Des Courbes pour le Codage et la Cryptographie

Lancelot Pecquet
Projet CODES
INRIA-Rocquencourt
B.P. 105, 78153 Le Chesnay Cedex
Lancelot.Pecquet@inria.fr

Introduction

Objectif: Protéger l'information contre:

sa détérioration accidentelle	$\longrightarrow$	codes correcteurs d'erreurs
les méchants	$\longrightarrow$	cryptographie

La protection mathématique s'ajoute aux dispositifs de protection physiques.

Codes Correcteurs d'Erreurs

1. Motivation
2. Concept de code, code en bloc, distance minimale
3. Décodage trivial, décodage algébrique: introduction des corps finis
4. Taux de transmission optimalité et théorème de SHANNON
5. Codes faciles à manipuler: les codes linéaires
6. Exemples de codes linéaires
7. Bons codes, décodage et complexité
8. Courbes et codes géométriques de GOPPA
9. Perspectives

Motivation

Pour faire communiquer la sonde *Mars Global Surveyor* avec la Terre, on se heurte à deux problèmes:

- la communication est très bruitée
  $\longrightarrow$ fiabilisation;
- le coût en énergie de la communication doit être minimal
  $\longrightarrow$ efficacité.

Concept de code

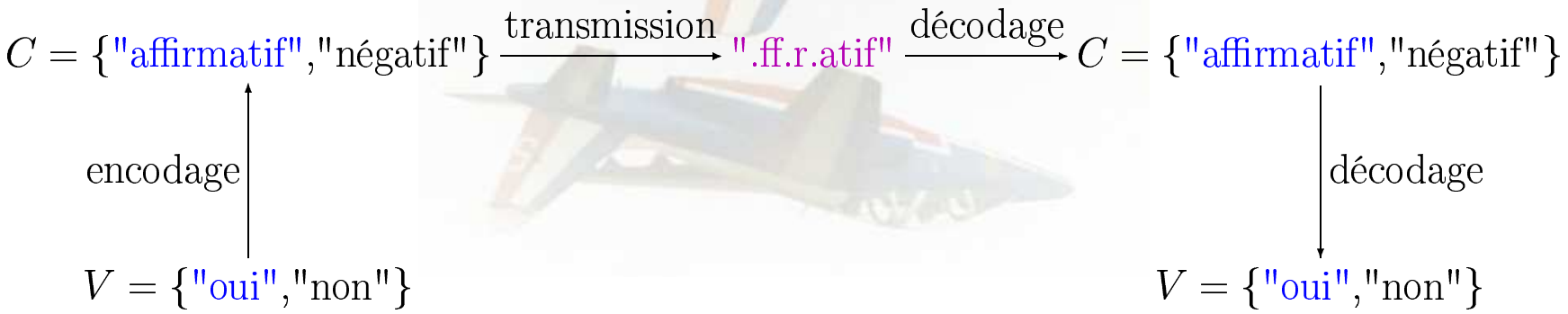
Idée centrale: choisir un vocabulaire adapté à la transmission:

Code = Dictionnaire

Qualités:

- Les mots doivent être très distinguables les uns des autres (résistance au bruit);
- Les mots ne doivent pas être trop longs par rapport à ceux du vocabulaire initial (efficacité).

Application:



Code en bloc, distance minimale

Déf: Un code en bloc de longueur n sur l'alphabet A est un dictionnaire de mots de taille n dont les lettres sont dans A.
Déf: La distance de HAMMING entre deux mots $a = a_1 \cdots a_n$ et $b = b_1 \cdots b_n$ est le nombre d'indices i où $a_i \neq b_i$. On la note $\mathrm{d}(a, b)$. La distance minimale d'un code C est l'entier

$$d = \min_{\substack{a,b \in C \\ a \neq b}} \mathrm{d}(a, b) \ .$$

Ex: Pour $C = \{$"cassis","goyave","mangue","banane"$\}$, on a $n = 6$ et:

d("cassis","goyave")	6
d("cassis","mangue")	5
d("cassis","banane")	5
d("goyave","mangue")	5
d("goyave","banane")	4
d("mangue","banane"	3

La distance minimale de C est $d = 3$.

Décodage: première approche

Le code $C = \{$"cassis","goyave","mangue","banane"$\}$ peut toujours corriger $t = \frac{d-1}{2} = \frac{3-1}{2} = 1$ erreur: c'est la capacité de correction de C.

Une erreur. On reçoit "bangue", les distances avec les mots du code sont: $\longrightarrow$

"cassis"	5
"goyave"	5
"mangue"	1
"banane"	2

$\longrightarrow$ Le mot de code le plus proche susceptible d'avoir été émis est "mangue"

Deux erreurs. On reçoit "bangie", les distances avec les mots du code sont: $\longrightarrow$

"cassis"	4
"goyave"	5
"mangue"	2
"banane"	2

$\longrightarrow$ On ne sait plus décider entre "banane" et "mangue".

Décodage algébrique: introduction des corps finis

Problème: énumération des mots de code impossible dans la pratique (trop de mots).

Solution: utiliser des relations algébriques entre les lettres des mots de code pour calculer les symboles perdus.

Alphabet fini à q lettres avec opérations algébriques $(+, -, \times, /)$: corps fini $\mathbf{F}_q$.

Ex: le corps fini à deux éléments $\mathbf{F}_2 = \{0, 1\}$, avec les opérations suivantes:

$0+0=0$	$0+1=1$	$1+0=1$	$1+1=0$
$0-0=0$	$0-1=1$	$1-0=1$	$1-1=0$
$0\times 0=0$	$0\times 1=0$	$1\times 0=0$	$1\times 1=1$
$0/0$: indéfini	$0/1=0$	$1/0$: indéfini	$1/1=1$

Taux de transmission et optimalite

Soit k la longueur des mots du vocabulaire de départ et n la longueur des mots du code, le taux de transmission du codage est:

$$R = \frac{k}{n} \in [0,1] \ .$$

Ex: Pour $V = \{001, 011, 111\}$ on a $k = 3$. Le code $C = \{000000, 000111, 111000\}$ est de longueur $n = 6$ et de distance minimale $d = 3$. L'encodage se fait par exemple ainsi:

$$001 \longrightarrow 000000$$
$$011 \longrightarrow 000111$$
$$111 \longrightarrow 111000$$

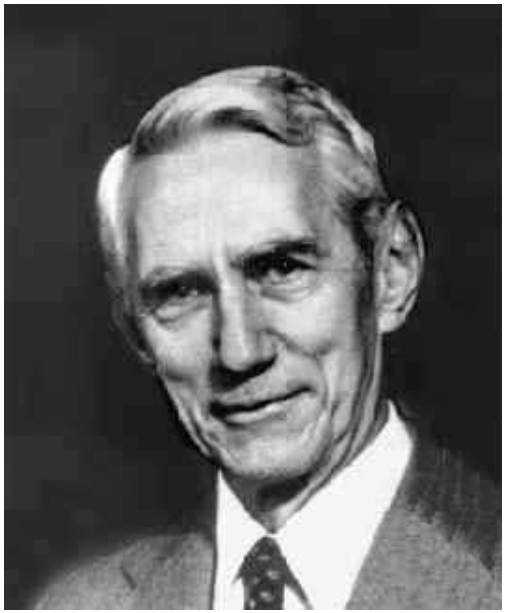
le taux de transmission est $R = k/n = 3/6 = 1/2$.

Un bon codage doit avoir deux qualités simultanées:

- un taux de transmission R *maximal i.e.* un *coût minimal*;
- une distance minimale d *maximale i.e.* une *fiabilité maximale.*

Théorème de SHANNON

Th:(SHANNON, 1948) Si $R > 0$ est inférieur à la capacité du canal de communication, alors pour tout choix de α, il existe un code C de longueur n, corrigeant $\alpha\%$ des erreurs, et une façon de coder tout mot de k symboles en un mot de C tels que le taux de transmission soit $k/n = R$.



En d'autres termes: pour une exigence de fiabilité décidée (α), il existe toujours un (long) code optimal ($R \simeq$ capacité du canal) qui satisfasse cette exigence.

Problème: Construction?

Codes faciles à manipuler: les codes linéaires

On choisit k vecteurs de longueur n, linéairement indépendants $c_1, \ldots, c_k$, dont les coordonnées sont dans $\mathbf{F}_q$. Le code de base $c_1, \ldots, c_k$ est l'ensemble des combinaisons linéaires des mots $c_1, \ldots, c_k$. Le code C se résume à une matrice génératrice:

$$G = \begin{pmatrix} c_{1,1} & \cdots & c_{1,n} \\ \vdots & \ddots & \vdots \\ c_{k,1} & \cdots & c_{k,n} \end{pmatrix}$$

C est le sous-espace vectoriel de $\mathbf{F}_q^n$ engendré par les lignes de G. Il est de dimension k, et contient donc q^k mots. On dit que c'est un $[n,k]$-code (un $[n,k,d]$-code si sa distance minimale est d).

On sait encoder n'importe quel mot de k lettres sur $\mathbf{F}_q$ en un mot de C: on a bien un taux de transmission de $R = k/n$.

Exemple de code linéaire: le $[7,4,3]$-code de HAMMING binaire

Sur $\mathbf{F}_2$, $n = 7$ on peut choisir les $k = 4$ mots:

$$c_1 = 1000011, c_2 = 0100101, c_3 = 0010110, c_4 = 0001111\ ,$$

Dans l'exemple C est constitué des $q^k = 2^4 = 16$ mots:

$$C = \{0000000, 0111100, 0001111, 1110000, 1000011, 1111111, 0101010, 0110011,$$
$$1011010, 1101001, 0011001, 1100110, 0010110, 0100101, 1001100, 1010101\}$$

et une matrice génératrice est:

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

Ici $d = 3$ (atteinte entre 0000000 et 0100101 par exemple): il peut donc corriger $(d-1)/2 = 1$ erreur. Le taux de transmission de C est $R = k/n = 4/7$).

Autre exemple: les codes de REED-SOLOMON

Déf: Soient $p_1, \ldots, p_n$, des éléments distincts de $\mathbf{F}_q$, on appelle code de **Reed-Solomon** de dimension k sur $p_1, \ldots, p_n$, le code linéaire:

$$C = \{(f(p_1), \ldots, f(p_n)) \, , f \in L\} \, , \quad \text{où} \quad L = \{f \in \mathbf{F}_q[X] \mid \deg f < k\} = \langle f_1, \ldots, f_k \rangle \, .$$

Une matrice génératrice est:

$$G = \begin{pmatrix} f_1(p_1) & \cdots & f_1(p_n) \\ \vdots & \ddots & \vdots \\ f_k(p_1) & \cdots & f_k(p_n) \end{pmatrix}$$

Avantages: Excellents (MDS);
Désavantages: Alphabet très grand.

Utilisation: CD, DVD, communications satellites ...
Cas du CD: un $[28, 24, 5]$-code C_1 ($\#C_1 = 256^{24} \simeq 10^{58}$) et un $[32, 28, 5]$-code C_2 ($\#C_2 = 256^{28} \simeq 10^{67}$) sur $\mathbf{F}_{256}$ sont entrelacés et peuvent corriger des bouffées d'erreurs de 4000 bits (2.5 mm).

Bons codes, décodage et complexité

Th:(GILBERT, 1952 - VARSHAMOV, 1957): Sur tout alphabet, il existe de bons codes linéaires: il suffit d'en prendre un au hasard!

Th:(BERLEKAMP,MCELIECE,VANTILBORG, 1978): Si C est un code linéaire sur $\mathbf{F}_q$, et y est un mot sur $\mathbf{F}_q$ de même longueur, trouver un mot de C le plus proche de y (décodage à maximum de vraisemblance) est NP-complet.

Th:(VARDY, 1997): Trouver la distance minimale d'un code linéaire est NP-complet.

Conj: Si C est un code linéaire corrigeant t erreurs sur $\mathbf{F}_q$, et y est un mot sur $\mathbf{F}_q$ de même longueur, trouver, s'il existe, un mot de C à distance plus petite que t de y (décodage borné) est NP-complet.

Nouvelle problématique: Trouver des bons codes que l'on sait décoder, même partiellement.

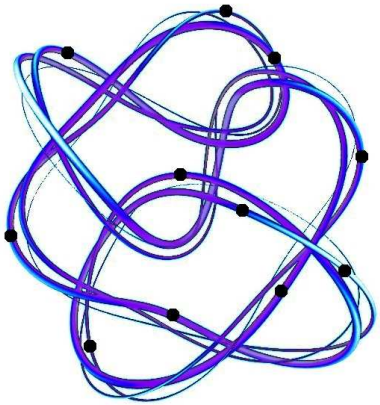
Codes géométriques de GOPPA

Déf: Soit $\mathcal{X}$ une courbe, $P_1, \ldots, P_n$, des points de $\mathcal{X}$ et L un espace vectoriel de fonctions $f : \mathcal{X} \longrightarrow \mathbf{F}_q$ de dimension k, on appelle code géométrique de GOPPA associé le code linéaire:

$$C = \{(f(P_1), \ldots, f(P_n)) \, , f \in L\} \, , \quad \text{où} \quad L = \langle f_1, \ldots, f_k \rangle \, .$$

Une matrice génératrice est:

$$G = \begin{pmatrix} f_1(P_1) & \cdots & f_1(P_n) \\ \vdots & \ddots & \vdots \\ f_k(P_1) & \cdots & f_k(P_n) \end{pmatrix}$$



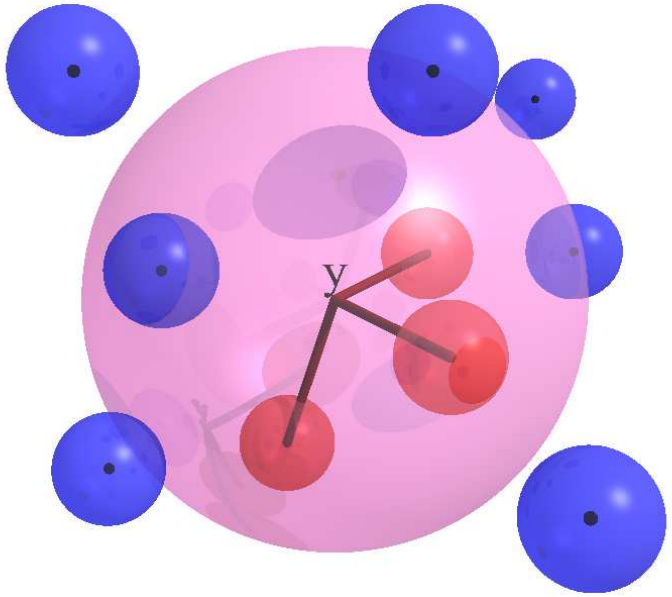
Th:(TFASMANN-VLĂDUŢ-ZINK, 1981) Pour $q > 49$, on sait construire des codes géométriques dépassant la borne de GILBERT-VARSHAMOV.

On sait partiellement les décoder. (HØHOLDT,PELIKAAN,FENG,RAO,DUURSMA, ... $\simeq$ 1995)

Perspectives

Améliorer le temps de construction des codes géométriques (HACHÉ,LAURSEN, 1996);

Travail intensif sur l'algorithme de SUDAN (1998) (*list-decoding* au delà de la capacité de correction!)



Cryptographie

1. Motivation
2. Cryptographie “antique” et moderne
3. Cadre de travail: algorithme publique à clé
4. Fonction à sens unique et cryptographie à clé publique
5. Logarithme discret et échange de clés de DIFFIE/HELLMAN
6. Chiffrement d’ELGAMAL. Exemple des corps finis
7. Courbes elliptiques et logarithme discret
8. Perspectives

Motivation

Protection contre les méchants: (Gargamel, Dark Vador, Bloffeld, Le Grand Stratéguerre, ...)

- Chiffrement;
- Authentification;
- Cryptanalyse.

Cryptographie “antique”

Système de CÉSAR (1e siècle av. J.C.): décalage de trois lettres à droite dans l’alphabet (A⟶C, B⟶D, . . .).

Système de BATTISTA (1568), VIGENÈRE (1586): substitutions polyalphabétiques: choix d’un *mot-clé* et décalage variable en fonction de ce mot; **Ex:** avec le mot clé: "INRIA", pour chiffrer, on décale "I" sur "A" pour chiffrer la première lettre, puis "N" sur "A" . . .

Cryptanalyse: par statistique de fréquences (KASISKI, 1863)

Cryptographie moderne

1883	KERKHOFFS écrit la *Cryptographie militaire*;
1917-26	MAUBORGNE et VERNAM inventent le masque jetable (*one time pad*);
1939-44	TURING décrypte le système allemand *Enigma*;
1949	SHANNON décrit la notion de *sécurité inconditionnelle*;
1973-77	standardisation du DES;
1976	DIFFIE et HELLMAN inventent la cryptographie à clef publique;
1978	Découverte du système RIVEST-SHAMIR-ADELMAN.
1985	ELGAMAL propose un système à clé publique basé sur le logarithme discret;
1985	KOBLITZ et MILLER créent les premiers cryptosystèmes à base de courbes elliptiques

Cadre de travail: algorithme publique à clé

Algorithme public parametré par une clé $\Longrightarrow$ sécurité!

Cryptographie à clé secrète: une clé secrète est partagée par deux personnes
Avantages: algorithmes rapides;
Inconvénients: Il faut s'entendre sur un secret *avant* de communiquer / Il faut autant de secrets que de personnes avec lesquelles on veut communiquer.

Cryptographie à clé publique: une paire de clés sert, l'une à chiffrer, l'autre à déchiffrer.
Avantages: il suffit d'avoir la clé publique de son interlocuteur pour lui envoyer un message chiffré / Permet l'autentification
Inconvénients: Algorithmes lents.

Fonction à sens unique et cryptographie à clé publique

Déf: Une fonction $f : x \longmapsto f(x) = y$ est à sens unique ssi:

- Si x est donné, il est facile de calculer y
- Si y est donné, il est *infaisable* de calculer x.

où *infaisable* signifie irréalisable par un ordinateur (dépend de la technologie). Par exemple s'il faut plus de 2^{64} opérations pour le faire.

Logarithme discret et échange de clés de Diffie/Hellman

Soit un groupe G. Pour un élément $g \in G$ et un entier $x \in \mathbf{N}$ donné, il est souvent facile de trouver $y = g^x$. Inversement, le problème du logarithme discret de base g dans G consiste, étant donné $y \in G$, à trouver, s'il existe, un entier $x \in \mathbf{N}$ tel que $y = g^x$.

La clé publique est $\gamma \in G$.

1. Alice choisit en secret $k_A \in \mathbf{N}$, calcule γ^{k_A}, et l'envoie à Bob;
2. Bob choisit en secret $k_B \in \mathbf{N}$, calcule γ^{k_B} et l'envoie à Alice;
3. Alice reçoit γ^{k_B} et peut calculer $(\gamma^{k_B})^{k_A}$;
4. Bob reçoit γ^{k_A} et peut calculer $(\gamma^{k_A})^{k_B}$;

Alice et Bob partagent le secret: $\gamma^{k_A k_B}$.

Le décryptement peut se faire si, connaissant γ, γ^{k_A} et γ^{k_B}, on peut trouver $\gamma^{k_A k_B}$. C'est en particulier le cas si on sait calculer le logarithme en base γ. (La réciproque est conjecturée.)

Chiffrement d'ElGamal. Exemple des corps finis

Soit $m \in G$ le message à chiffrer, Alice et Bob ont choisi une clé de session avec le protocole de Diffie-Hellman.

1. Bob choisit en secret $l \in \mathbf{N}$ et envoie à Alice $a = \gamma^l$ et $b = m\gamma^{lk_A}$;
2. Alice multiplie b à droite par $a^{-k_A} = \gamma^{-lk_A}$ et obtient le clair:

$$(m\gamma^{lk_A}) \cdot \gamma^{-lk_A} = m \ .$$

Dans le cas où $G = \mathbf{F}_q^*$, le meilleur calcul de log discret se fait avec le crible de corps de nombres (Lenstra×2, 1993 Gordon, 1995) en

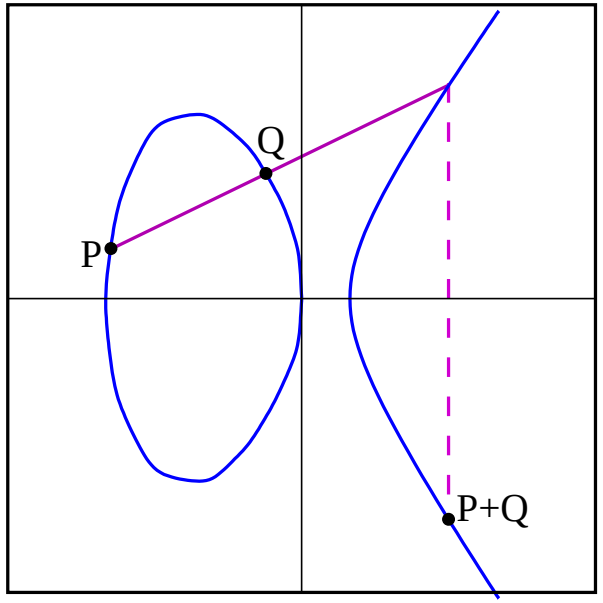
$$\exp\left(\mathrm{O}\left(\sqrt[3]{\log q \log\log^2 q}\right)\right) \ .$$

En 1998, Joux et Lercier ont calculé un log discret dans $\mathbf{F}_p^*$ avec p comprenant 90 chiffres décimaux.

Courbes elliptiques et logarithme discret

Une courbe elliptique est une courbe plane cubique définie par une équation du type:

$$Y^2 + a_1XY + a_3Y - (X^3 + a_2X^2 + a_4X + a_6) = 0$$



La courbe a une structure de groupe abélien: généralisation des cryptosystèmes à logarithme discret (Diffie-Hellman, ElGamal)

Avantage: Choix plus grand dans le groupe du log / absence d'algorithme sous-exponentiel connu pour le log discret.
Utilisation réelle: IEEE P1363 (Échange de clés de sessions, 1998), ANSI X9.62 et ISO/IEC 14888 (Signature Electronique, 1998), ...

Perspectives

Danger? MENEZES,OKAMOTO,VANSTONE 1993 utilisent le couplage de WEIL et réduisent le logarithme discret sur une courbe elliptique définie sur $\mathbf{F}_q$ au logarithme discret dans une extension $\mathbf{F}_{q^k}$. Cela dit seule une petite classe de courbes — les supersingulières — admettent une réduction avec k petit et l'attaque ne semble pas fonctionner ailleurs (BALASUBRAMANIAN-KOBLITZ, 1998).

Idée de sécurisation: Travail dans la jacobienne des courbes hyperelliptiques dans laquelle les calculs se font plus vite que pour une courbe quelconque. Si J_q est la jacobienne d'une courbe hyperelliptique définie sur $\mathbf{F}_q$, et que $|J_q|$ est divisible par un nombre premier ℓ d'au moins 40 chiffres décimaux, ne divisant pas $q^k - 1$ pour tous les k petits ($\leq 2000/\log_2 q$), alors le système résiste à l'attaque de FREY-RÜCK (1994) qui généralise l'attaque de MOV.

Conclusions

Les courbes sont des objets dont la connaissance sert les développements récents du codage et de la cryptographie.

Les propriétés intéressantes de ces courbes sont à la fois d'ordre théoriques (*e.g.* nombre de points rationnels) et pratiques (*e.g.* algorithmes de calcul rapide).