République Algérienne Démocratique et Populaire

Ministère de l''Enseignement Supérieur et de la Recherche Scientifique

Université de MENTOURIE Constantine
Faculté des Sciences
Département des Mathématiques

N° d'ordre :
Série :

Mémoire

Présenté pour obtenir le diplôme
de Magister en Mathématiques

THEME

option: systèmes dynamiques et topologie algébrique

Par

Mr : ADOUI Salah

Soutenue le : ***devant le jury composé de :***

Président	***: Mr. BENKAFADAR M.N***	***Prof***	***Université Mentourie Constantine.***
Rapporteur	***: Mr. NOUI Lemnouar***	***Prof***	***Université de Batna.***
Examinateur :	***Mr. BOUKAROURA Ali***	***M.C***	***Centre. Universitaire de Mila***
Examinateur :	***Mr. ZAÏMI Toufik***	***M.C***	***Université Larbi Ben Mhidi OEB.***

Outils algébriques et cryptosystèmes

DEDICACE:

Je dédie ce modeste travail à l'esprit de mon père KHELIFA, à ma chère mère FATMA, à ma petite famille (AMANI, MERIEM et KHALIL), ainsi que mes sœurs, frères et tous mes proches, en particulier la famille AKSA. J'espère la réussite que je vais accueillir soit une petite récompense et un remerciement infini pour eux tous et pour tous ceux qui m'ont aidé de proche ou de loin pour que ce travail puisse voir la lumière.

Remerciements:

Nous remercions tout d'abord et avant tout le tout puissant ALLAH qui nous a réussi à achever ce travail.

Je tiens à remercier Mr. NOUI Lemnouar professeur à l'université de Batna qui a accepté de bon cœur et de bienveillance, de diriger mon travail, et de me suivre patiemment dans toutes les étapes de cette étude. Je lui suis reconnaissant de ses remarques nombreuses, sa gentillesse et sa patience.

Mes remerciements vont également à Monsieur BENKAFADAR Mohamed Nacer professeur à l'Université de Constantine d'avoir accepté de présider le jury.

De même, je remercie Monsieur BOUKAROURA Ali maître de conférence au centre Universitaire de Mila et Monsieur ZAÏMI Toufik maître de conférence à l'Université d'Oum Elbouaghi; d'avoir d'accepté d'examiner et juger ce travail.

Merci au Dr. RAHMANI le chef du département et à toute l'équipe de département de Mathématiques de l'Université de Constantine.

Merci à mes amis BENSALAH A. Aldjabar maître de conférence à l'université de Batna, MENSOURI Kamel et BENBRAHIM A. Elouahab chargés de cours aux Centres Universitaire de M'Sila et Oum Bouaghi de leurs conseils.

Enfin merci à tous ceux que j'ai oublié de remercier.

TABLE DES MATIERES

Vocabulaire:

- **Chiffrer**: transcrire, à l'aide d'un algorithme un message clair en une suite incompréhensible de symboles.
- **texte en clair**: le message à chiffrer.
- **texte chiffré**: le résultat du chiffrement.
- **Déchiffrer**: retrouver le texte en clair à partir du texte chiffré à l'aide d'un algorithme paramétrable.
- **clé**: le paramètre des algorithmes de chiffrement et de déchiffrement.
- **Décrypter**: retrouver le texte en clair à partir du texte chiffré sans la clé.
- **Cryptographie**: science du chiffrement.
- **Cryptanalyse**: science du décryptage.
- **Cryptologie**: cryptographie et cryptanalyse.
- **Cryptosystème**: ensemble des méthodes de chiffrement et de déchiffrement utilisables en sécurité.

Ø Introduction générale

La moitié de ce siècle est celle de la révolution numérique et de l'utilisation systématique de l'algèbre dans la transmission de données.

L'information est précieuse lors de son stockage ou sa transmission, il est nécessaire de la protéger. Deux grands types de protection se distinguent:
La protection contre les ennemis malicieux et la protection contre les altérations dues à des problèmes physiques (canaux bruitées). La théorie des codes correcteurs d'erreurs et la cryptologie sont les domaines de recherche associés à ces problématiques.

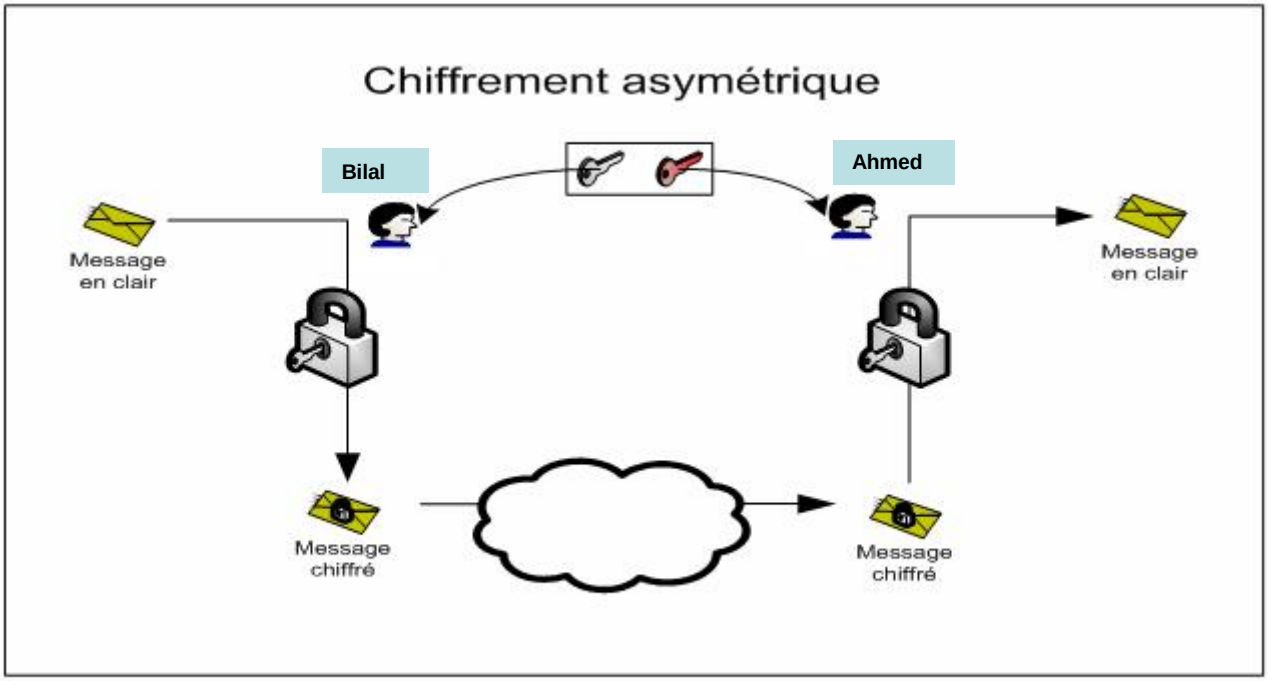
La cryptologie, qui nous intéresse plus particulièrement ici, se divise en deux disciplines complémentaires et indissociables: la cryptographie: la conception de systèmes de protection et la cryptanalyse: l´étude des attaques des systèmes connus.

Dans ce mémoire nous nous intéressons à la conception de quelque cryptosystèmes et l'étude de leurs sécurité élémentaire. Les systèmes ordinaires, les systèmes à base de courbes algébriques. Dans ce cadre la sécurité est liée au problème du logarithme discret dans des groupes ordinaires ou des groupes liés aux courbes.
On va s'intéresser aussi aux cryptosystèmes basés sur les codes correcteurs d'erreurs .

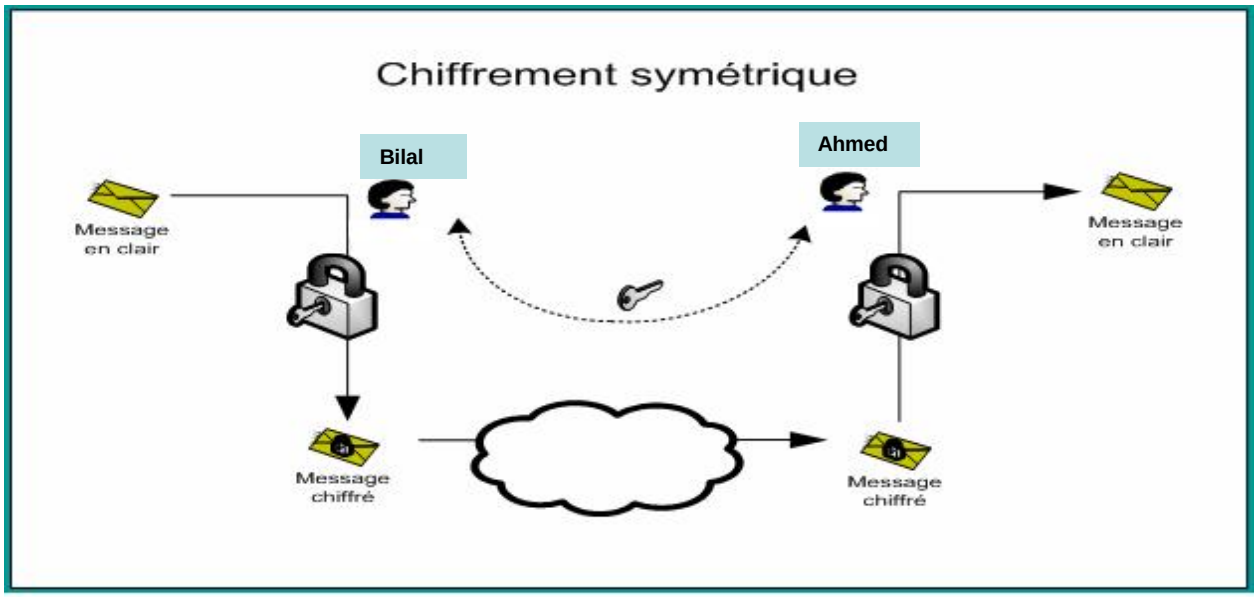
Cryptosystèmes:

Commençons par donner une idée des différents types de cryptosystèmes. Il en existe principalement deux types:

- Les systèmes à clé publique ou cryptosystèmes asymétriques: la clé pour coder le message est connue de tout le monde mais ne permet pas d'en déduire la clé qui permet de décrypter le message. Cette clé-ci n'est connue que par le destinataire. Par exemple le RSA, AlGamel sont des cryptosystèmes asymétriques.



- Les systèmes à clé privée ou cryptosystèmes symétriques: dans ce cas les correspondants se mettent d'accord sur une clé secrète que seul eux connaissent. Il leur faut alors un moyen sûr pour s'échanger la clé. Par exemple le protocole de Diffie-Hellmann, que nous allons présenter, permet d´échanger une clé en toute sécurité.



… Les systèmes à clé publique ou cryptosystèmes asymétriques sont les plus utilisés en pratique dans la vie courante ; dans Les distributeurs de billets, Les téléphones mobiles, Les cartes à puces, Le commerce électronique, Les numéros de séries, etc...

Chapitre 1:

Notions fondamentales

d'algèbre

Chapitre 1: Notions fondamentales **d'algèbre.**

Dans ce chapitre on va acquérir les notions fondamentales utilisées dans les cryptosystèmes, à partir des notions de groupes, corps, matrices, applications linéaires, courbes algébriques planes….

1-1- Structures algébriques fondamentales:

1-1-a-Groupes, Groupes cycliques:

Définition.1**:** On appelle groupe un couple $(G, *)$ où G est un ensemble non vide et $*$ est une loi de composition interne associative, possédant un élément neutre **e** et pour laquelle tout élément de G admet un symétrique.

Si de plus $*$ est commutative, on dit que G est un groupe commutatif ou abélien.

Notation**:** Les notations les plus utilisées sont: la notation additive avec 0 pour élément neutre, $-x$ pour le symétrique (dit aussi opposé) et la notation multiplicative où l'élément neutre est noté 1, x^{-1} pour le symétrique (dit aussi inverse).

Exemples:

- Z, R, Q: sont des groupes additives commutatifs.
- $(Z/pZ)^*$ est un groupe multiplicatif commutatif.

avec $(Z/pZ)^* = \{1, 2, 3, \ldots, (p-1)\}$, p: premier.

Définition 2 : Soit G un groupe fini. L'ordre d'un groupe est son cardinal et on le note |G| ou O(G) ou card (G).

Définition 3 **:** Soit G un groupe fini et soit g un élément de G. L'ordre de g est l'ordre du groupe engendré par g.

Définition 4**:** On dit qu'un groupe G est monogéne s'il est engendré par un élément g et qu'il est cyclique si en plus il est fini.

$G=<g>=\{g^0, g^1, g^2,\ldots\}$ ⟶ monogéne.

$G=<g>=\{g^0, g^1, g^2,\ldots, g^p\}$ ⟶ cyclique.

Exemples:

((Z/7Z)*,.) est un groupe cyclique.

card(G)=6; card (Z/7Z)*= {1,2,3,4,5,6}.

$G=<3>=\{3^0, 3^1, 3^2,3^3,3^4,3^5\}=\{1,3,2,6,4,5\}$;

3 est un générateur du groupe ((Z/7Z)*,**.**)

Théorème 5: [1]

L'ordre d'un élément d'un groupe fini G divise l'ordre de G.

Morphisme de groupes:

Définition 6**:** Soient $(G, *)$, $(H, \top)$ deux groupes et f une application de G dans H. On dit que f est un morphisme de groupes, si:

Ø $\forall a, b \in G;\ f(a * b) = f(a) \top f(b)$.

Si en plus f est bijective, on dit que f est un isomorphisme de plus G = H, on parle d'endomorphisme et d'automorphisme.

1-1-b- Anneaux, corps, corps finis:

Définition 7**:** Un anneau est un triplet (A,+, .) où A est un ensemble et **+** et **.** deux lois de composition interne tels que:

(A,+) soit un groupe commutatif, la loi **.** une loi associative, possédant un élément neutre noté 1 et distributive par rapport à la loi +.

Si de plus la loi **.** est commutative, on parle d'anneau commutatif.

Définition 8**:** K= (A,+, .) est un corps ssi:

- ü K= (A,+, .) est un anneau.
- üChaque élément non nul de A possède un inverse pour l'opération (**.**).

q Si a, b∈A, alors: a(-b)=(-a)b=-ab et a.0=0.a=0. [6].

Définition 9:

Un corps fini est un corps qui possède un nombre fini q d'éléments, et on le note par F_q .

Proposition 10 **:** [1]

Tout anneau intègre fini est un corps.

Définition 11**:** Soient G, H deux anneaux et f une application de G dans H. On dit que f est un morphisme d'anneaux, si:

- Ø∀a, b ∈ G; f (a **+** b) = f (a) **+** f (b) et f (a **.** b) = f (a) **.** f (b) .

Théorème 12 : [1]

Deux corps finis ayant le même nombre d'éléments sont isomorphes.

Propriété d'un corps fini: [1]

- ØTout corps fini est commutatif.
- ØLa cardinalité d'un corps fini F_q est une puissance d'un nombre premier c-à-d: $q=p^m$ avec p est premier et m>0.

Exemples de corps finis:

Exemple 1**:** Corps binaire F_2 .

F_2={0,1} est le corps binaire; c'est le plus petit corps fini.

Exemple 2: Corps fini F_8.

F_8 ($8=2^3$;p=2et m=3). F_8 est de caractéristique 2, donc F_8 est construit comme extension du corps premier Z/2Z .

-Soit le polynôme: $f(x)=x^3+x+1$; ce polynôme est irréductible sur F_2, f(x) est une cubique qui n'a pas de zéro dans F_2 , car f(0)=1 et f(1)=1.

Comme f(x) est de degré 3 sur F_2, les éléments de F_8 sont de la forme: $ax^2+ bx+c$ avec

a, b, c des éléments de F_2. Donc: $x^3+x+1=0 \mod(x^3+x+1)$;

Alors : $x^3=x+1 \mod (x^3+x+1)$;

Et par conséquent : $x^4=x^2+x \mod (x^3+x+1)$;

Calculons les puissances successives de x :

$x^0=1$

$x^1=x$

$x^2=x^2$

$x^3=x+1$

$x^4=x^2+x$

$x^5=x^3+x^2=x^2+x+1$

$x^6=x^3+x^2+x=x^2+1$

$x^7=x^3+x=1$

Donc: $F_8= \{0,1,x, x^2, x+1, x^2+x, x^2+x, x^2+x+1\}$.

Exemple 3**:**

corps fini F_9.

F_9 ($9=3^2$; p=3 et m=2). F_9 est de caractéristique 3, donc F_9 est construit comme extension du corps premier Z/3Z.

-Soit le polynôme: $P(x)=x^2+1$; ce polynôme est irréductible sur F_3; P(x) n'a pas de zéro dans F_3, car: P(0)=1, P(1)=2 et P(2)=2 .

Comme P(x) est de degré 2, les éléments de F_9 sont de la forme: ax+ b avec a, b ∈ F_3

donc: $x^2+1=0 \mod(x^2+1)$;

alors $x^2=2 \mod (x^2+1)$;

et par conséquent : $x^3=2x \mod (x^2+1)$; $x^4=1$…

Alors F_9= {0,1, 2, x, 2x, x+1, 2x+1, x+2, 2x+2}.

1-1-c- Espaces vectoriels, Applications linéaires et Matrices:

Définition 12: Soit K un corps commutatif.

Un ensemble E est un espace vectoriel sur K ssi:

1- il existe une opération dans E qui rend E groupe abélien

(cette opération sera noté additivement).

2- il existe une application: $K \times E \longrightarrow E$

$(\alpha, x) \longrightarrow \alpha.x$; tel que:

- Pour tout α, β ∈ K et x ∈ E: (α+β).x= αx+βx.
- Pour tout α ∈ K et y, x ∈ E: α(x+y)= αx+ αy.
- Pour tout α, β ∈ K, x ∈ E: α(βx)= (αβ)x.
- Pour tout x ∈ E: 1.x=x (l'élément neutre de **.** dans K).
- Les éléments de K sont appelés: des scalaires .
- Les éléments de E sont appelés: des vecteurs .

Remarque:

Les règles de calculs dans un espace vectoriel E sur un corps K sont analogues à celle sur les vecteurs libres de la géométrie ordinaire.

Exemples d'espaces vectoriels:

Exemple 1: Soit l'ensemble R muni de l'addition et et la multuplication usuelles.

R est un corps et R^n est un espace vectoriel sur R, pour tout entier n >0.

Exemple 2: Soit K un corps commutatif, on peut toujours considérer K^n comme espace vectoriel sur le corps K. En particulier si $K=F_q$ donc:

$(F_q)^n$ est un espace vectoriel sur le corps fini F_q.

Définition 13: Soit E un espace vectoriel sur K. Une partie non vide F de E est un sous espace vectoriel de E ssi:

- Pour tout x, y $\in$ F: x-y $\in$ F.
- Pour tout $\alpha \in K$ et $x \in F$: $\alpha.x \in F$.

Définition 14: Soient $x_1,..., x_p$ des vecteurs de E (espace vectoriel sur K).

Ces p vecteurs sont dits linéairement indépendants ssi:

$\alpha_1x_1+\ldots+ \alpha_p x_p=0$ implique $\alpha_1=\ldots= \alpha_p=0$, avec les $\alpha_i \in K$.

Définition 15: Les vecteurs $x_1,\ldots, x_n$ de E forment une base de E si et seulement si:

- $x_1,\ldots, x_n$ sont linéairement indépendants.
- Pour tout x de E, ils existent $\alpha_1,.., \alpha_n$ de K tel que:

$$x = \alpha_1x_1+\ldots+\alpha_n x_n .$$

Exemple: R^n est un espace vectoriel sur R.

$e_1=(1,0,0,\ldots,0)$, $e_2=(0,1,0,...,0)$,..., $e_n=(0,0,...,0,1)$

est une base de R^n (base canonique).

Pour tout a $\in R^n$ avec $a=(a_1,a_2,...,a_n)$ on a:

$a=a_1e_1+a_2e_2+\ldots+a_n e_n$.

Définition 16: E est dit de dimension finie s'il admet une base finie.

Définition 17: Applications linéaires.

Soient E, F deux espaces vectoriels sur le même corps K.

f : E ⟶ F est une Application linéaire ssi:

- Pour tout x, y ∈ E: f(x+y)=f(x)+ f(y).
- pour tout α ∈K et x ∈K, f(α.x)= α f(x).

Définition 18:

Soit f : E ⟶ F une application linéaire. E, F de dimension

finies n, p (respectivement) de N*, et de base $B_E=(u_1,u_2,\ldots,u_n)$ et $B_F=(v_1,v_2,\ldots,v_p)$.

On appelle Matrice de f dans les bases B_E et B_F le tableau constituant de « n » colonnes et « p » lignes, chaque colonne « j » est constituée par les cordonnées du vecteur $f(u_j)$ dans la base B_F pour $1 \leq j \leq n$.

Notation:

On note par $M_{p,n}$ (K) par l'ensemble de matrices de n colonnes et p lignes.

$$M_{p,n}(K)=\begin{matrix}\begin{bmatrix} a_{11} & a_{12}\ldots\ldots\ldots\ldots\ldots & a_{1n} \\ a_{21} & a_{22}\ldots\ldots\ldots\ldots\ldots & a_{2n} \\ \ldots & \ldots & \ldots \\ \ldots & \ldots & \ldots \\ a_{p1} & a_{p2}\ldots\ldots\ldots\ldots\ldots & a_{pn} \end{bmatrix} \begin{matrix} v_1 \\ v_2 \\ : \\ : \\ v_p \end{matrix} \\ f(u_1) \quad f(u_2)\ldots\ldots\ldots\ldots\ldots\ldots \quad f(u_n)\end{matrix}$$

*Si E=F alors $M_{p,n}$ (K) l'ensemble de matrices carrés d'ordre n.

Exemple: pour n=3 et p=2:

$$\begin{cases} f(u_1)= 2v_1+v_2 \\ f(u_2)= -v_1+3v_2 \\ f(u_3)= v_1+2v_2 \end{cases} \text{ implique : } \quad M_f = \begin{bmatrix} 2 & -1 & 1 \\ 1 & 3 & 2 \end{bmatrix}$$

Définition 19 :

Une matrice carrée est une matrice dont le nombre de lignes est égal au nombre de colonnes. Ce nombre s'appelle l'ordre de la matrice.

Nous noterons $M_n(K)$ l'ensemble des matrices carrée d'ordre n à coefficients dans K.

Nous savons que $M_n(K)$ est un espace vectoriel sur K. De plus le produit de deux matrices carrée d'ordre n est toujours défini et c'est une matrice carrée d'ordre n .

Théorème 20: [1]

$M_n(K)$ est un anneau. Cet anneau n'est pas commutatif. [1]

Définition 21:

Soit $A \in M_n(K)$. S'il existe une matrice $B \in M_n(K)$ telle que :

$A.B = I_n$ (1); la matrice A est inversible.

Proposition 22: [1]

$A \in M_n(K)$ est inversible si est seulement si l'application linéaire associée à la matrice, Est bijective.

Proposition 23: [1]

A est inversible si est seulement si les vecteurs-colonnes de A sont linéairement Indépendants.

Définition 24 :

Soit E un espace vectoriel sur un corps K, de dimension n. Soit $B=(e_i)$ et $B'=(e_i')$ deux bases de E, i=1,…,n.

La matrice P dont la j-ème colonne est formée des composantes de e_j' dans la base B est appelée la matrice de passage de la base B à la base B'.

Théorème 25: [1]

La matrice de passage P est inversible. [1].

---

(1) I_n la matrice d'identité avec des 1 sur la diagonale et des 0 ailleurs.

Matrices équivalentes:

Définition 26 :

Soient A, B deux matrices de $M_{q,p}(K)$, s'il existe une matrice carrée P d'ordre p, inversible et une matrice carrée S d'ordre q inversible, telle que $B=S^{-1}.A.P$, on dit que la matrice B est équivalente à La matrice A.

Proposition 27: [1]

''La matrice B est dite équivalente à La matrice A'': cette relation est une relation d'équivalence dans $M_{q,p}(K)$.

Transposé d'une matrice :

Définition 28:

La transposé d'une matrice A, notée A^t est la matrice dont les lignes sont les colonnes de A.

Propriétés générales:

Soient $A, B \in M_n(K)$.

1) $(A.B)^{-1}=B^{-1}.A^{-1}$, si A et B sont inversibles.

2) $(A.B)^t=B^t.A^t$

3) $(A+B)^t=A^t+B^t$.

4) Si A est inversible: $(A^t)^{-1}=(A^{-1})^t$.

Définition 29:

Rang d'une matrice A noté rg(A) est le nombre maximum des colonnes linéairement indépendantes.

Théorème 30: [8]

Soit $A \in M_{n,p}(K)$. Le rang de A est le plus grand rang des sous-matrices carées inversible de A.

1-2- Courbes elliptiques :

Résume:

Cette partie définira ce qu'est une courbe elliptique et le groupe topologique d'une telle courbe il sera également montré qu'une courbe elliptique peut s'écrire sous une forme particulière appelée équations de: Weierstrass.

1-2-a-Définition d'une Courbe elliptique:

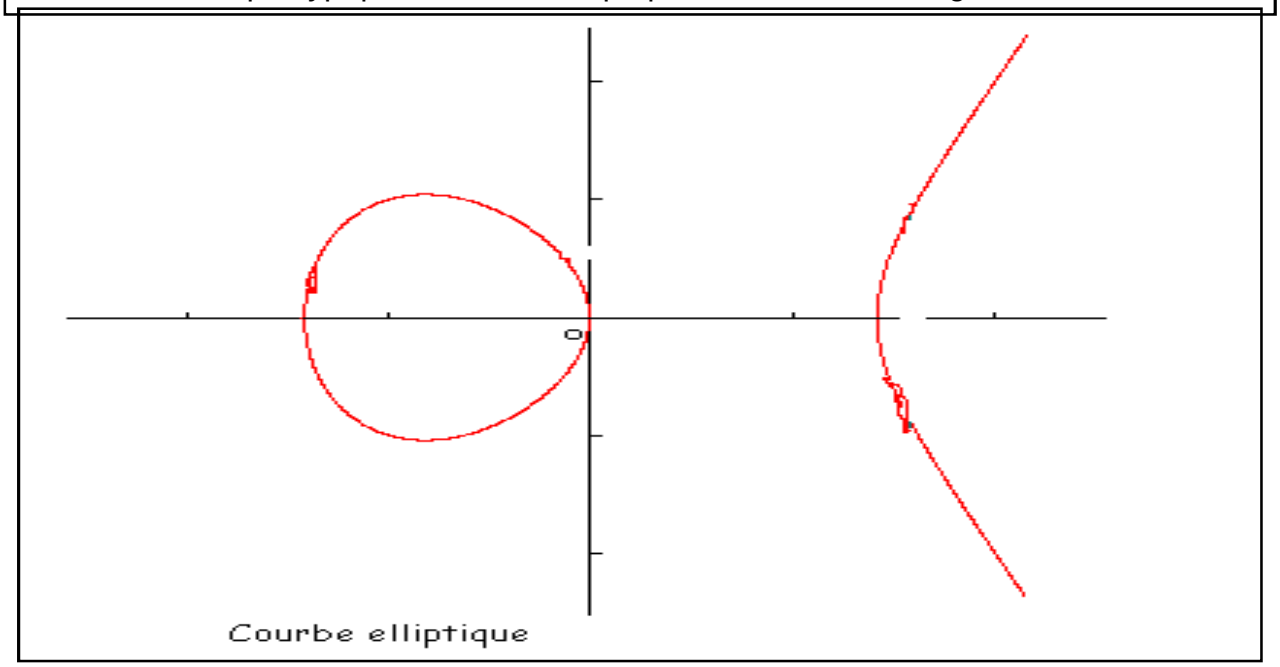
Définition 1:

Une courbe elliptique est une cubique plane E non singulière (1), d'équation de la forme:

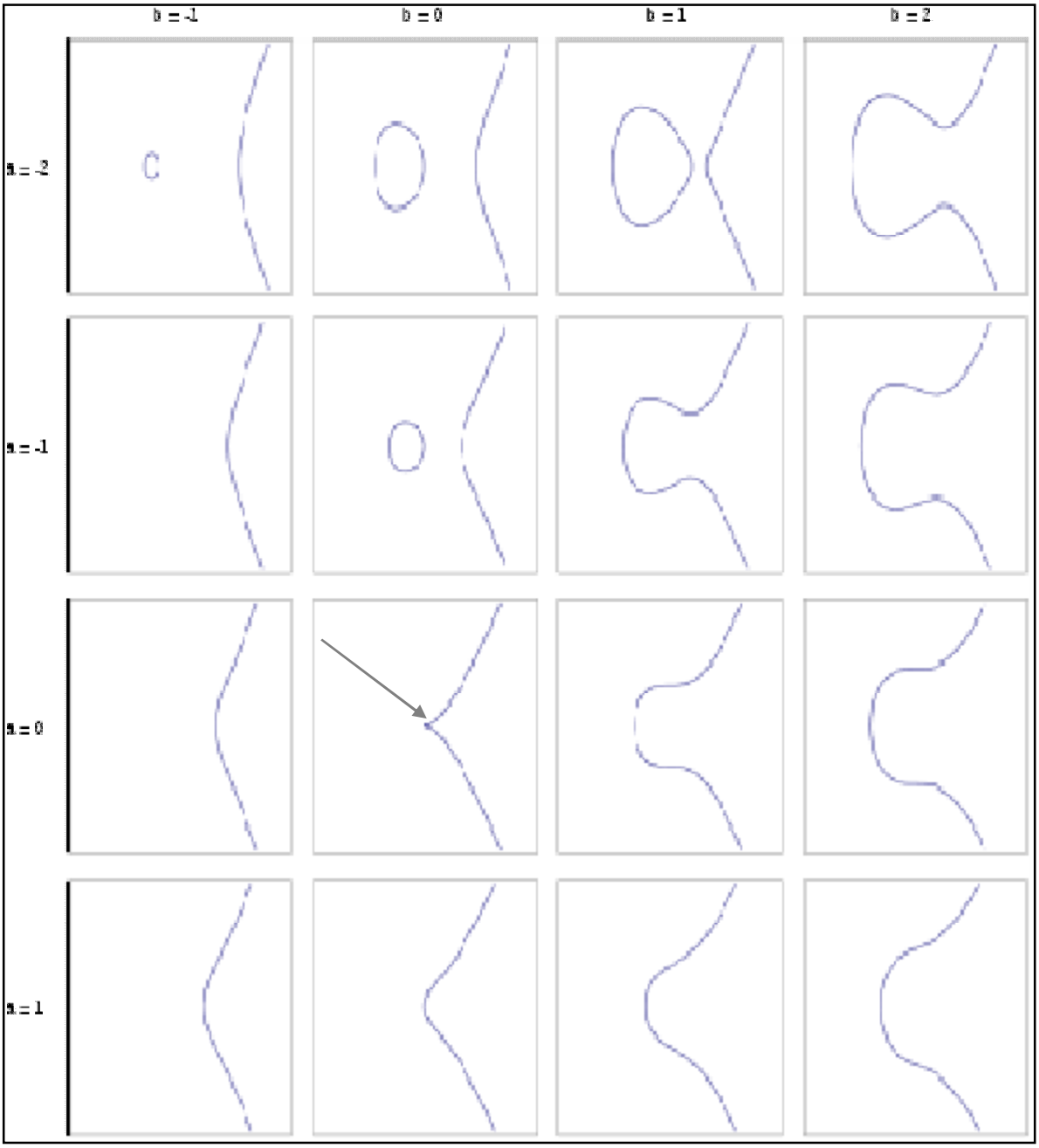
E: $y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$; Les cinq coefficients a_i sont des élément d'un corps commutatif quelconque K. les deux variables x et y sont des zéros de cette équation.

L'équation ci-dessus est dite L'équation de Weierstrass.

Ø Un exemple typique de courbe elliptique est donné sur la figure ci-dessous.



(1) **Les points singuliers** sonts les sont: les nœuds, les point de rebourssements,…, voir page16



Une sélection de courbes cubiques réelles définies par l'équation $y^2 = x^3 + ax + b$. La région montrée est $[-3,3]^2$. La courbe pour a=b=0 n'est pas elliptique.

1-2-b-Invariants de courbes elliptiques :

Toute courbe elliptique E possède plusieurs invariants:

Un discriminant, un invariant modulaire, un invariant différentiel, un conducteur, un régulateur, ….

Définition 2:

Le discriminant d'une courbe elliptique E, sur un corps K, est le polynôme 'homogène ' de l'anneau $K[b_2, b_4, b_6, b_8]$, égal à:

$$\Delta(E)=9b_2b_4b_6-8(b_4)^3-27(b_6)^2-(b_2)^2b_8$$

avec: $b_2=(a_1)^2+4a_2$, $b_4=a_1a_3+2a_4$, $b_6=(a_3)^2+4a_6$,

$4b_8=b_2b_6-(b_4)^2$ et carac(K) ≠ 2, 3.

v L'équation d'une courbe elliptique définie sur le corps des nombres réels peut être mise sous la forme plus simple:

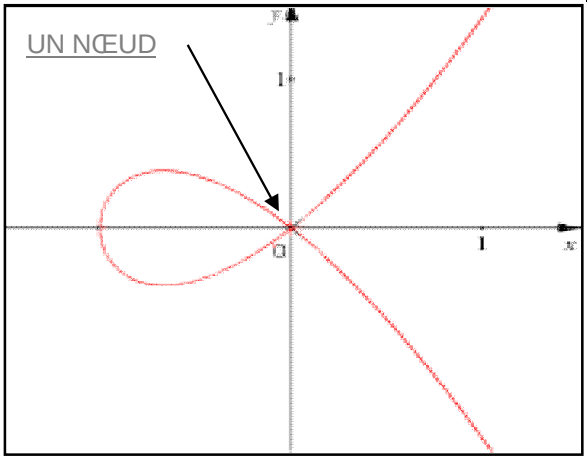
$$y^2 = x^3 + ax + b$$

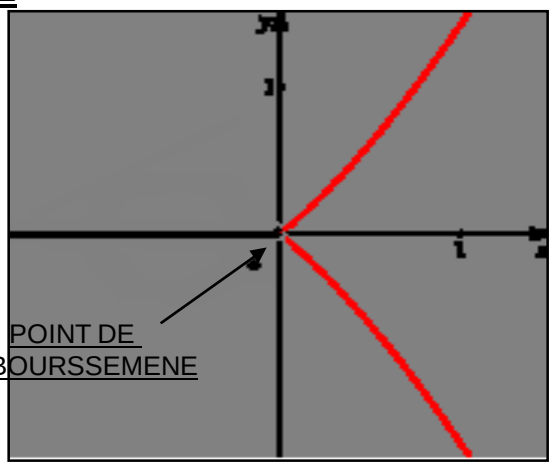
Où les coefficients a, b sont des nombres réels. Selon le choix de ces coefficients, les graphes correspondants ont des formes variées.

Et le discriminant devient : $\Delta(E) = -16(4a^3 + 27b^2)$.

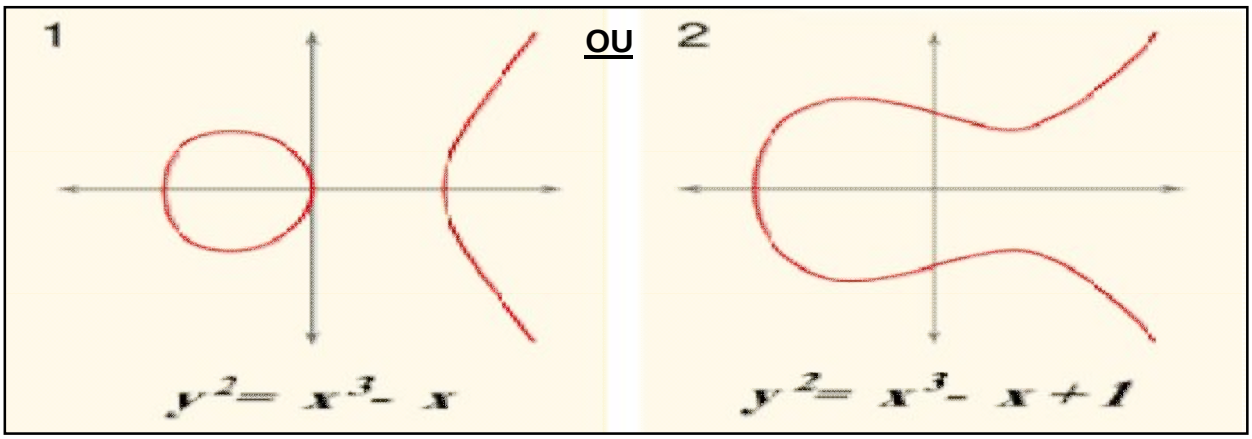
ØSi $\Delta(E) = 0$: la cubique plane E est singulière (n'est plus une courbe elliptique), son graphe est l'une des formes suivantes:

1OU 2





Ø Si $\Delta(E) \neq 0$: la cubique plane E est non singulière (c'est une courbe elliptique), son graphe de la forme:



1-2-c-Groupe de MORDELL-WEIL:

Les points de la courbe sont tous ceux dont les coordonnées (réelles) vérifient l'équation, ainsi qu'un point à l'infini. Comprendre comment et pourquoi ce point doit être pris en compte nécessite de se placer dans le cadre de la géométrie projective. Ce point à l'infini est essentiel car ce sera l'élément neutre (le zéro) pour l'addition des points de la courbe. Intuitivement, il suffit ici de l'imaginer comme le point à l'intersection de toutes les droites verticales.

Loi de groupe : résumé et justification

On vient donc de définir, géométriquement et sur les coordonnées, une ***loi de composition*** sur les points de la courbe E, notée + : autrement dit, on a défini pour tous les points Pet Q de la courbe le point P+Q .

On prend le point à l'infini comme ***élément neutre*** (on le note traditionnellement 0_E).

On prend pour opposé d'un point *P* le symétrique de *P* par rapport à la droite des abscisses; on le note − *P*.

Théorème 3:

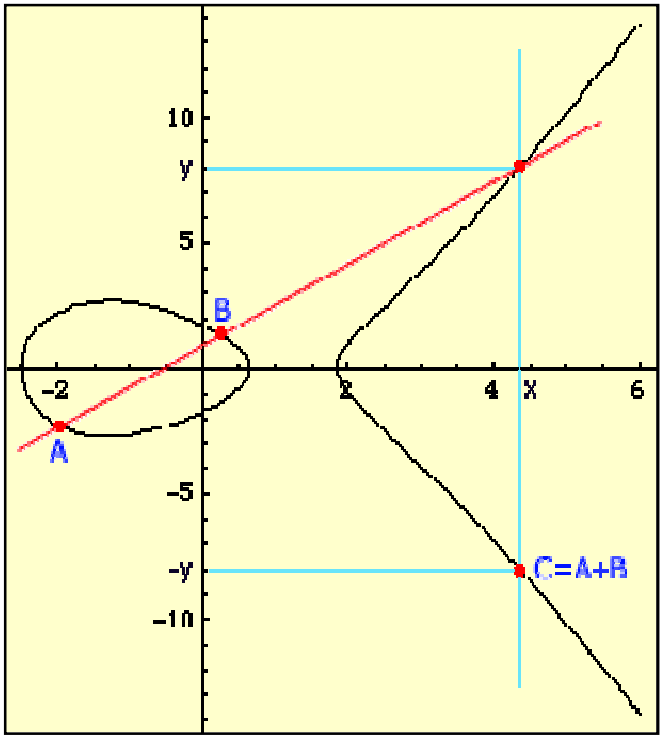
L'ensemble des points à coordonnées réelles de la courbe (en incluant le point à l'infini), muni de cette loi de composition, est un ***groupe commutatif*** [12].

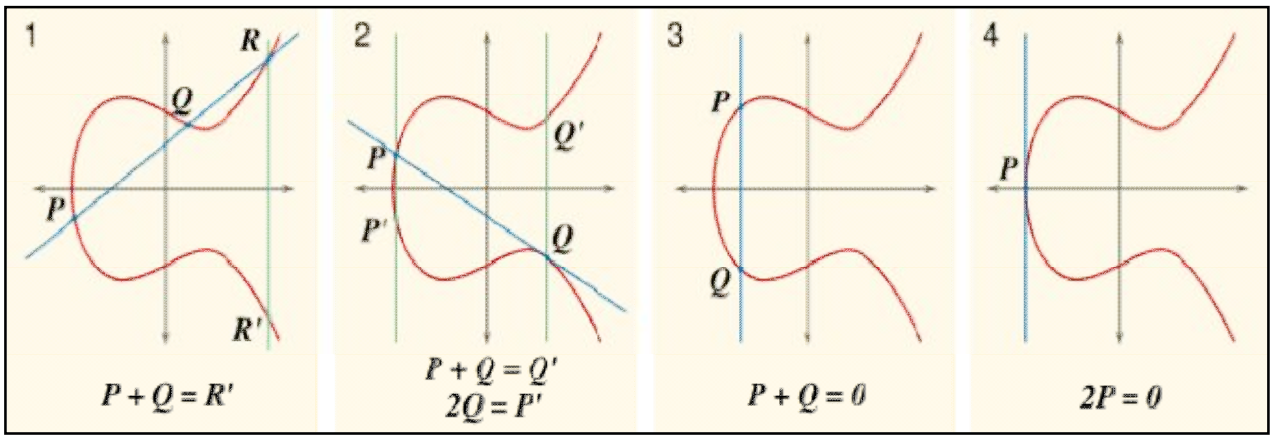
Additionner les points: définition par la méthode des tangentes et des sécantes:

L'addition de deux points sur une courbe elliptique est rendue possible par la propriété suivante, qui est la règle géométrique suivante « 3 POINTS COLINEAIRES DE E ONT UNE SOMME NULLE: $P+R+S=0_E$»

Addition de deux points

Prenons deux points A et B sur cette courbe. En général, la courbe passant par A et B recoupe la courbe en un troisième point de coordonnées (x, y). Son symétrique (x, -y) est lui aussi sur la courbe et on le désigne par A+B pour signifier qu'il est construit à l'aide de A et B. La chose surprenante est que cette opération "+" possède toutes les propriétés de l'addition des nombres c'est-à-dire que l'on peut faire tous les calculs de type addition, soustraction et division avec un reste entier que nous faisons sur la droite des nombres réels sur cet objet tordu que constitue une courbe elliptique.





Cas possibles d'addition de deux points.

Si K est un corps, disons de caractéristique différente de 2 et 3, on peut tout autant considérer l'ensemble des courbes (x, y) de K vérifiant $y^2 = x^3 + ax + b$, plus un point O à l'infini. En revanche, les calculs algébriques réalisés restent valables, et en appliquant les formules ci-dessous on peut toujours munir la courbe elliptique d'une structure de groupe commutatif noté E(K).

Si le corps K est fini, la courbe elliptique est un groupe commutatif fini, mais dont l'ordre et la structure sont difficiles à déterminer. Ils sont, en quelque sorte, des analogues plus compliqués aux groupes (Z/p Z)*.

Règles de calculs :

- P+O=O+P=P pour tout P dans E.
- Si P=(x,y), alors (x,y)+(x,-y)=O dans P. Le point (x,-y) est noté –P.
- Si P=(x,y) alors P+P=(x′,y′) avec $x'=t^2-2x$, $y'=-t^3+3tx-y$. $t=(3x^2+a)/2y$.
- Si $P=(x_1,y_1)$, $Q=(x_2,y_2)$, P¹Q, alors $P+Q=(x_3,y_3)$ avec $x_3=t'^2-x_1-x_2$, $y_3=-t'^3+t'(2x_1+x_2)-y_1$ où $t'=(y_2-y_1)/(x_2-x_1)$.

(E,+) est un groupe commutatif. [12]

Courbes elliptiques sur les corps finis:

Soit $K=F_q$ un corps fini à q éléments et E une courbe elliptique définie sur ce corps. Un premier résultat important concernant le nombre de points d'une courbe elliptique sur un corps fini, est le suivant :

1-2-d-Cardinalité d'une courbe elliptique sur un corps fini:

Théorème 4: (de Hasse [10])

Si E est une courbe elliptique définie sur le corps fini F_q alors :

$$q+1-2\sqrt{q} \leq card\ \ E(F_q) \leq q+1+2\sqrt{q}$$

Compter les points d'une courbe elliptique sur un corps fini:

Dans cette partie nous allons montrer qu'il est facile de calculer le cardinal d'une courbe $E(F_{q^n})$ si nous connaissons son cardinal pour $E(F_q)$. Ensuite nous allons donner un algorithme qui nous permet de calculer Card $E(F_p)$ pour un p premier.

Théorème 5 [10]:

Soit $CardE(F_q) = q + 1 - \varepsilon$; avec ε est un entier.

Posons: $X^2 - \varepsilon X + q = (X - \alpha)(X - \beta)$, ou, α et $\beta \in C$.

Alors: $CardE(F_{q^n}) = q^n + 1 - (\alpha^n + \beta^n)$, pour tout $n > 0$.

Exemple.

Considérons la courbe elliptique E: $y^2 = x^3 + 2$, définie sur F_7, alors un simple calcul montre que: $E(F_7) = \{O, (0, 3), (0, 4), (3, 1), (3, 6), (5, 1), (5, 6), (6, 1), (6, 6)\}$.

Ainsi $CardE(F_7) = 9$ et $\varepsilon = 7 + 1 - 9 = -1$ et nous avons le polynôme suivant:

$$x^2 + x + 7 = \left(x - \frac{-1+\sqrt{-27}}{2} \right)\left(x - \frac{-1-\sqrt{-27}}{2} \right)$$

Nous pouvons donc calculer le cardinal de tout groupe $E(F_{7^n})$.

Par exemple Pour n=60:

$$\left(\frac{-1+\sqrt{-27}}{2} \right)^{60} + \left(\frac{-1-\sqrt{-27}}{2} \right)^{60} = 1804985852\ 6119884806\ 006498$$

et donc: Card $E(F_{7^{60}}) = 7^{60} + 1 - 1804985852611988480600649 8 =$

508021860739623365322188179602357975652549718829504.

Grâce à ce théorème nous pouvons très vite calculer la cardinalité d'un groupe $E(F_{p^n})$ du moment que nous connaissons Card $E(F_p)$.

L'algorithme de Schoof:

C'est un algorithme due à René Schoof qui permet de calculer Card $E(F_p)$ pour tout nombre premier p. Sa complexité est O(ln8 p) [9]. Ainsi nous pourrons calculer Card $E(F_{p^n})$ grâce au théorème 5.

Schoof a évalué l'ordre d'un groupe $E(F_p)$ sous la forme : $p+1-t$,

ou t est une racine d'une équation dite de Frobenius [9].

q	$E(F_q)$	Card $E(F_q)$	Temps (sec)
11	$y^2 = x^3 + 8x + 1$	17	0.164835
13	$y^2 = x^3 + 2x + 9$	17	0.000000
17	$y^2 = x^3 + 9x + 5$	11	0.054945
19	$y^2 = x^3 + 5x + 12$	19	0.054945
23	$y^2 = x^3 + 2x + 6$	29	0.000000
29	$y^2 = x^3 + 22x + 16$	37	0.054945
31	$y^2 = x^3 + 5x + 3$	41	0.054945
37	$y^2 = x^3 + 8x + 14$	47	0.000000
41	$y^2 = x^3 + 8x + 4$	43	0.274725
43	$y^2 = x^3 + 27x + 22$	29	0.000000
47	$y^2 = x^3 + 38x + 6$	37	0.054945
53	$y^2 = x^3 + 5x + 12$	43	0.054945
59	$y^2 = x^3 + 4x + 49$	53	0.000000
61	$y^2 = x^3 + 31x + 49$	61	0.054945
67	$y^2 = x^3 + 2x + 56$	37	0.000000
71	$y^2 = x^3 + 57x + 14$	47	0.054945
73	$y^2 = x^3 + 33x + 34$	79	0.000000
79	$y^2 = x^3 + 75x + 6$	61	0.054945
83	$y^2 = x^3 + 3x + 78$	67	0.000000
89	$y^2 = x^3 + 54x + 52$	103	0.054945
97	$y^2 = x^3 + 32x + 33$	97	0.054945

Table: Programme Performance De schoof [13].

Chapitre 2:

Logarithme discret

ordinaire.

ØChapitre 2: Logarithme discret ordinaire (D-L).

Soient a, x, y des entiers.

Dans ce chapitre on va étudier l'équation $y=a^x$, donc trouver y (avec x donné) c'est Le problème d'exponentiation, réciproquement le calcul de x est difficile.

2-1-Définition du problème D-L:

Soit le groupe multiplicatif cyclique (Z/pZ)* avec p premier. Soit g un générateur de ce groupe (tous les éléments du groupe sont des puissances de g).

Le problème du logarithme discret de base g dans (Z/pZ)* est le suivant:

Problème: Étant donné un élément x de (Z/pZ)*, trouver l'entier y tel que l'on ait:

$$x = g^y \bmod(p).$$

On note parfois cet entier y, $\log_g(x)$, avec $0 \leq y \leq (p-1)$.

Donc : $\log_g(x)$: $(Z/pZ)^* \rightarrow (Z/pZ)^*$

$$x \rightarrow \log_g(x) = y$$

Exemple 1:

(Z/7Z)*= {1,2,3,4,5,6}. Remarquant que 3 et 5 sont des générateurs de ce groupe. Donc on peut définir un logarithme discret à la base 3 et à la base 5 comme suit:

X	1	2	3	4	5	6
$\log_3(x)$	6	2	1	4	5	3
$\log_5(x)$	6	4	5	2	1	3

Exemple 2:

Prenons le corps de cardinal 27, K est isomorphe au $F_3[x]/(x^3 + 2x + 1)$.

(Le polynôme $x^3 + 2x + 1$ est irréductible dans $F_3[x]$ vu qu'il est de degré 3 et qu'il n'a pas de racines dans F_3).

Le groupe K* est d'ordre 26. Les ordres de ses éléments autres que l' élément neutre, sont donc 2, 13 ou 26. En fait, −1 est le seul élément d'ordre 2 de K*, car par exemple ±1 sont les seules racines du polynôme $x^2 - 1$ de K[x]. Notons:

α la classe de x modulo $x^3 + 2x + 1$. Vérifions que α est un générateur de K*. On a, $\alpha^3 = \alpha - 1$, d'ou $\alpha^9 = \alpha^3 - 1$ (car K est de caractéristique 3) i.e. $\alpha^9 = \alpha + 1$, d'ou $\alpha^{12} = \alpha^2 - 1$ puis $\alpha^{13} = -1$ et notre assertion.

Résolvons alors le problème du logarithme discret de base α dans K*. Tout élément de K s'écrit de manière unique sous la forme: $a + b\alpha + c\alpha^2$ avec a, b, c de F_3. Il s'agit donc pour chacun de ces éléments de déterminer l'entier y tel que l'on ait $a + b\alpha + c\alpha^2 = y$ avec $0 \leq y \leq 25$.

On vérifie les calculs suivants:

X	1	2	α	α+1	α+2	2 α	2 α+1
$\log_{\alpha}(x)$	0	13	1	9	3	14	16

X	2 α+2	α^2	α^2+1	α^2+2	$2\alpha^2$	$2\alpha^2+1$	$2\alpha^2+2$
$\log_{\alpha}(x)$	22	2	21	12	15	25	8

X	$\alpha^2+\alpha+1$	$\alpha^2+2\alpha+1$	$\alpha^2+2\alpha+2$	$\alpha^2+\alpha+2$	$2\alpha^2+2\alpha+1$
$\log_{\alpha}(x)$	6	18	7	11	24

X	$2\alpha^2+2\alpha+2$	$2\alpha^2+\alpha+2$	$2\alpha^2+\alpha+1$	$\alpha^2+\alpha$
$\log_{\alpha}(x)$	19	5	20	10

X	$\alpha^2+2\alpha$	$2\alpha^2+2\alpha$	$2\alpha^2+\alpha$
$\log_{\alpha}(x)$	4	23	17

2-2-Algorithmes pour calculer le D-L:

Il est toujours possible, pour calculer le **Logarithme discret** de x, d'énumérer les éléments: g^0, g^1, g^2,…, jusqu'a ce que l'on rencontre x. Cependant, si cette méthode est tout à fait raisonnable pour les petits groupes, elle est totalement imaginable quand la cardinalité du groupe augmente.

Il y a plusieurs algorithmes pour calculer le D-L:

- L'algorithme baby-step giant-step dū de Shanks.
- L'algorithme ζ de Pollard.
- L'algorithme de Pohlig-Helman.
- L'algorithme de calcul d'index.

On va indiquer dans la suite l'un de ces algorithmes qui est l′ algorithme de baby-step giant-step dû de Shanks.

soit l'équation**: $g^{y} = x \bmod(p)$……..(1), (on cherche y).**

La résolution de cette équation se fait d'après Shanks [11] comme suit:

On écrit: y=au+b,

avec: u= $\lfloor\sqrt{p}\rfloor$ et: 0 **≤** a, b **≤** u-1, o**ŭ** $\lceil\sqrt{p}\rceil$ est le plus petit entier rationnel ›p.

L'équation (1) devient : $g^{au} = x\, g^{-b} \bmod(p)$.

On crée alors deux listes, constituant ainsi la méthode dite

''pas de géant, pas de bébé''.

pas de géant « g^{au} » :
1
g^{u}
g^{2u}
..
..
$g^{(u-1)u}$

pas de bébé « $x.g^{-b}$ » :
x
$x.g^{-1}$
$x.g^{-2}$
..
..
$x.g^{-(u-1)}$

La création des listes utilise $o(\sqrt{p})$ opérations et leur consultation a un coût de $o(\sqrt{p} \log p)$ [4].

Exemple [4]:

p=23, g=11, x=14.

Alors u=5, les deux listes sont alors:

Suite g^{au}:1, 5, 2, 10, 4 avec: 0 ≤ a ≤4.

Suite $x.g^{-b}$:14, 18, 10, 3, 17 avec: 0 ≤ b ≤4.

Il ya égalité pour a=3 et b=2 d'où **y=17**.

Complexité d'algorithmes : [3]

La complexité d'algorithmes pour calculer le Logarithme discret s'expriment en fonction de la taille du groupe G (card G). Mais du fait que la mach-ine (le processeur) fonctionne en binaire, on considérera la taille de groupe en $\log_2(p)$. Aussi une complexité polynômiale est en O(lnr p), r réel, et une complexité exponentielle est en $O(pr) = O(e^{r.\ln p})$. On définit naturellement une complexité sous exponentielle comme étant de la forme $O(e^{c.\ln a + o(1)p})$ où $a < 1$ et $c \in R$.

Le problème du logarithme discret est un problème généralement difficile, c'est-à-dire non résoluble en temps polynomial. On a donc construit des cryptosystèmes basés sur ce problème (comme les protocoles: Elgamal, …).

2-3-Cryptosystème d'Algamel

2-3-a-L'algorithme d'Algamel avec l'organigramme:

En 1985, Algamal a proposé un algorithme de chiffrement à clé publique. Cet algorithme concerne le problème de la confidentialité des messages envoyés, et son efficacité est aussi basée sur la difficulté du problème du logarithme discret, une personne, Ahmed, demande à Bilal de lui envoyer des messages confidentiels.

Une description est donnée ci-dessous:

Soient p un nombre premier et g un générateur du groupe (Z/pZ)*.

Ø Données communes: p et g.

Ø Clé privée d'Ahmed: x ∈ (Z/pZ)*.

Ø Clé publique d'Ahmed: $y = g^x \bmod p$.

v Chiffrement : soit m un message à chiffrer par Bilal.

Ce message m est codé comme un élément de (Z/pZ)*.

Bilal choisit un élément k de (Z/pZ)* puis il calcule R et S:

$R = g^k \bmod p$ et $S = m. y^k \bmod p$,

Un chiffré de m est la paire (R; S).

v Déchiffrement : seule Ahmed est capable de retrouver m

à partir du chiffré, grâce à sa connaissance de x.

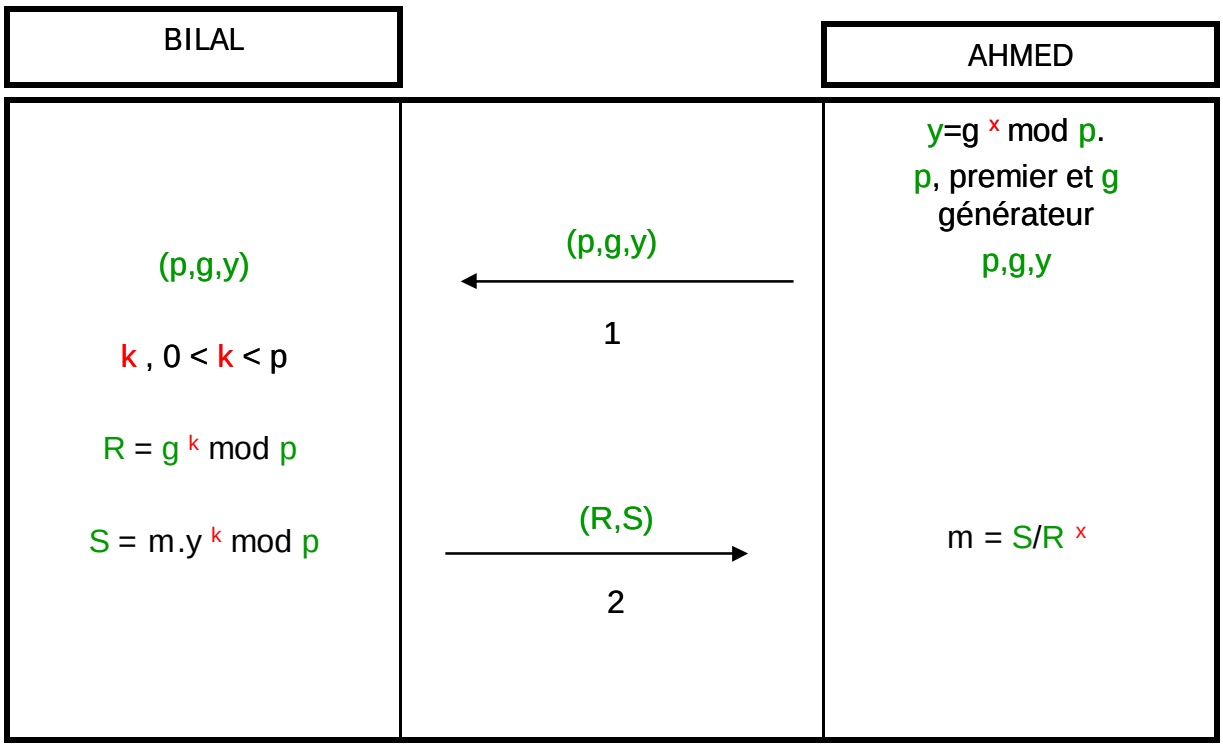
En effet,

$y^k = g^{xk} = (g^k)^x = R^x \bmod p$

Ainsi, **m** = $S/R^x \bmod p$.

Algorithme d'Algamal

l'organigramme:



m: le message à envoyer.

Les symboles en verts sont publiques.

Les symboles en rouge sont secrets.

2-3-b-exemple:

Soit le dictionnaire suivant:

A	B	C	D	E	F	G	H	I	J
01	02	03	04	05	06	07	08	09	10

K	L	M	N	O	P	Q	R	S	T
11	12	13	14	15	16	17	18	19	20

U	V	W	X	Y	Z	.	espace	?
21	22	23	24	25	26	27	28	29

Première étape:

Ahmed choisit p=31, premier.

Et soit g=11 un générateur du groupe (Z/31Z)*.

$y=g^{x} \bmod 31=(11)^{10}=5$, pour x=10 (x clé secrète de Ahmed).

Donc Ahmed publie (p, g ,y)=(31,11,5) et garde sa clé secrète x=10.

Deuxième étape:

Bilal veut envoyer à Ahmed le message suivant:

⟶ Il fait beau.

Chiffrement:

Ø Il convertit ce message à une suite d'entier **m** de (Z/31Z)*.

Ø Donc **m**= Il fait beau. = 091228060109202802050121 27 Bilal choisit un élément k=8 de (Z/31Z)* puis, il calcule R et S:

$R = g^{k} \bmod 31=11^{8} \bmod 31=19,$

$S = \mathbf{m}.y^{k} \bmod 31$

$= 091228060109202802050121 27.\mathbf{5^{8}\ mod}\ 31$

$=091228060109202802050121 27.\mathbf{25\ mod}\ 31$

=082118262508041821012529 24

Bilal envoi à Ahmed le chiffré de paire (R; S).

Déchiffrement :

Ahmed déchiffre le message **m** en utilisant sa clé secrète x=10 en calculant: S/R^{x} .

Car: $S/R^{x} = S/19^{10} \bmod 31$

$= S/25 \bmod 31$

$= S.5 \bmod 31$

$= (082118262508041821012529 24).5 \bmod 31$

= 0912280601092028020501212 7

= Il fait beau. → le message initiale.

Chapitre3:

MDS codes et exemple d'application:

Chapitre3: codes MDS et exemple d'application

3-1-Introduction:

La cryptographie tente de protéger les secrets des attaques d'éventuels fraudeurs, le codage à pour but de protéger les messages transmis des imperfections des systèmes de transmission tels que lignes téléphoniques, réseau hertzien, communication avec un satellite, disques compacts,….

Les imperfections peuvent être entre autres: parasites, défauts sur la ligne, problèmes de compréhension.

La théorie des codes étudie donc les moyens de protéger l'information que l'on désire transmettre des altérations qu'elle pourrait subir, altérations provoquées par les imperfections du moyen physique de transmission. La méthode utilisée consiste à envoyer sur le canal plus de données que la quantité d'information à transmettre.

Une redondance est ainsi introduite. Si cette redondance est structurée de manière exploitable, il est alors possible de corriger d'éventuelles erreurs introduites sur le canal.

On peut alors, malgré le bruit, retrouver l'intégralité des informations transmises au départ.

Une grande famille de codes correcteurs d'erreurs est constituée des codes par blocs. Pour ces codes l'information est d'abord coupée en blocs de taille constante et chaque bloc est transmis indépendamment des autres, avec une redondance qui lui est propre. La plus grande sous-famille de ces codes rassemble ce que l'on appelle les codes linéaires.

Dans un code linéaire, les messages que l'on veut coder sont lus sous la forme d'un k-uplet d'éléments d'un corps fini F: cet élément de F^k est en suite transformé en un élément de F^n par une application linéaire. La longueur n est choisie plus grande que la dimension k du code et c'est ainsi que la redondance est ajoutée [9].

En 1978, R.J. Mc-Eliece présenta un système de chiffrement à clé publique dont la sécurité reposait sur le problème du décodage borné d'un code correcteur d'erreurs. La clé privée est constituée d'un code correcteur structuré pour lequel on dispose d'un algorithme polynomial de décodage, et la clé publique est constituée d'une matrice génératrice du code, préalablement déstructuré. Tandis que d'autres systèmes furent cryptanalysés avec succès, celui-ci est l'un des rares à encore résister à toute cryptanalyse.

Dans ce chapitre, on s'intéresse à l'étude d'un cryptosystème de chiffrement basé sur les codes correcteurs d'erreurs.

3-2- Codes linéaires et introduction aux codes MDS :

Définition 3.2.1: Soit A un alphabet . Soit l'ensemble des vecteurs de dimmention n (n entier positif) formés à partir des éléments de A. Un code C est un sous-ensemble de vecteurs de A.

C sera dit binaire si A = {0, 1}; un élément de C est appelé mot de code.

Remarque **:** L' alphabet A sera le corps fini F_q de q éléments.

Définition 3.2.2: Soient u et v deux vecteurs de taille n:

$[u = (u_1,\ldots,u_n), v=(v_1,\ldots,v_n)]$, alors on définit :

– Le poids de u (noté wt(u)) est le nombre de composants non nuls de u.

–La distance de Hamming de u et v (notée d(u, v)) est le cardinal de l'ensemble des indices i tels que u_i soit différent de v_i: La fonction d(., .) ainsi définie vérifie les propriétés d'une distance au sence métrique [5].

Définition 3.2.3:

La distance minimale d'un code C notée d est égale à:

$d= \min \{d(u,v), \text{ pour } u, v \in C, u \neq v\}$.

Définition 3.2.4:

Un code linéaire C est un sous-espace vectoriel de l'espace $(F_q)^n$. On dit [n, k, d]code; n, k, d sont appelés les paramètres du code C.

n: longueur du code C

k: dimension du code C et d: la distance minimale .

conséquence [6]: d(u; v)= wt(u-v) et

$d= \min \{wt(z), \text{ pour tout mot } z \in C\}, z \neq (0, 0,\ldots,0)$

Définition 3-2-4: *Soit* C *un code linéaire de longueur* n *et de dimension* k. *Une* matrice génératrice *de* C *est une matrice:* k×n *dont les lignes forment une base de* C.

L'encodage**:**

Soit l'alphabet F_q de q éléments.

On définit l'espace des messages $(F_q)^k$.

L'encodage de C associé à la matrice génératrice G est l'application linéaire:

$$f : (F_q)^k \longrightarrow C \subset (F_q)^n$$

$$m = (m_1,\ldots,m_k) \longrightarrow f(m) = c = m.G \in C$$

$$= (c_1, \ldots, c_k,\ldots, c_n).$$

L'application f est appelé l'encodeur.

Définition-3-2-5: code systématique.

On dit que la matrice génératrice *du* [**n, k, d**] *code linéaire* C est sous forme standard ou normalisée si:

$G= [I_k \mid B]$, I_k la matrice d'identité, et B est une k×(n-k) matrice.

Un code C est dite systématique si sa matrice génératrice est sous forme standard.

Dans ce cas :

$$f : (F_q)^k \longrightarrow C \subset (F_q)^n$$

$$m = (m_1,\ldots, m_k) \longrightarrow f(m) = c = m.G$$

$$= (c_1,\ldots, c_k,\ldots, c_n)$$

$$= (m_1,\ldots, m_k , c_{k+1},\ldots, c_n).$$

Donc les k premiers symboles de C sont les symboles de l'information (le message), et les (n-k) suivants sont les symboles de contrôles.

Exemple:

Soient q=2, le corps F_2={0,1} et G la matrice génératrice *du code linéaire* C de paramètres [5, 3, d] telle que:

$$G = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \end{bmatrix}$$

G est sous forme standard donc C est systématique.

Si m=(m_1,m_2,m_3) de $(F_2)^3$ est un message, donc il est codé par le mot c $\in (F_2)^5$ tel que: c=m.G=(m_1, m_2, m_3, m_1+m_2, m_2+m_3).

Une base du code C est:

{a=(1,0,0,1,0), b=(0,1,0,1,1), c=(0,0,1,0,1)},

card(F_2^3) = 8.

Donc:

C={00000, 10010, 01011, 00101, 11001, 10111, 01110, 11100}.

Pour : m=000 → m.G= 00000

100 → 10010

010 → 01011

001 → 00101

110 → 11001

101 → 10111

011 → 01110

111 → 11111

On note aussi C={0_E, a, b, c, a+b, b+c, c+a, a+b+c}.

Définition 3-2-6: matrice de contrôle:

Soit C *un code linéaire de longueur* n *et de dimension* k.

Et soit S une application linéaire surjective:

S : $(F_q)^n \longrightarrow (F_q)^{n-k}$ telle que: Ker S =C,

on a donc: x ∈ C implique S(x)=0.

Définition 3-2-7**:**

La matrice de contrôle du code C est une matrice H de $M_{(n-k),n}(F_q)$ telle que:

pour tout $x=(x_1, x_2,\ldots, x_n) \in C$,

$$H.\begin{bmatrix} x_1 \\ x_2 \\ : \\ : \\ x_n \end{bmatrix} = 0$$, et on a: S(x)=0

Conséquence: [6]

H est une matrice de contrôle du code C implique $H.G^t = 0$.

Corollaire**:**

Si $G= [I_k \mid B]$ alors $H= [-B^t \mid I_{(n-k)}]$.

Par exemple :

$$G=\begin{bmatrix} 1 & 0 & a & b & c \\ 0 & 1 & d & e & f \end{bmatrix} \quad \text{implique}: H=\begin{bmatrix} -a & -d & 1 & 0 & 0 \\ -b & -e & 0 & 1 & 0 \\ -c & -f & 0 & 0 & 1 \end{bmatrix}$$

Et on vérifie que: $H.G^t=0$.

Remarque:

On considère le produit scalaire habituel sur $(F_q)^n$, et un code C de paramètre [n, k, d].

L'orthogonale de C, noté $C^{\perp}$, est un code linéaire de paramètres [n, n-k, d'].

Mais il n'y a pas une relation entre d et d'.

$C^{\perp} = \{ y \in (F_q)^n, x.y = 0, \text{pour tout } x \in C \}$.

Une matrice génératrice *de* C est une matrice de contrôle du code $C^{\perp}$, et réciproquement [10].

BORNE DE SINGLETON:

Proposition 3-2-8 : [6]

Un code linéaire C de paramètres [n, k, d] vérifie: $d - 1 \leq n - k$.

Définition 3-2-8:

Un code linéaire C de paramètre [n, k, d] est dit MDS (en Anglais: Maximum Distance Separable) si: $d - 1 = n - k$.

(BORNE DE SINGLETON est atteinte).

Codes MDS triviaux**:** sont des codes de paramètres:

$$[n, n, 1], [n, 1, n], [n, n-1, 2].$$

Théorème 3-2-9 [6]:

Un code linéaire de paramètre [n, k, d] détecte (d-1) erreurs (1) et corrige [(d-1)/2] erreurs.

[x]: désigne la partie entière de x.

Remarque:

Les codes MDS sont les meilleurs codes pour les corrections d'erreurs, (ayant une correction maximale d'erreurs).

(1)l'érreur c-a-d le changement dans les composants des vecteurs.

3-3-Le décodage des codes linéaires:

Dans ce paragraphe on considère un *code linéaire* C de paramètres [**n,k,d**], de matrice génératrice *G, et de* matrice de contrôle H, et t= [(d-1)/2] la capacité de correction.

S: $(F_q)^k \rightarrow C \subset (F_q)^{n-k}$ est l'application linéaire de matrice H, le syndrome de $x \in (F_q)^n$ est: $S(x)=H.x \in (F_q)^{n-k}$.

Conséquence : [6]

q $y \in C$ implique $S(y)=0$.

q Si $y=c+e$, où $c \in C$ est le mot émis et $e \in (F_q)^n$ l'erreur,

alors: $S(y)=S(e)$

q Si $wt(y_1) \leq t$ et $wt(y_2) \leq t$ alors:

$S(y_1)= S(y_2)$ implique $y_1=y_2$.

Décodage borné:

Pour décoder tout mot reçue y de $(F_q)^n$, contenant au plus t erreurs:

$y= c+e_y$, avec $c \in C$ est $wt(e_y) \leq t$.

On calcule $S(y) = H.y^t = s$; on sait que: $S(y) = s(e_y)$:

1-Si s figure dans la table à e_i on décode y par $y-e_i$.

(e_i: tous les erreurs dans $(F_q)^n$ telles que $wt(e_i) \leq t$).

2-Sinon, on peut dire que y est affecté de plus de t erreurs et on ne peut pas le décoder.

-Cette méthode est efficace mais coûteuse.

Exemple: Soit le code C sur F_2 de matrice génératrice G:

$$G=\begin{bmatrix}1&0&0&0&1&1&1\\0&1&0&0&1&0&1\\0&0&1&0&1&1&0\\0&0&0&1&0&1&1\end{bmatrix} \text{ donc } H=\begin{bmatrix}1&1&1&0&1&0&0\\1&0&1&1&0&1&0\\1&1&0&1&0&0&1\end{bmatrix}$$

Ce code permet donc de coder 4 symboles d'information, et si le message est $m=(m_1,m_2,m_3,m_4)$, le mot de code correspondant est: $c= m.G$

$c=(m_1, m_2, m_3, m_4, m_1+m_2+m_3, m_1+m_3+m_4, m_1+m_2+m_4)$.

Le code $C=\{0000000,1000111,0100101,\ldots,1111111\}$, est de 16 éléments, $(\text{card } F_2)^4=16$.

La distance minimale de ce code est d=3, donc ce code corrige une erreur car

$t= [(d-1)/2]=1$.

Par exemple si, $y =(1,1,0,1,1,1,1)$ alors $S(y)=H.y^t=(1,1,0)$

Donc l'erreur est: $e=(0,0,1,0,0,0,0)$.

On décode y par $c = y-e = y+e =(1,1,1,1,1,1,1)$.

3-4-Construction des codes MDS:

But **:** Construire une matrice kxn dont toutes les k colonnes sont linéairement indépendants.

Il ya plusieurs constructions pour obtenir des codes MDS, citons quelque constructions:

Code de Reed-Solomon: Soit a un élément de F_q d'ordre n ($a^n=1$), la matrice G_{RS} suivante génère un code MDS:

$$G_{RS}=\begin{bmatrix}(a^0)^0 & (a^0)^1 & (a^0)^2\ldots\ldots\ldots\ldots & (a^0)^{n-1}\\(a^1)^0 & (a^1)^1 & (a^1)^2\ldots\ldots\ldots\ldots & (a^1)^{n-1}\\\ldots & \ldots & \ldots & \ldots\\(a^{k-1})^0 & (a^{k-1})^1 & (a^{k-1})^2\ldots\ldots\ldots\ldots & (a^{k-1})^{n-1}\end{bmatrix}$$

Exemple :

soit le corps fini F_q, avec q=11.

Soit a=3, a est d'ordre 5, car:

$a^5=3^5=1 \bmod 11$. Donc on peut construire un code MDS de matrice génératrice de Reed-Solomon:

c-à-d : $$G_{RS}=\begin{bmatrix}1 & 1 & 1 & 1 & 1\\ 1 & 3 & 9 & 5 & 4\\ 1 & 9 & 4 & 3 & 5\end{bmatrix}\in M_{3,5}(F_{11})$$

Code de Maximum Runk-Distance:

Soient: $a_1, a_2, \ldots\ldots, a_n$ des éléments distincts deux à deux de F_q,

avec $q=p^m$, la matrice G_{MRS} génère un code MDS:

$$G_{MRS}=\begin{bmatrix}(a_1)^1 & (a_2)^1 & (a_3)^1 \ldots\ldots\ldots\ldots & (a_n)^1\\ (a_1)^p & (a_2)^p & (a_3)^p \ldots\ldots\ldots\ldots & (a_n)^p\\ \ldots & \ldots & \ldots & \ldots\\ (a_1)^{p^{k-1}} & (a_2)^{p^{k-1}} & (a_3)^{p^{k-1}} \ldots\ldots\ldots\ldots & (a_n)^{p^{k-1}}\end{bmatrix}$$

Code MDS générés par les matrices de Cauchy:

Soient dans le corps F_q les éléments (a_i) et (b_j), avec $a_i+b_j \neq 0$ pour tout i=1,…, n et j=1,…, k, Les matrices de Cauchy sont de la forme:

$$G=\begin{bmatrix}\frac{1}{a_1+b_1} & \frac{1}{a_2+b_1} & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \frac{1}{a_n+b_1}\\ \frac{1}{a_1+b_2} & \frac{1}{a_2+b_2} & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \frac{1}{a_n+b_2}\\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots\\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots\\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots\\ \frac{1}{a_1+b_k} & \frac{1}{a_2+b_k} & \ldots & \ldots & \ldots & \ldots & \ldots & \ldots & \frac{1}{a_n+b_k}\end{bmatrix}$$

Code de Reed-Solomon généralisés [2]:

Les codes de Reed-Solomon généralisés dits codes GRS sont MDS, leurs paramètres atteignent la borne de Singleton d=n-k+1.

Soient $V=(v_1,v_2,\ldots,v_n)$ un vecteur de $(F^*_q)^n$, et $A=(a_1,a_2,\ldots,a_n)$ un vecteur de $(F_q)^n$, ou les a_i sont distincts deux à deux.

La matrice:

$$GRS_k(A,V)=\begin{bmatrix} v_1.a_1^{\,0} & v_2.a_2^{\,0} \ldots\ldots\ldots\ldots\ldots & v_n.a_n^{\,0} \\ v_1.a_1^{\,1} & v_2.a_2^{\,1} \ldots\ldots\ldots\ldots\ldots & v_n.a_n^{\,1} \\ \ldots & \ldots & \ldots \\ \ldots & \ldots & \ldots \\ v_1.a_1^{\,k-1} & v_2.a_2^{\,k-1} \ldots\ldots\ldots\ldots\ldots & v_n.a_n^{\,k-1} \end{bmatrix}$$

engendre un code MDS de longueur n.

L'ensemble des codes $GRS_k(A,V)$ est appelé famille des codes de **Reed-Solomon généralisés.**

Propriétés: [2]

P1. Il existe un algorithme de décodage en temps polynomial de $GRS_k(A,V)$ qui permet de décoder $t=[(d-1)/2]$ erreurs.

P2. Le code orthogonal $C^{\perp}$ de $GRS_k(A,V)$ est le code $GRS_{n-k}(A,V')$, V' est déterminé en fonction de V.

3-5- Cryptosystème basé sur les codes.

3-5-a- Cryptosystème Mc–Elliece

L'algorithme avec l'organigramme:

C'est un cryptosystème à clé publique basé sur les codes correcteurs d'erreurs . On va s'intéresser au code MDS de Reed-Solomon généralisé à dimension et longueur donnés ce sont les codes qui ont la plus grande distance minimale possible.

Pour une taille convenable de clé ces codes offrent la sécurité optimale contre les attaques par décodage pour les cryptosystèmes basés sur les codes correcteurs.

Les codes de Reed-Solomon (MDS) peuvent donc être utilisés comme espace des clés d'un cryptosystème de Mc–Elliece.

Le cryptosystème de Mc–Elliece fut inventé en 1978 par Robert Mc-Eliece, la première qualité qu'ont peut trouver à ce cryptosystème est sa vitesse de chiffrement qui est aussi rapide que celle des autres cryptosystèmes asymétriques car il repose sur les calculs matriciels.

Génération des clés:

1-Ahmed génère un code linéaire (ce code supposé MDS) de paramètres [n, k, d], de matrice génératrice $G = G_{RS}$ (pour avoir une sécurité optimale). Ce code doit posséder un algorithme de décodage efficace.

2-Il sélectionne une matrice $S \in M_{k,k}$ aléatoire et inversible.

3-Il sélectionne aussi une matrice de permutation $P \in M_{n,n}$.

4-Il calcule $G' = SGP$, de $M_{k,n}$.

5-Il sélectionne un entier t tel que $t \leq [(d-1)/2]$.

Donc:

(G', t) est la clé publique.

(S, G, P) est la clé privée.

v Le chiffrement:

Pour chiffrer une suite de message m de longueur k il faut:

1-Calculer le vecteur $c' = m.G'$.

2-Générer un vecteur d'erreur e de poids t.

3-Calculer et envoyer le chiffré: $c = c' + e$.

v Le déchiffrement :

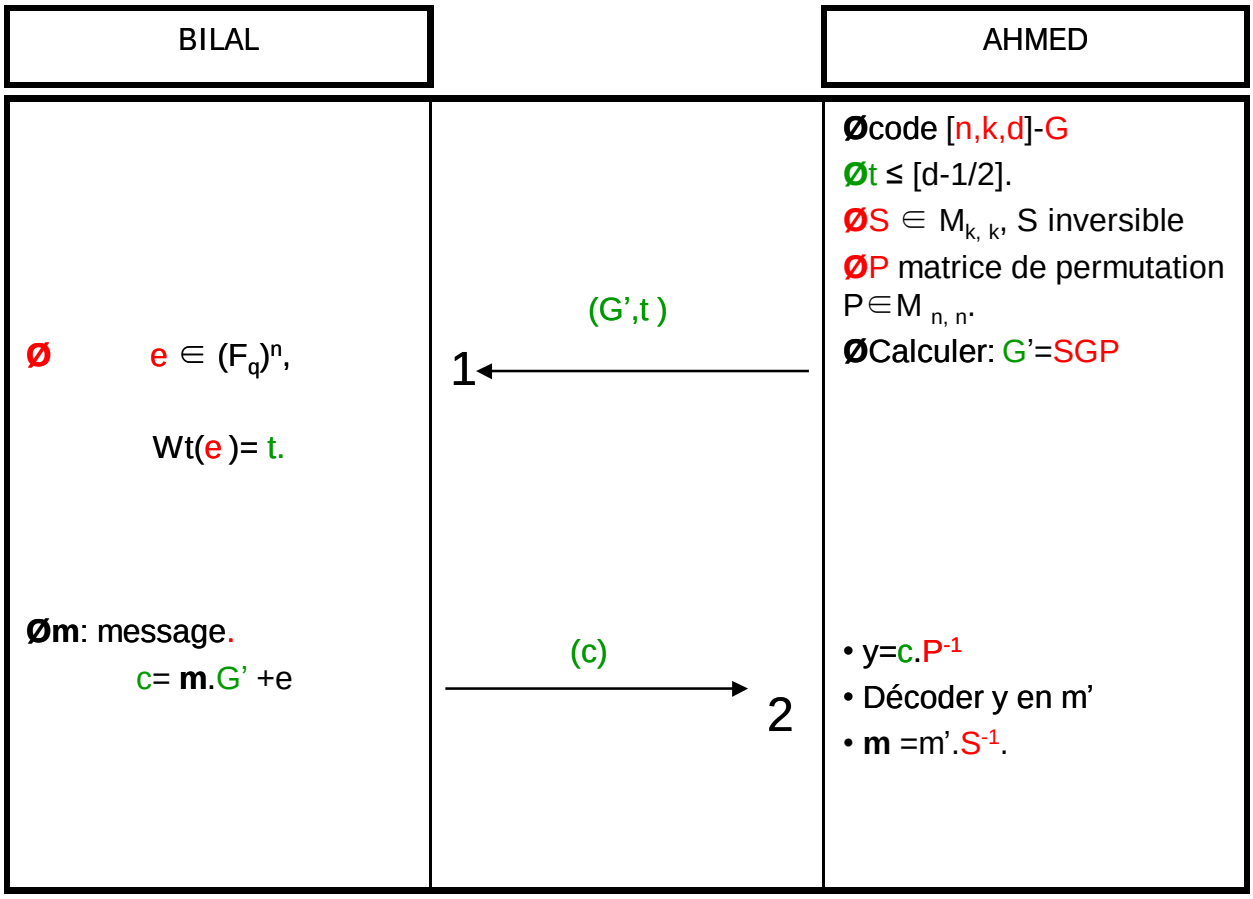
1-Calculer: $y = c.P^{-1} = mSG + e.P^{-1}$.

2-Utiliser l'algorithme de décodage du code C pour décoder y en un mot m',

(car $e.P^{-1}$ et e ont le même poids donc m.S est un mot de C).

3-Calculer: $m'.S^{-1} = m$, pour reconstruire le message m.

l'organigramme:



m: le message à envoyer.

Les symboles en verts sont publiques.

Les symboles en rouge sont secrets.

3-5-b-Exemple numérique sur le Cryptosystème Mc–Elliece:

Génération des clés:

Ø1-Soit le code C de paramètre [5,3,3] , de matrice génératrice *G* =GRS sur le corps F_7={0, 1, 2, 3, 4, 5, 6}.

Et soient V un vecteur de $(F_7)^5$; V=(2,3,4,5,2) et un vecteur générateur A=(6,3,2,5,4).

Donc: $G = GRS = \begin{bmatrix} 2 & 3 & 4 & 5 & 2 \\ 5 & 2 & 1 & 4 & 1 \\ 2 & 6 & 2 & 6 & 4 \end{bmatrix}$, $H = \begin{bmatrix} 1 & 0 & 1 & 5 & 2 \\ 2 & 1 & 0 & 6 & 6 \end{bmatrix}$

ce code corrige une erreur (t=1).

Ø2-Sélectionnement d'une matrice S de $M_{3,3}$ quelquonce et inversible.

Soit: $S = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 2 & 0 \\ 1 & 0 & 4 \end{bmatrix}$ inversible, donc $S^{-1} = \begin{bmatrix} 6 & 0 & 2 \\ 0 & 4 & 0 \\ 2 & 0 & 5 \end{bmatrix}$

Ø3-Sélectionnement aussi d'une matrice de permutation P de $M_{5,5}$.

Soit: $P = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}$ donc : $P^{-1} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \end{bmatrix}$

Ø4-Calcul de G'=SGP $\in M_{3,5}$

$$G'=SGP=\begin{bmatrix}2 & 4 & 4 & 6 & 6\\ 4 & 1 & 3 & 2 & 2\\ 6 & 1 & 3 & 4 & 5\end{bmatrix}$$

Ø5-Donc :

(G',t)=(G',1) est la clé publique.

(S, G, P) est la clé privée.

v Le chiffrement:

Soit le message **m** à envoyer, **m**=[1 1 0].

1-L'émetteur génère un vecteur d'erreur e=[2 0 0 0 0].

2-Il calcule et envoi: c=c'+e = m.G'+e = [1 5 0 1 1].

v Le déchiffrement:

1-Le récepteur calcule: $y=c.P^{-1} = [1\ 5\ 0\ 1\ 1].P^{-1}$

$= [0\ 1\ 1\ 5\ 1].$

2-Il décode y en un mot de code C le plus proche de y.

Le plus proche de [0 1 1 5 1] est [0 6 1 5 1].

Car C={00000, 23452, 52141, 26264,……, **06151**,…….}.

le mot correspondant à [0 6 1 5 1] est:

m'= [1 2 1].

3-Il calcule: $m'.S^{-1}= [1\ 2\ 1].S^{-1}= [1\ 1\ 0]=m$: qui est le message initiale.

Conclusion :

Dans cette étude on remarque que les points forts de cryptosystème Mc-Eliece sont la rapidité et la sûreté, mais la taille des clés est un vrai problème.

En effet, ce problème a été pendant longtemps l'inconvénient majeur qui a valu à ce cryptosystème d'être inexploitable et inutilisable.

Chapitre 4:

Cryptographie basée sur les Courbes Elliptiques:

Chapitre 4: Cryptographie basé sur les Courbes Elliptiques:

4-1- introduction:

Dans ce chapitre en va donner un exemple sur les cryptosystèmes symétriques qui utilisent la même clé pour le chiffrement et le déchiffrement, donc il nous faut un moyen pour transporter cette clé commune, il faut l'échanger en toute sécurité.

4-2-le protocole d'échange de clé de Diffie-Hellman :

Il s'agit d'un échange de clé par une courbe elliptique.

1-Ahmed et Bilal se mettent d'accord ensemble publiquement sur une courbe elliptique $E: y^2=x^3+ax+b$, sur un corps fini: $K=F_p=Z/pZ$, p: premier.

- Ils se mettent aussi d'accord sur un point P de E(K).

2-Secrètement:

Ahmed choisit un entier K_A et Bilal un entier K_B .

3-Ahmed envoie à Bilal le point K_AP, et Bilal envoie à Ahmed le point K_BP.

4-Chacun d'eux est capable de calculer $K_A(K_BP)=K_B(K_AP)=(K_AK_B)P$ qui est un point de la courbe E(K), ce dernier constitue leur clé secrète.

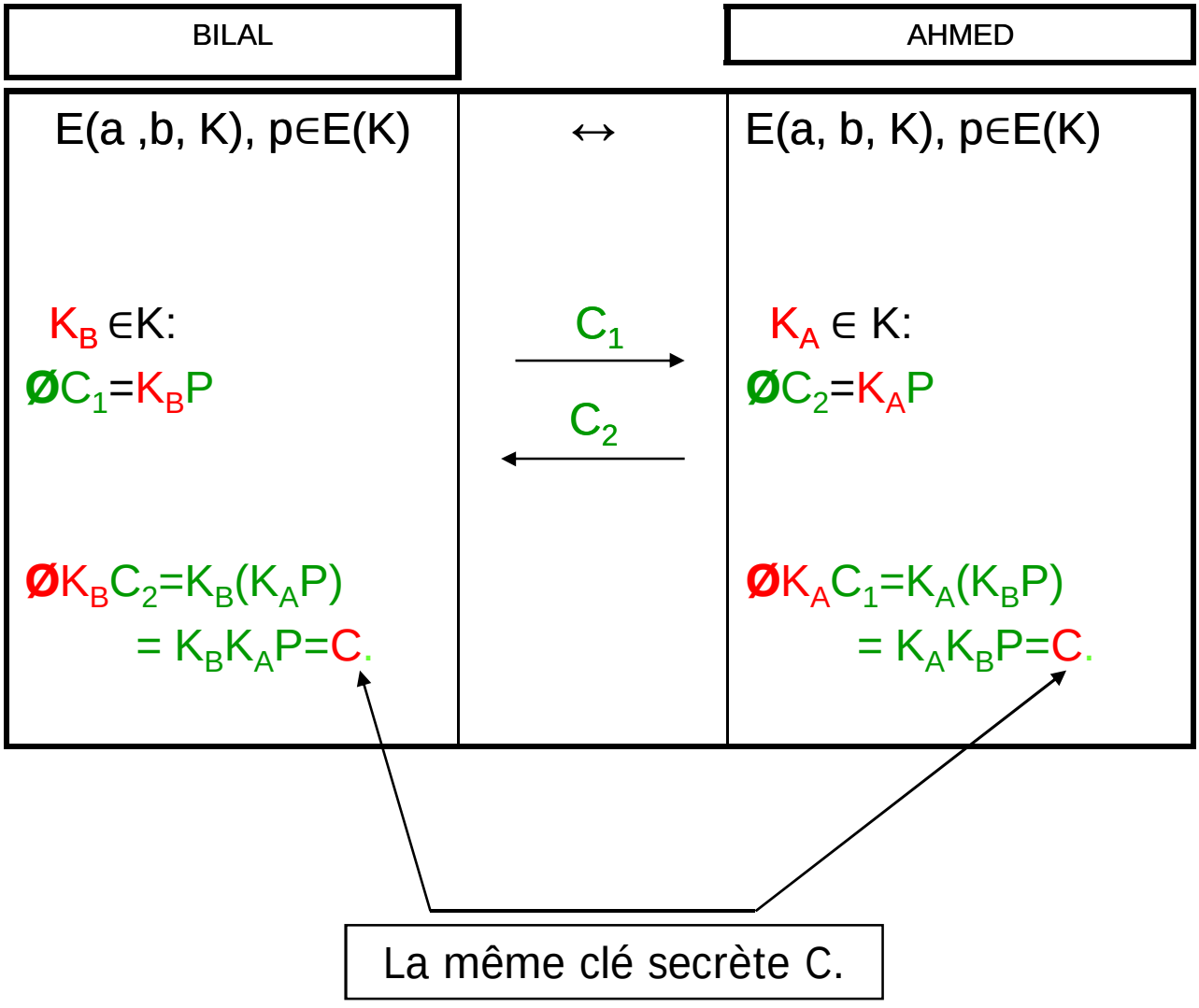
Protocole d'échange de clé de Diffie-Hellman

Si quelqu'un a espionné leur échange, il doit connaître E(a, b, K), P, K_A P, K_BP pour pouvoir calculer K_AK_BP et il faut résoudre un problème semblable au problème du logarithme discret mais sur une courbe elliptique, ce qui n'est pas évident en effet

Il faut pouvoir calculer K_A connaissant P et K_AP.

-Le logarithme discret est déjà difficile à résoudre dans les groupes bien connues (Z/pZ)*. Pour les groupes des courbes elliptiques, c'est encore plus difficile….

l'organigramme:



Exemple: -le protocole de Diffie-Hellman :

Soit la courbe elliptique $E(F_{23})=E(Z/23Z)$ d'équation:

$y^2=x^3+x+1$, et soit le point $P=(3,10)\in E$.

$E(F_{23})=\{0_E, (0,1), (0,22), (1,7), (1,16), (3,10), (3,13), (4,0), (5,4), (5,19), (6,19), (6,4),$,(7,12), 7,11) ,(9,7) ,(9,16) ,(11,3) ,(11,20) , (12,4) , (12,19) , (13,16) , (13,7) , (17,3) (17,20), (18,3), (18,20), (19,5), (19,18)}.

1-Pour $K_A=4$, Ahmed calcule et envoit à Bilal:

$C_2=K_AP=4P=4(3,10)=(17,3)=C_2$.

-Pour $K_B=2$, Bilal calcule et envoit à Ahmed:

$C_1=K_BP=2P=2(3,10)=(7,12)=C_1$.

2-Ahmed et Bilal calculent chacun d'eux:

Ahmed: $K_AC_1=K_A(K_BP)= 4(7,12) =(\mathbf{13},16) = C$.

Bilal: $K_BC_2 = K_B(K_AP) = 2(17,3) = (\mathbf{13},16) = C$.

-Donc C=(13,16) est un point de la courbe $E(F_{23})$ qui est un point secret commun de Ahmed et Bilal. Alors la clé secrète commune sera par exemple la première composante du point C c-à-d:

K_c = clé secrète commune = **13**.

4-3-a- Cryptosystème basé sur le protocole Diffie-Hellman:

L'algorithme:

On suppose que Ahmed et Bilal ont suivi le protocole d'échange de clé de Diffie-Hellman.

Bilal veut envoyer à Ahmed un message m.

1-Il convertit tout d'abord son message à une suite de points m sur la courbe elliptique E(a, b, K).

2-Il choisit secrètement un entier β et envoi à Bilal le chiffré (m_1, m_2) avec:

$m_1 = \beta.P$ et $m_2 = m + \beta.K_C P$.

P: point quelconque de la courbe E.

K_C: la clé secrète commune échangée suivant le protocole.

3- Ahmed déchiffre le message initiale en calculant :

$$m_2 - K_C m_1 = m.$$

Car: $m_2 - K_C m_1 = (m + \beta.K_C P) - K_C(\beta.P)$

$$= m + \beta.K_C P - K_C \beta.P = m.$$

m: le message à envoyer.

Les symboles en verts sont publiques.

Les symboles en rouge sont secrets.

4-3-b-Exemple numérique:

Supposant que la clé secrète commune échangé suivant le protocole de Diffie-Hellman est $K_C=13$ donné dans l'exemple précédent.

-Ahmed veut envoyer à Bilal le message m=(12,4) qui est un point de la même courbe $E(F_{23})=E(Z/23Z)$ d'équation :

$y^2=x^3+x+1$, et soit le même point P=(3,10).

1- Le message m=(12,4) qui est un point de la même courbe.

2-Il choisit secrètement un entier β=2 et envoi à Bilal le chiffré (m_1,m_2) avec

$m_1= \beta.P=2P=2(3,10)=(7,12)$ et $m_2=m+ \beta.K_C P= (12,4)+2.13.(3,10)$

$m_2 = (12,4)+\underline{26.(3,10)}$

$= (12,4)+ (7,11).$

$=(17,3).$

Car 26P=13(2P)

=13P' , pour P'=2P=2(3,10)=(7,12).

=P'+12P' =P'+6(2P')

=P'+6P'', pour P''=2P'=2(7,12)=(17,3).

=P'+3(2P'') =P'+3P''', pour P'''=2P''=2(17,3)=(13,16).

=P'+P'''+2P''' =(P'+P''')+P'''', pour P''''=2P'''=2'13,16)=(5,19).

=(6,4)+(5,19) = (7,11).

3-Bilal déchiffre le message initiale en calculant:

$m_2 – K_C m_1 = (17,3) -13(7,12)$

$= (17,3)- (7,11)$

$= (17,3)+(7,12)$

$= (12,4)= m.$

Rappel sur les règles de calculs:

- P=(x ,y) donc: -P=(x,-y).
- P=(x, y), P'=(x',y') donc: $P+P'=(x'', y'')=(t^2-x-x', -t^3+t(2x+x'')-y)$,
  avec $t=(y'-y).(x'- x)^{-1}$.
- P=(x, y) donc $P+P=2P=(x_{2P}, y_{2P})=(t^2-2x, -t^3+t(x-x_{2P})-y)$,
  avec $t=(3x^2+a).(2y)^{-1}$.

Conclusion:

La cryptographie à courbes elliptiques est une alternative à la cryptographie classique à clé publique.

La taille des clés permet de réserver un espace mémoire au niveau de processeur.

La cryptographie à courbes elliptiques est une approche appelée à se répandre dans les applications pratiques.

Il existe d'autres constructions basées sur les courbes elliptiques(courbes hyper-elliptiques de Kobler).

CONCLUSION:

A travers cette étude, nous avons constaté que les techniques de cryptage ont beaucoup progressé au cours des âges. Cette évolution est la conséquence des décryptages successifs qui ont poussé l'homme à développer des techniques de plus en plus élaborées. La sensibilité des informations aidant, l'homme fut obliger après avoir longtemps fait confiance à de simples substitutions (disque à chiffrer, Scytale...) à passer à des systèmes plus complexes.

On a pu donc partager l'histoire du cryptage en deux parties l'avant et l'après Enigma. La complexité de la machine allemande a forcé les ingénieurs à créer le premier ordinateur (Colossus) afin de pouvoir décrypter les messages en sa provenance. Dès l'avènement de l'informatique les techniques de cryptage ont franchis un grand pas : l'apparition des premiers réseaux informatiques et la multiplication des échanges de données sous forme numérique ont forcé les mathématiciens à révolutionner entièrement les systèmes de cryptage (AlGamel, Mc-Eliece, DLP,RSA…).

Le problème du logarithme discret sur les courbes elliptiques(DLP) est une preuve de la complexité des récentes techniques de cryptage dans le monde informatique. Ce sont tous les scientifiques et chercheurs qui se mettent au travail, il n'est plus question d'un génie qui trouve le miracle d'un cryptage inviolable mais de protocole et de systèmes entiers normalisés qui se mettent en place.

Aujourd'hui on se retrouve avec des codes accessibles au plus grand nombre et dont la complexité était inimaginable il y a encore cinquante ans. Ce n'est plus un sujet secret mais ce sont des recherches académiques lancées à travers le globe. C'est une démocratisation de la cryptographie qui ne se limite plus aujourd'hui à l'échange de données stratégiques. Malgré toutes les améliorations ces systèmes ne sont toujours pas inviolables. La bataille du cryptage entre ceux qui tentent de protéger les données et ceux qui tentent de les détourner a donc encore de beaux jours devant elle.

Nous restons ébahi par le bouleversement qu'à subit le monde la cryptologie en pensant que chaque évolution technologique en matière de communication affecte profondément l'univers de la cryptographie.

Aujourd'hui l'informatique a révolutionné le monde des réseaux. On peut extrapoler cette idée en se demandant quel sera l'avenir du cryptage sécurisé si un jour, l'informatique venait à être détrôné par une nouvelle technologie ? Les communications numériques cesseront-elles ? Vers quelles formes vont elle tendre ? Car son évolution impactera à jamais le monde la cryptographie….

Annexe:

Algorithmes pour les exemples précédents.

À l'aide d'un langage de programmation, notre objectif est de donner un programme qui avait la capacité de crypter et de décrypter un message en utilisant le protocole d'Algamel, le protocole des codes correcteurs, le protocole des courbes elliptiques.

L'étude des propriétés des courbes elliptique, des fonctions mathématiques et de la programmation a permit d' arriver à un résultat concluant.

Le programme a la capacité de crypter et de décrypter un message de longueur variable.

L'usage que l'on fait aujourd'hui de l'informatique dans nos communications et dans nos transactions bancaires, entre autres, exige un niveau de sécurité de plus en plus élevé. Les paiements de factures à l'aide du réseau Internet constitue un excellent exemple de ce besoin de sécurité. Le but de notre projet était de concevoir un programme informatique capable de crypter et de décrypter un message de longueur variable.

1-Algorithme d'AlgAmel:

Declaration des variables:

p, g, x, y, r, s, k, m, m1, m2, i :des entiers positives.

début

*donner la valeur de p: un nombre premier.

* donner la valeur de g: un générateur du groupe Z/pZ .

*donner la valeur de x: un élément quelconque de ce groupe.

*calculer $y= g^x$ **mod** p.

y= g *g*g…….*g **mod** p; x: fois

*Donc x: la clé secréte de recpteur ;

p, g, y: la clé publique.

Fin.

----------L'émméteur----------

début

* donner la valeur du méssage m.

* donner la valeur de k.

*calculer r, s **mod** p:

r = g*g*g……..*g **mod p** ; k: fois.

s = m.(y*y*y…..*y) **mod p**, k: fois.

*envoyer le chiffré: r et s.

Fin.

----------Le récepteur----------

Début

calcul er:

*m1=r*r*r…….*r mod p; x fois.*

calculer m2 l'inverse de m1:

Début

***Pour** i **de** 1 à (p-1) **faire** :*

Début

***Si:** i*m1 = 1 **mod** p*

***alors**: m2= i;*

***Sinon:** i:=i+1*

fin

fin

calculer

*m= s*m2 =m.*

le même message initiale.

***Fin**.*

1-Algorithme de Mc Elliece:

Declaration des données:

début

p, d, T, k, i, j :des entiers positives.

G[k, n] matrice de k: lignes et n: colonnes.

H[n-k, n] matrice de (n-k): lignes et n: colonnes.

P[n, n] matrice de permutation de n: lignes et n: colonnes.

S[k, k] matrice de k: lignes et k: colonnes.

Pinv[n, n] matrice inverse de P de n: lignes et n: colonnes.

Sinv[k, k] matrice inverse de S de k: lignes et k: colonnes.

G'[k, n] matrice de k: lignes et n: colonnes.

m[k] vecteur de k: composants.

c[n] vecteur de n: composants.

c'[n] vecteur de n: composants.

e[n] vecteur de n: composants.

e'[n] vecteur de n: composants.

m'[k] vecteur de k: composants.

y [n] vecteur de n: composants.

y' [n] vecteur de n: composants.

S[n-k] vecteur de (n-k): composants.

fin

création des clés par le récepteur:

début

*lire p: un nombre premier.

*lire k, n, d

début

lire G[i, j]; pour i=1,….,k et j=1,…,n.

lire S[i, j]; pour i=1,….,k et j=1,…,k

lire P[i, j]; pour i=1,….,n et j=1,…,n.

lire Sinv[i, j]; pour i=1,….,k et j=1,…,k.

lire Pinv[i, j]; pour i=1,….,n et j=1,…,n.

fin

calculer G' ,T:

début

G'=S*G*P; T=[(d-1)/2]

donc la clé secrète est : S, G, P

la clé publique est: G', T, p

fin

fin

----------L'émméteur/chiffrement----------

*soit m[k] le vecteur méssage voulu envoyer par l'émméteur.

*m[k] est de longueur k.

lire m[i] i=1,…,k.

*le chiffrement:

début

lire e[i] , i=1,…,n

* calculer c', c:

c' [n] =m [k] *G' [k, n] .

c [n] =c' [n] +e [n] .

envoyer le chiffré c [n] .

fin

----------Le récepteur/déchiffrement----------

* *calcul er y:* y [n] =c [n] * Pinv[n, n] .

*décoder y en y' suivant un algorithme de décodage:

début

générer la fonction f(e'[xj]), j=1,…,n et xj=1,…,p.

*calculer S[e'] , S[y] :

S(e'[n-k])= H[n-k, n] * e'[xj] , j=1,…,n et xj=1,…,p.

S(y[n-k])= H[n-k, n] * y[n].

si S(e'[n-k])= S(y[n-k]) donc :

y'[n]:= e'[n].

*chercher m'[k]:

y'[n]:= m'[k]*. G[k, n]

calculer : m'[k] Sinv[k, k] = m[k]= message iniciale..

Fin.

1-Algorithme pour le cryptage par courbe elliptique:

Declaration des données:

p, T, T', a, b, m, xp, yp, x2p, y2p, xq, yq, Ka, Kb, xKa, yKa, xKb, yKb, xkaKb, ykaKb, xkbKa, ykbKa, i, j , *β,* xm1, xm2, ym1, ym2,:xm, ym: des entiers positives.

données communes

lire : p, a, b, xp, yp

*calculer l'entier m:

début

pout i=1 à p faire:

si: i*2*yp= 1 **mod** p

alors m=i;

sinon: i:= i+1

fin

*calculer les cordonnées du point 2P :

début

T= [(3*xp*xp + a)*m] **mod** p.

x2p= [(T*T) -2*xp] **mod** p.

y2p= [-(T*T*T) -yp + 3*T*xp] **mod** p.

fin

***création de la clé commune:**

Ou protocole de **Diffie-Hellman**

----------émetteur-récepteur----------

lire Ka; (resp Kb)

**calculer le point Ka*P=(xKa, yKa) (resp : Kb*P=(xKb, yKb))……..@*

début

*Ka*P= 2P +* (Ka -2) *P= 2P+ (P+ P+……..+P); (Ka -2) fois .

calculer l'entier m:

début

pour i=1 à p faire:

si: i*(xp- x2p)= 1 **mod** p

alors m=i;

sinon: i:= i+1

fin

début

pour j=1 à (Ka -2) faire

T'= [(yp- y2p)*m] **mod** p.

x= [(T'*T') – xp –x2p] **mod** p.

y= [-(T'*T'*T') -yp + (2*xp +x2p)*T'] **mod** p.

x:=x2p; y:=y2p

fin

donc: *xKa= x, yKa= y. (resp xKb= x, yKb= y.)*

Fin.

Le récepteur- envoit à l'émméteur le point *Ka*P=(xKa, yKa):*

L'émméteur - envoit au récepteur le point *Kb*P=(xKb, yKb):*

----------émetteur-récepteur----------

Le récepteur- recoit le point *Kb*P=(xKb, yKb):*

L'émméteur - recoit le point *Ka*P=(xKa, yKa):*

* *calculer le point Ka(Kb*P)=(xKaKb, yKaKb) par le récepteur;*

*(resp le point Kb(Ka*P))=(xKbKa, yKbKa) par L'émméteur -).*

début

Ka(Kb*P)= 2(Kb*P)+* (Ka -2) *(*Kb*P)*.

Kb(Ka*P)= 2(Ka*P)+* (Kb -2) *(*Ka*P)*.

***** utiliser le même algorithme précédent @ ;

remplissant le point P par le point (KbP);

(resp remplissant P par (KaP)).

fin

Donc :

*Ka(Kb*P)= Kb(Ka*P)*

soit xKaKb= xKbKa= k.

k est suposé comme une clé secrète commune.

----------L'émméteur/ chiffrement----------

****soit le point M =(xm, ym): le méssage voulu envoyer***

**** lire xm, ym,*** *β ;entiers .*

****calculer les deux points M1 =(xm1, ym1) et M2=(xm2, ym2)***

début

*M1= β*P* (utiliser le même algorithme précédent @) ;

*M2= M + β *k*P =M + M'*

*****utiliser le même algorithme précédent @ pour calculer *β *k*P = M'*;

*utiliser le même algorithme précédent @ pour calculer M +M'.

(remplissant le point P par le point M ; le point 2P par le point M', et répéter

l'opération un seul foi.)

fin

----------Le récepteur/ déchiffrement----------

****recoit les deux points M1 =(xm1, ym1) et M2=(xm2, ym2)***

début

**calculer*

*Msg=M2 – k*M1 = M 2 + k*(-M1) ; (-M1)= - (**xm1, ym1) = (xm1, - ym1)***

Msg *= M 2 + k*M1'=M2 + M1''.*

*****utiliser le même algorithme précédent @ pour calculer *k*M1 = M1'*;

*utiliser le même algorithme précédent @ pour calculer M2 +M1''.

(remplissant le point P par le point M2 ; le point 2P par le point M1'',

et répéter l'opération un seul foi.)

fin

Bibliographie:

[1] B. Benzaghou. Introduction a l'Algèbre linéaire. OPU, 1975.

[2] P-L. CAYREL. Mathématiques et ses applications. Thèse de Doctorat de l'université de LIMOGES, 2008.

[3] J-Y. Enjalbert. Jacobiennes et cryptographie. Thèse de Doctorat de l'université de limoges, 2003.

[4] D-E. Knuth. The art of computer programming, vol.3, sorting and searching, Addison-Wesley. Reading, MA,1973.

[5] D-J. Mercier. Codage et cryptage, APMEP 421 , pages 219-232, 1999.

[6] D-J. Mercier. L'algèbre dans la correction des erreurs, APMEP 415, pages 173-191, 1998.

[7] L. NOUI. Algèbre, Notion fondamentales, Presses de l'université de Batna, 1998.

[8] L. NOUI. Algèbre linéaire, Presses de l'université de Batna, 1999.

[9] R. Schoof. Elliptic curves over finite fields and the computation of square Roots mod p. Mathematics of Computation, Vol. 44, pages 483-494, 1985.

[10] J. Théorie Nombres Bordeaux, 7. Counting points on elliptic Curve over finite fields. Pages 219-254, 1995.

[11] D. Shanks. Five number-theortic algorithms, proceeding of the second manitoba conference on numerical mathematics, pages 51-70, 1972.

[12] M. Zitouni. Géometrie Arithmétique et Algorithmique des courbes elliptiques, OPU 2007.

[13] These results were obtained on a Dell Pentium XPS P90.
Turbo C++ c 1990, 1992, version 3.0

ملخص :

أثناء تبادل المعلومات عبر الوسائل المختلفة(الإنترنيت....)يجب حفظ و حماية هذه المعلومة باستعمال أنظمة التشفير المختلفة. في هذه المذكرة نناقش ثلاثة أنظمة تشفير مختلفة، النظام الأول يعتمد على قوانين الحساب العددي، الثاني على الأكواد والأخير يعتمد على المنحنيات الجبرية و على وجه الخصوص على ما يسمى بالمنحنيات الناقصية المستوية.

Résumé :

On utilise des outils algébriques dans les systèmes cryptographiques pour protéger les informations échangées dans nos communications. Dans ce mémoire on a discuté trois différents cryptosystèmes, le premier basé sur la théorie des nombre et le calcul modulaire, le deuxième sur les codes correcteurs d'erreurs et le dernier basé sur les courbes algébriques en particulier les courbes elliptiques planes.

Abstract:

When we exchange information in our communication, we must protect this information using cryptography systems based on algebraic objects. In this Memory we discuss three cryptosystems, the first is based on the number theory, the second on the errors correcting codes and the last is based on algebraic curves, in particular on planes elliptic curves .

ملخص :

أثناء تبادل المعلومات عبر الوسائل المختلفة (الإنترنيت....) يجب حفظ و حماية هذه المعلومة باستعمال أنظمة التشفير المختلفة.

في هذه المذكرة نناقش ثلاثة أنظمة تشفير مختلفة، النظام الأول يعتمد على قوانين الحساب العددي، الثاني على الأكواد والأخير يعتمد على المنحنيات الجبرية و على وجه الخصوص على ما يسمى بالمنحنيات الناقصية المستوية.

Résumé :

On utilise des outils algébriques dans les systèmes cryptographiques pour protéger les informations échangées dans nos communications.

Dans ce mémoire on a discuté trois différents cryptosystèmes, le premier basé sur la théorie des nombre et le calcul modulaire, le deuxième sur les codes correcteurs d'erreurs et le dernier basé sur les courbes algébriques en particulier les courbes elliptiques planes.

Abstract:

When we exchange information in our communication, we must protect this information using cryptography systems based on algebraic objects.

In this Memory we discuss three cryptosystems, the first is based on the number theory, the second on the errors correcting codes and the last is based on algebraic curves, in particular on planes elliptic curves .