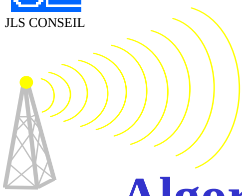
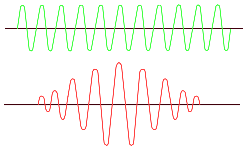
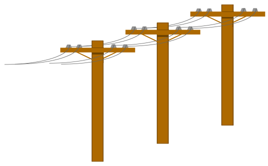
JLS CONSEIL







Algorithmes DES et AES

Jean-Luc Stehlé

Bases mathématiques pour la sécurité informatique

EPITA

6 juin 2013

Jean-Luc.Stehle@NormaleSup.org

JLS CONSEIL

A retenir du cours du 30/05/2013

Notamment les algorithmes

- **Euclide étendu**

  $(a,b) \rightarrow (\lambda,\mu,d)$ tels que $d = \lambda a + \mu b$ avec $d = \mathrm{pgcd}(a,b)$

- **Square and Multiply**

  Indication : Comment décoder un nombre binaire en lisant de gauche à droite

- **Rabin Miller**

  Test de primalité

  utilisant théorème de Fermat et racines carrées non triviales de 1

Sommaire du cours du 06/06/2013

JLS CONSEIL

- **DES** (Data Encryption Standard)
- **Compléments d'algèbre**
- **AES** (Advanced Encryption Standard)

Pour des détails, Consulter les sites du gouvernement américain

`http://csrc.nist.gov/publications`

`http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf`

DES : Principe général

JLS CONSEIL

- **Travaille sur des blocs de 64 bits**
- **Clés de 64 bits, 8 bits de parité, 56 bits utiles**
- **16 rondes**
- **Très rapide en hardware**
- **Facilement implémentable**
- **Algorithme de déchiffrement quasi-identique**

Conçu à l'époque des processeurs 8 bits

DES : Principe général

JLS CONSEIL

- Permutation initiale PI
- 16 itérations (16 rondes)

  $G_i = D_{i-1}$

  $D_i = G_{i-1} \oplus f(D_{i-1}, K_i)$

- Permutation inverse PI^{-1}

DES^{-1} : Il suffit de prendre les clés dans l'autre sens (décalages à droite)

- Sélection des clés K_i (48 bits)

  Clé initiale (56 bits) Permutation PC1

  ==> (g_0,d_0) (28 bits + 28 bits)

  Décalages gauches successifs sur g et d, de $k_i \in [1;2]$

  ==> (g_i,d_i) $(g_{16},d_{16}) = (g_0,d_0)$

  Permutations avec oubli ==>48 bits

  K_i = PC2 (g_i,d_i)

- Fonction f

  fonction d'extension E : D_{i-1} (32 bits) ==> 48 bits

  $\oplus$ K_i (48 bits) ==> 8 blocs de 6 bits

  On leur applique les Sboxes

  (8 boites noires : tableaux 4x16 de 4 bits)

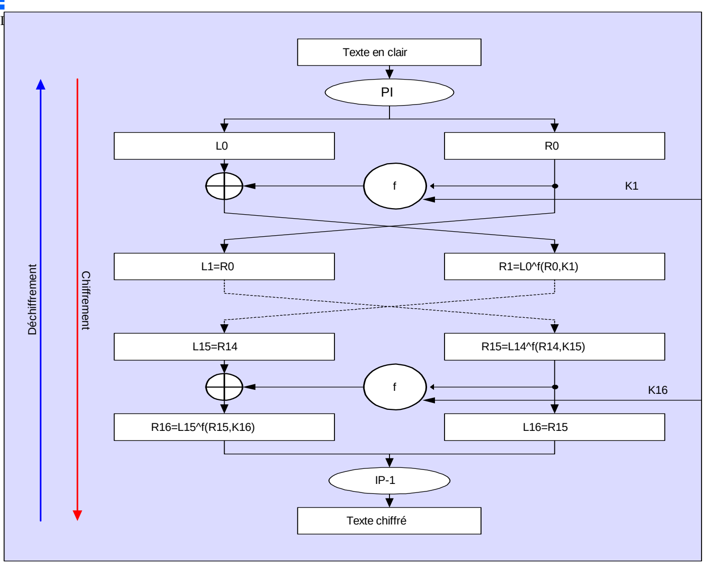
  Bit 1 et 6 = Numéro de ligne Bit 2 à 5 = Numéro de colonne

  On lit 4 bits dans la Sbox ==> **32 bits résultats**

Cf. Robin ou Beckett ou Schneier

DES : Principe général

JLS CONSEIL



JLS CONSEIL

DES : Gestion des clés (1)

Clé initiale à 64 bits, dont 56 utiles + bits de parité

Permutation de clé

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

Décomposé en deux demi clés de 28 bits

Permutation circulaire avant chaque ronde, d'un ou de deux crans vers la gauche

Effectué séparément sur chaque demi clé

Ronde :	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Décalage :	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

$K_{16} = K_0$

Au déchiffrement, on génère les mêmes clés, dans l'ordre inverse, en faisant des décalages vers la droite

Ronde :	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Décalage :	0	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

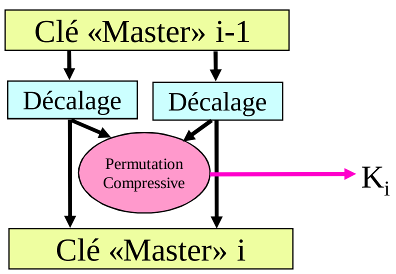
JLS CONSEIL

DES : Gestion des clés (2)

Permutation compressive, génère 48 bits

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

Chaque bit utile de la clé initiale est utilisé approximativement 14 fois durant les 16 rondes



JLS CONSEIL

DES : Permutations initiale et finale

Permutation initiale PI

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

Le bit 58 vient en position 1, le bit 50 en position 2 etc

Mélange les bits des divers octets et met les 32 bits de rang pair en premier

16 rondes

Permutation finale PI^{-1}

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

JLS CONSEIL

Détail des 16 rondes : schéma de Feistel

$G_i = D_{i-1}$ $D_i = G_{i-1} \oplus f(D_{i-1}, K_i)$ **(ronde N° i du chiffrement)**

Cette fonction a pour fonction inverse

$D_{i-1} = G_i$ $G_{i-1} = D_i \oplus f(G_i, K_i)$ **(ronde N° 17-i du déchiffrement)**

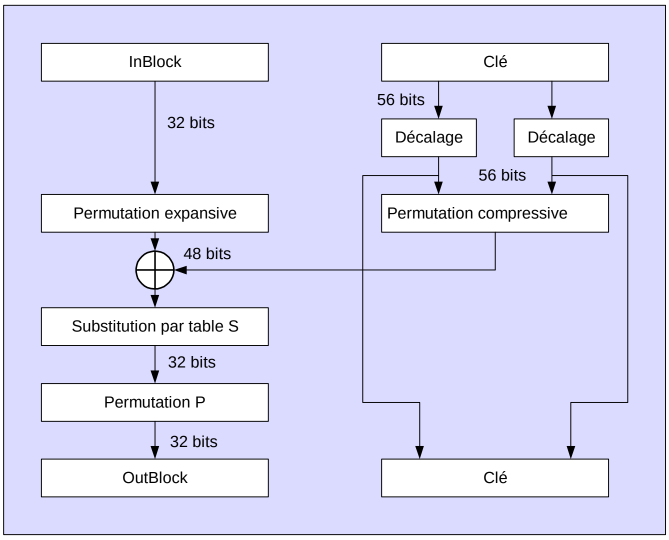
Le déchiffrement est similaire au chiffrement
mais en inversant droite et gauche

Détail de la fonction f (R, K) *(R à 32 bits, K à 48 bits)*

- **Permutation expansive : génère 48 bits à partir de 32 bits**
- **$\oplus$ K_i puis découpé en 8 blocs de 6 bits**
- **Chaque bloc spécifie une entrée dans une Sbox où on lit un résultat à 4 bits**
- **On recolle ces 8 fois 4 bits = 32 bits et la permutation P sur ces 32 bits donne le résultat final**

JLS CONSEIL

La fonction f (R, K)



JLS CONSEIL

La fonction f (R, K) (1)

Phase 1 : Permutation expansive (32 bits => 48 bits)

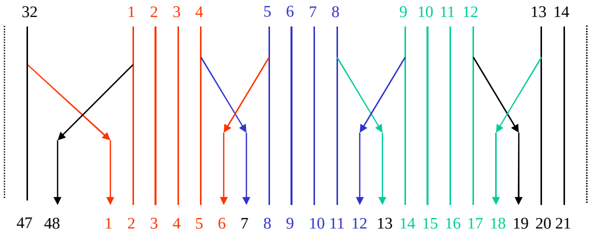
32	1	2	3	4	5	4	5	6	7	8	9
8	9	10	11	12	13	12	13	14	15	16	17
16	17	18	19	20	21	20	21	22	23	24	25
24	25	26	27	28	29	28	29	30	31	32	1

La fonction f (R, K)   (2)

JLS CONSEIL

Phase 1 : Permutation expansive (32 bits => 48 bits)

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

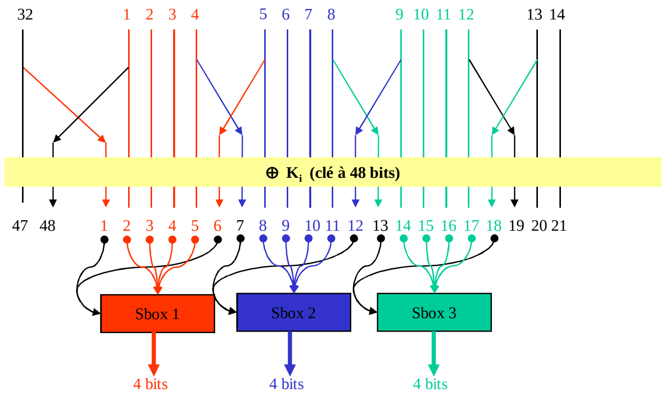


Phase 2 : ⊕ K_i (clé à 48 bits)

La fonction f (R, K)   (3)

JLS CONSEIL

Phase 3 : Entrée dans les 8 S-boxes



Résultats : 8 fois 4 = 32 bits

JLS CONSEIL

Détail des S-boxes

S-box 1

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

... ...

Chaque ligne contient une permutation des entiers entre 1 et 15

JLS CONSEIL

La fonction f (R, K) (4)

Phase 4 : Permutation (32 bits => 32 bits)

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

Le bit 21 vient en position 4, le bit 4 en position 31 etc

JLS CONSEIL

Propriétés du DES

Effet d'avalanche

- **Après peu de passes, chaque bit du texte chiffré dépend de chaque bit du texte en clair et de chaque bit de la clé.**
- **Modifier un seul bit de la clé ou du texte en clair induit une modification d'apparence aléatoire, portant sur environ 32 bits du résultat, ces bits étant répartis de façon apparemment aléatoire.**

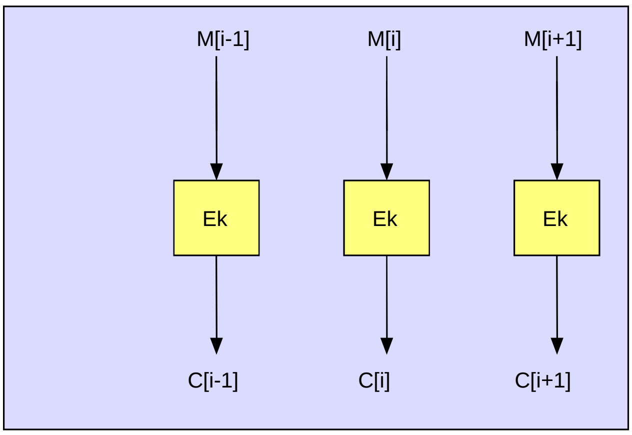
Conception des S-boxes

- **Conçues de façon à contrer tous les types d'attaque connus.**

JLS CONSEIL

Révisions

Mode « standard » ECB



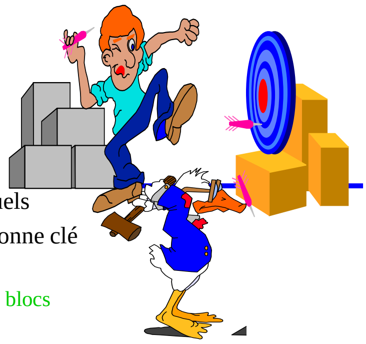
Risque : l'attaquant peut se constituer un carnet de codage = dictionnaire (Electronic code book)

Révisions — DES

JLS CONSEIL

- *Casser le DES ?*
  - **Recherche exhaustive**

    - $2^{56} = 72 \times 10^{15}$ clés à tester
    - A la porté des moyens de calcul actuels
    - Comment savoir qu'on a trouvé la bonne clé
      - Attaque à clair connu
      - D'autant plus facile qu'on a plus de blocs



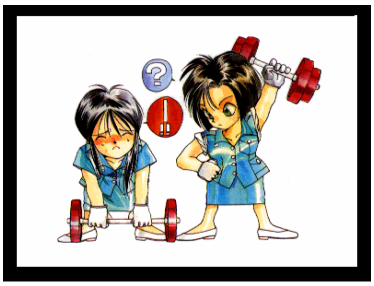
  - **Attaques sans cassage de code**
    - *Blocs rejoués*
    - *Ajouter un MAC ?*

Banque émetteur	15 blocs
Banque bénéficiaire	15 blocs
Nom déposant	6 blocs
Numéro compte	2 blocs
Montant du dépôt	1 bloc

Révisions — Amélioration du DES

JLS CONSEIL

- *Rendre le chiffrement plus fort*

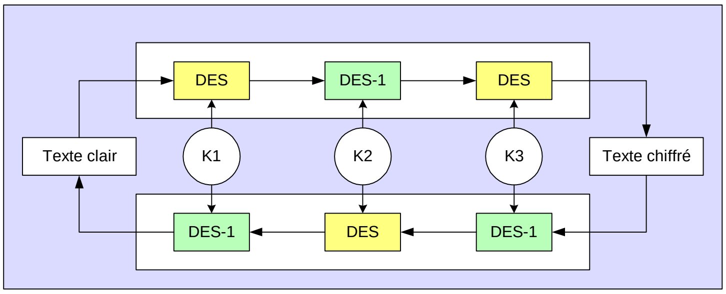


- Triple DES
- Mode chaîné CBC/CTS

Révisions

JLS CONSEIL

Triple DES

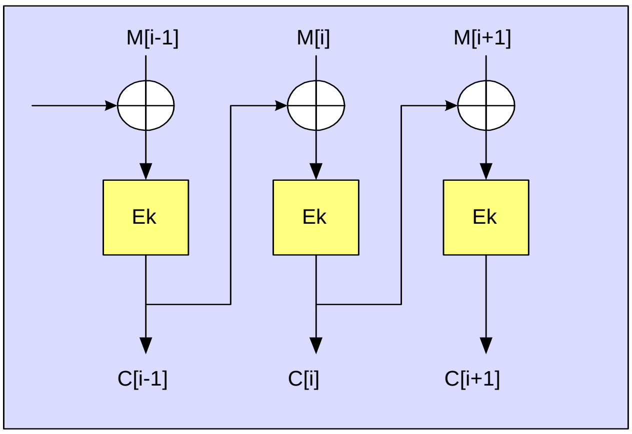


- K1 peut être égal à K3, l'algorithme est alors plus facilement réversible

Révisions

JLS CONSEIL

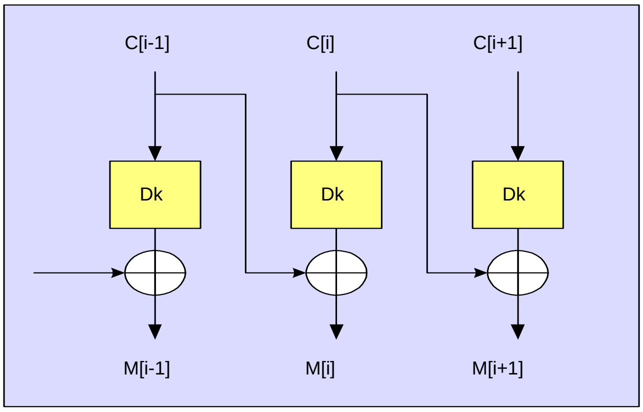
Mode chaîné CBC



Révisions

JLS CONSEIL

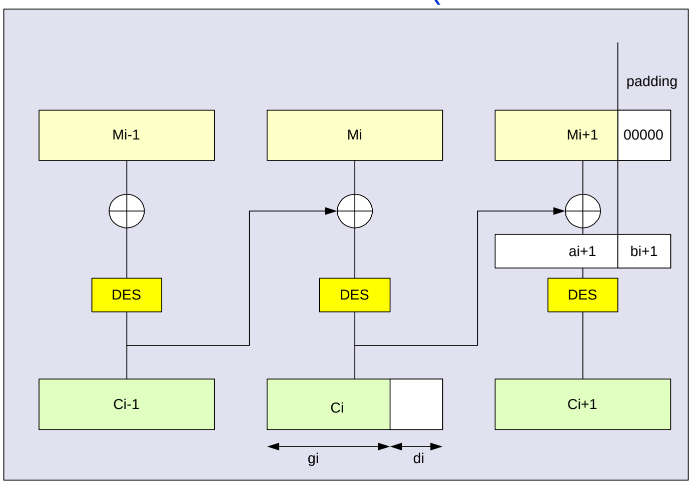
Mode chaîné CBC



Révisions

JLS CONSEIL

Mode chaîné CTS
(Vol de texte chiffré)



Révisions

JLS CONSEIL

Mode CTR

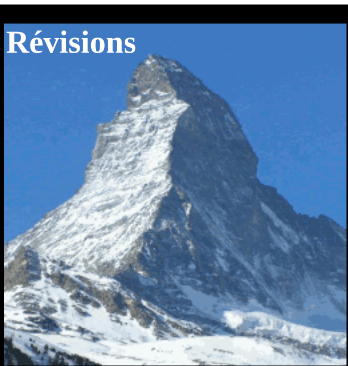
- **On chiffre un compteur, le résultat du chiffrement est XORé avec le texte à chiffrer/déchiffrer**
  - **Chiffrement = déchiffrement**
  - **Pratique pour le chiffrement de supports à accès direct**
    - ***Inutile de tout lire pour déchiffrer un secteur***

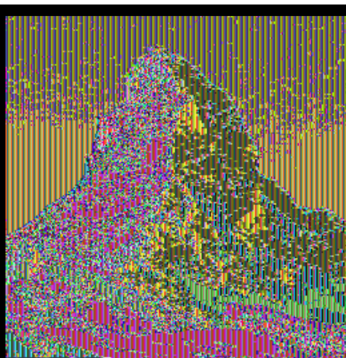
Masque[n] = DES_K(f(n))

CT[n] = PT[n] ⊕ Masque[n]

Tous ces modes de chaînage sont valables pour tous les algorithmes de chiffrement par blocs

Révisions





Chiffrement d'une image en DES ou AES

En mode EBC

Avec chaînage

© Wikipedia

Structures algébriques classiques : Les Groupes

JLS CONSEIL

- **Groupes**

  - **(Z/NZ , +) : N éléments**
  - **(U_N , ×) : Φ(N) éléments**

  - **Loi de groupe sur l'ensemble des points d'un plan R²**
  - **Loi de groupe ° sur l'ensemble des points d'une courbe elliptique**
  - **Groupe des diviseurs d'une courbe algébrique**
  - **Groupe de Jacobi d'une courbe algébrique**

Structures algébriques classiques : Les Groupes

JLS CONSEIL

- **Changement d'élément neutre dans un groupe commutatif G**
  - **A partir de la loi (G,o) et d'un élément quelconque ω∈ G, on peut construire un nouveau groupe (G,o_ω) ayant pour élément neutre ω**
    - **(A,B) ⟶ AoB o ω^{o-1}**
    - **(A o ω, B o ω) ⟶ A o B o ω**

  - **Exercice :** Faire la construction géométrique sur les vecteurs de R²
  - **Exercice :** Faire la construction géométrique sur une courbe elliptique
  - **Exercice :** Sur une courbe elliptique, montrer l'équivalence entre le groupe de Jacobi et la loi de groupe classique (avec un élément neutre ω quelconque)
  - **Exercice :** Sur une courbe elliptique, la somme de trois points alignés est nulle si l'élément neutre est un point d'inflexion

JLS CONSEIL

Structures algébriques classiques : Les Anneaux

- **Anneau :**
  - **Groupe commutatif** (loi par exemple notée +)
  - **Muni d'une seconde loi** (par exemple notée ×)

Exemple : Anneau Z des entiers relatifs

- **Corps**

Tout élément non nul est inversible

K* = K - {0} est un groupe pour la loi ×

JLS CONSEIL

Structures algébriques classiques : Espaces Vectoriels et Modules

- **Espace vectoriel sur un corps K**
  - **Groupe commutatif** (loi par exemple notée +)
  - **Multiplication par un scalaire**
- **Module sur un anneau A**
  - **Groupe commutatif** (loi par exemple notée +)
  - **Multiplication par un scalaire**

Un groupe commutatif peut toujours être considéré comme un Z-module

JLS CONSEIL

Structures algébriques classiques : Les Algèbres

- **Algèbre sur un corps commutatif K**
  - **Espace vectoriel sur K,**
  - **Muni d'une seconde loi** (par exemple notée $\times$) **qui lui confère une structure d'anneau**

- **Algèbre des polynômes sur un corps commutatif K**
  - **Noté K[X]**
  - **Degré d'un polynôme**

Le produit d'un polynôme de degré q et d'un polynôme de degré r est un polynôme de degré q.r

JLS CONSEIL

Structures algébriques classiques : Propriétés des Anneaux

- **Propriétés de l'anneau Z des entiers relatifs**
  - **Notion d'idéal**
  - **Division euclidienne**
  - **PGCD**
  - **Théorème de Bezout**
  - **Quotient d'un anneau par un idéal : Z/NZ**
    - **Si idéal maximal (ensemble des multiples d'un nombre N premier), alors le quotient est un corps)**

Démonstration :

$x \in$ Z/NZ inversible $\Leftrightarrow$ x premier à N

Si N est premier, tout x non multiple de N (donc non nul modulo N) est premier à N

Structures algébriques classiques : Division euclidienne

JLS CONSEIL

- **Exemple dans Z**

```
3 2 2 8 1 | 2 5
2 5 0 0 0 |--------
----      | 1 2 9 1
  7 2 8 1 |
  5 0 0 0 |
  ---     |
  2 2 8 1 |
  2 2 5 0 |
  -----   |
      3 1 |
      2 5 |
      ---  |
        6 |
```

```
32281 = 0     × 25   +  32281
32281 = 1000  × 25   +   7281

32281 = 1200  × 25   +   2281

32281 = 1290  × 25   +     31

32281 = 1291  × 25   +      6
```

$a = b\,q_i + r_i$
De proche en proche, on augmente q_i et on baisse r_i jusqu'à obtenir r vérifiant $0 \leq r < b$ et $a = b\,q + r$

Structures algébriques classiques : Division euclidienne

JLS CONSEIL

- **Exemple dans R[X]**

```
X⁴ + 2   X³  +  2   X²   +   5 X  +  3 |  2 X²  +  3 X   + 1
X⁴ + 1.5 X³  +  0.5 X²                 |-----------------------
-------------------------              |0.5 X² + 0.25 X + 0.375
     0.5 X³  +  1.5 X²   +   5 X       |
     0.5 X³  +  0.75 X²  +  0.25 X
     ------------------------------
                0.75 X²  +  4.75 X  + 3
                0.75 X²  +  1.125 X + 0.375
                ---------------------------
                            3.625 X + 2.625
```

$X^4 + 2X^3 + 2X^2 + 5X + 3 = (2X^2 + 3X + 1) \times (0.5X^2 + 0.25X + 0.375) + (3.625X + 2.625)$

$a[X] = b[X] \times q[X] + r[X]$ avec $\text{Degré}(r[X]) < \text{Degré}(b[X])$

Méthodologie : $a[X] = b[X] \times q_i[X] + r_i[X]$.
On part de $q_0[X]=0$ et $r_0[X]=a[X]$
de proche en proche, on tue les termes de degré élevé dans $r_i[X]$ jusqu'à obtenir un degré inférieur à celui de b[X]

JLS CONSEIL

Structures algébriques classiques : Division euclidienne dans K[X]

- **Unicité du résultat**

$a[X] = b[X] \times q_1[X] + r_1[X]$ avec $Degré(r_1[X]) < Degré(b[X])$

$a[X] = b[X] \times q_2[X] + r_2[X]$ avec $Degré(r_2[X]) < Degré(b[X])$

$0 = b[X] \times (q_1[X] - q_2[X]) + (r_1[X] - r_2[X])$

Si $(q_1[X]-q_2[X]) \neq 0$ alors $Degré(b[X] \times (q_1[X]-q_2[X])) > Degré(b[X])$

Mais $Degré(r_1[X]-r_2[X]) < Degré(b[X])$ d'où contradiction

JLS CONSEIL

Structures algébriques classiques : Idéal dans K[X]

$$\left.\begin{array}{l} a[X] \in \mathfrak{I} \\ b[X] \in \mathfrak{I} \\ c[X] \in K[X] \end{array}\right\} \Rightarrow \begin{array}{l} a[X] + b[X] \in \mathfrak{I} \\ c[X] \times a[X] \in \mathfrak{I} \end{array}$$

Dans K[X], tout idéal est principal, c'est-à-dire égal à l'ensemble des multiples d'un polynôme d[X]

Soit d[X] un élément de degré minimal dans $\mathfrak{I}$
Pour tout $a[X] \in \mathfrak{I}$ on peut faire une division euclidienne
$a[X] = d[X]q[X] + r[X]$ avec $Degré(r[X]) < Degré(d[X])$ et $r[X] \in \mathfrak{I}$ donc $r[X] = 0$
Donc a[X] est multiple de d[X]

Si $degré(a[X]) = degré(d[X])$ alors $a[X] = \lambda\, d[X]$ avec $\lambda \in K$

Le polynôme d[X] est défini au produit près par un $\lambda \in K$ non nul

Structures algébriques classiques : Notion de pgcd dans $K[X]$

JLS CONSEIL

Etant donnés $a[X], b[X] \in K[X]$ ils engendrent l'idéal

$$\mathcal{I}(a[X],b[X])=\{a[X]u[X]+b[X]v[X] ; \forall u[X],v[X]\in K[X]\}$$

Il existe un polynôme $d[X]$ est défini au produit près par un $\lambda\in K$ non nul tel que $\mathcal{I}$ soit l'ensemble des multiples de $d[X]$.

$d[X]$ est le pgcd de $a[X]$ et $b[X]$

On a le théorème de Bezout :

$d[X]$ est le pgcd de $a[X]$ et de $b[X]$ si et seulement si

$\exists\ p[X],q[X]\in K[X]$ tels que $d[X]= p[X]a[X]+ q[X]b[X]$

Structures algébriques classiques : Notion de pgcd dans $K[X]$

JLS CONSEIL

Définition

$c[X]\in K[X]$ est irréductible

s'il n'a pas de diviseur autre que lui-même (à une constante multiplicative près) ou les polynômes constants

Théorème

$a[X], b[X] \in K[X]$ sont premiers entre eux

(n'ont aucun diviseur commun autre que les polynômes constants)

Si et seulement si

$\exists\ p[X],q[X]\in K[X]$ tels que $p[X]a[X] + q[X]b[X] = 1$

Si $c[X]$ est irréductible,

il est premier à tout $a[X]\in K[X]$ non multiple de $c[X]$

Irréductible $\cong$ Premier

Structures algébriques classiques : Quotient de K[X] par un idéal

JLS CONSEIL

Soit <m> l'idéal formé par l'ensemble des multiples de m[X]∈K[X]

On considère dans K[X] la relation d'équivalence

a[X] ~ b[X] ⇔ a[X] - b[X] ∈ <m>

L'anneau quotient est noté K[X]/<m>

m[X] est défini à une constante ∈K multiplicative près.

S'il est de degré n, on peut toujours supposer qu'il s'écrit

m[X] = X^n – p[X] (avec Degré(p[x])< n)

Donc X^n~p[X] et tout a[X]∈K[X] est à un polynôme de degré<n

K[X]/<m> est un espace vectoriel de dimension n

Structures algébriques classiques : Eléments inversibles dans K[X]/<m>

JLS CONSEIL

b[X] est l'inverse de a[X] si et seulement si a[X].b[X] ~ 1

donc a[X].b[X] - 1 est un multiple de m[X]∈K[X]

a[X].b[X] + u[X].m[X] = 1

On peut trouver b[X] et u[X]si et seulement si a[X] est premier à m[X] (Bezout)

a[X] est inversible dans K[X]/<m>
si et seulement s'il est premier à m[X]

K[X]/<m> est un corps
si et seulement si m[X] est irréductible

Surcorps de K et Espace vectoriel de dimension n sur K

Structures algébriques classiques : Diviseurs de zéro dans un corps

JLS CONSEIL

Dans un corps, il n'y a pas de diviseur de zéro autre que zéro

$a.b = 0 \Rightarrow a^{-1}.a.b = 0 \Rightarrow 1.b = 0 \Rightarrow b = 0$

Si `m[X]= a[X].b[X]` alors `a[X]` est diviseur de zéro dans `K[X]/<m>`

`K[X]/<m>` est un corps
si et seulement si `m[X]` est irréductible

Structures algébriques classiques : Quotient de `K[X]` par un idéal

JLS CONSEIL

Exemple :

`K` est le corps des réels R et `m[X]= X²+1` (irréductible sur R)

`R[X]/X²+1`

est un surcorps de R et un espace vectoriel de dimension 2 sur R

Il est isomorphe au corps C des nombres complexes :

$a+bX \cong a+bi$

Structures algébriques classiques : Les Algèbres de polynômes

JLS CONSEIL

- **K[X]** **K[X]/<P[X]>**
- **Comparaison entre l'arithmétique en grands nombres et l'arithmétique dans K[X]**
  - Le report des retenues
  - Homomorphisme K[X] $\longrightarrow$ K

Cas particulier : K = Z/2Z = {0,1}

- Un élément de K est un bit
- Un polynôme de degré q est une suite de q bits
  - Addition = XOR bit à bit, noté $\oplus$
  - Multiplication par un scalaire évidente (0 ou Id)
  - Multiplication de polynômes noté $\otimes$

Structures algébriques classiques : Polynômes sur Z/2Z

JLS CONSEIL

$a[X] = \sum a_i X^i$ $\quad$ $a \oplus b[X] = \sum (a_i \text{ XOR } b_i)X^i$

$b[X] = \sum b_i X^i$

$c[X] = \sum c_i X^i$

$c[X] = a[X] \otimes b[X]$ donné par $c_i = \text{XOR}_{j+k=i} (a_j \text{ AND } b_k)$

- **Multiplier par X^e est un simple décalage de e bits vers la gauche**

Notation : L'octet {6B} représente le polynôme $X^6+X^5+X^3+X+1$

JLS CONSEIL

Structures algébriques classiques : Corps de Galois à 2^e éléments

- **On choisit une fois pour toutes un pôlynome m[X] de degré e et on travaille dans $Z/2Z\,[X] \;/\; <m>$**
  - ❖ **$m[X] = X^e + \sum_{0 \le i \le e-1} a_i X^i$**
  - ❖ **$X^e \equiv \sum_{0 \le i \le e-1} a_i X^i$**

- **Si m[X] est irréductible, tout polynôme de degré inférieur à e est inversible modulo m (Bezout)**

$$GF(2^e) = Z/2Z\,[X] \;/\; <m>$$

Tous ces corps sont isomorphes entre eux

JLS CONSEIL

Structures algébriques en AES : Corps de Galois à 2^8 éléments

- **Pour AES, le polynôme `m[X]` est normalisé**

  `m[X] = X⁸ + X⁴ + X³ + X + 1`
- **On note • la multiplication modulo `m`**

- **En AES un octet représente un élément du corps de Galois $GF(2^8) = Z/2Z\,[X] \;/\; < X^8 + X^4 + X^3 + X + 1 >$ muni des opérations**
  - **⊕ c'est-à-dire XOR bit à bit**
  - **• c'est-à-dire multiplication modulo m[X]**

Exemple : {57} •{ 83} = {c1}

$(X^6+X^4+X^2+X+1)(X^7+X+1) = X^7+X^6+1$

JLS CONSEIL

Exemple : {57}•{83} = {c1} dans Z/2Z [X] / <m>

$(X^6+X^4+X^2+X+1) \bullet (X^7+X+1) = X^7+X^6+1$

```
            0 1 0 1   0 1 1 1     {57}
            1 0 0 0   0 0 1 1     {83}
        -----------------------
            0 1 0 1   0 1 1 1
          0 1 0 1 0   1 1 1
0 1 0   1 0 1 1   1
---------------------------------
0 1 0   1 0 1 1   0 1 1 1   1 0 0 1   {2B}{F9}    = {57}•{83} dans Z/2Z [X]
  1 0   0 0 1 1   0 1 1                   +     m × X^5 = 0  tue le terme X^13
---------------------------------
        1 0 0 0   0 0 0 1   1 0 0 1
        1 0 0 0   1 1 0 1   1             +     m × X^3 = 0  tue le terme X^11
---------------------------------
                  1 1 0 0   0 0 0 1     {c1}
```

Pour une implémentation informatique, on préférera précalculer tout et stocker en mémoire la table de multiplication de • (soit 64 Koctets)

JLS CONSEIL

Structures algébriques en AES : Polynômes sur GF(2^8)

- **Chaque coefficient est un octet c'est-à-dire un élément de GF(2^8)**
- **Les calculs sont faits avec les opérations**
  - ⊕ **c'est-à-dire XOR bit à bit**
  - • **c'est-à-dire multiplication modulo m**
- **En AES on travaille modulo X^4+1**
  - ➢ **Attention : ce polynôme n'est pas irréductible, le quotient est un anneau mais n'est pas un corps.**
  - ➢ **$X^4+1 = (X^2+1)^2 = (X+1)^4$**
  - ➢ **$X^i \equiv X^{i \bmod 4}$**
- **En AES un mot de 32 bits = 4 octets représente un élément de l'anneau GF(2^8)/<X^4+1>**

a[X] = {03}X^3 + {01}X^2 + {01}X + {02} est inversible (premier à X^4+1)
Le calcul de l'inverse est difficile et nécessite un algorithme similaire à Euclide étendu
a[X]$^{-1}$ = {0b}X^3 + {0d}X^2 + {09}X + {0e}
La vérification est facile et est laissée en exercice

Structures algébriques en AES : Opérations de base sur les mots

JLS CONSEIL

Rappel : on travaille sur GF(2^8)`[X] / <X⁴+1>`

avec GF(2^8) = Z/2Z `[X] / < X⁸ + X⁴ + X³ + X + 1 >`

- **`RotWord()` :** multiplication par X^3 = permutation circulaire sur les octets
- **`MixColumns()` :** multiplication par le polynôme $\{03\}X^3 + \{01\}X^2 + \{01\}X + \{02\}$

Ces opérations sont inversibles

<u>Exercice laissé au lecteur</u> : quelles sont les opérations inverses ?

Structures algébriques en AES : Opérations de base sur les mots

JLS CONSEIL

Dans l'espace vectoriel (GF(2^8)[X])^4 `MixColumns()` revient à faire un produit matriciel (opérations ⊕ et •) par la matrice

```
{02} {03} {01} {01}

{01} {02} {03} {01}

{01} {01} {02} {03}

{03} {01} {01} {02}
```

Structures algébriques en AES : La fonction `SubBytes()`

JLS CONSEIL

On travaille sur des octets représentant des éléments du corps $GF(2^8)$

$b \in GF(2^8)$

Etape 1 : On calcule l'inverse de b dans ce groupe (0 si b=0)

Etape 2 : Fonction affine sur les bits de b, la constante additive étant c={63}

$b'_i = b_i \oplus b_{(i+4) \bmod 8} \oplus b_{(i+5) \bmod 8} \oplus b_{(i+6) \bmod 8} \oplus b_{(i+7) \bmod 8} \oplus c_i$

Revient à faire dans $(Z/2Z)^8$ un produit matriciel par

$$\begin{matrix} 1&0&0&0&1&1&1&1 \\ 1&1&0&0&0&1&1&1 \\ 1&1&1&0&0&0&1&1 \\ 1&1&1&1&0&0&0&1 \\ 1&1&1&1&1&0&0&0 \\ 0&1&1&1&1&1&0&0 \\ 0&0&1&1&1&1&1&0 \\ 0&0&0&1&1&1&1&1 \end{matrix}$$

Suivi d'une addition avec `0 1 1 0  0 0 1 1`

Cette fonction est inversible et peut se tabuler dans une S-box

Principes de l'AES

JLS CONSEIL

- Appel d'offres NIST
- Lauréat : Rijndael Université de Leuven
  - Vincent Rijmen
  - Joan Daemen
- Objectifs
  - Portabilité (processeurs 32 bits, cartes à puce à processeur 8 bits…)
  - Aussi sûr que Triple DES mais beaucoup plus rapide
  - Compromis sécurité-performance
- Caractéristiques
  - Blocs de 128 bits = 16 octets = 4 mots de 32 bits
  - Nombre de rondes : N_r = 10, 12 ou 14
  - Clés à 128, 192 ou 256 bits

JLS CONSEIL

Principes de l'AES

- **Le calcul des clés de ronde :**

Algorithme de cadencement (scheduling)

Part d'une clé originelle de N_k = 4, 6, 8 mots de 32 bits

Génère N_r+1 clés de 4 mots de 32 bits

Routines mises en œuvre

- `SubWord()` applique `SubBytes()` octet par octet (substitution par S-box)
- `RotWord()` permutation circulaire sur les octets
- `Rcon[i]` constante = $[x^{i-1}, \{00\}, \{00\}, \{00\}]$ avec $x=\{02\} \in GF(2^8)$

w : Tableau résultat (N_r+1 fois 4 mots).

Les N_k premiers mots (indices entre 0 et N_k-1) sont remplis par la clé originelle

Puis pour i variant de N_k à $4(N_r+1)-1$

```
temp:=w[i-1]
Si i divisible par Nk  temp := SubWord(RotWord(Temp)) XOR Rcon[i/Nk]
Sinon Si Nk>6 et (i Mod Nk) = 4   temp := SubWord(Temp)
w[i]:=w[i-Nk] XOR temp
```

Chaque mot dépend du précédent et de celui N_k positions avant

Plus complexe pour clés à 256 bits (version renforcée)

JLS CONSEIL

Principes de l'AES

- **Détail des rondes sur un bloc (4 fois 4 octets)**

XOR avec la clé de ronde 0 (Au départ 128 premiers bits de la clé originelle)

Boucle pour i variant de 1 à N_r

`SubBytes` (Substitution, agit indépendamment sur chaque octet)

`ShiftRow` Permutation circulaire sur les lignes de 0,1,2,3 crans vers la gauche

Si i <N_r `MixColumns` Produit matriciel dans l'espace vectoriel $(GF(2^8)[X])^4$ (Agit indépendamment sur chaque colonne)

XOR avec la clé de ronde i

Fin de boucle Il faut donc au total N_r+1 clés de ronde

JLS CONSEIL

Principes de l’AES^{-1}

Toutes les opérations élémentaires sont inversibles

Il suffit de les prendre dans l’ordre inverse

N.B.: l’ AES^{-1} est inutile en mode CTR

JLS CONSEIL

Questions?