Utilisation de l'algèbre dans les systèmes d'information

Dany-Jack Mercier

Utilisation de l'algèbre dans les systèmes d'information.

Dany-Jack Mercier

IUFM de Guadeloupe, Morne Ferret,
BP517, Abymes, cedex 97178, France
dany-jack.mercier@univ-ag.fr

2 juin 2000

Résumé – L'apprentissage de l'algèbre et de l'arithmétique en lycée et dans les premières années d'université peut être considéré comme inutile par de nombreux étudiants. Et pourtant les congruences, le Théorème de Bezout, les espaces vectoriels, les matrices et les polynômes sont devenues inséparables de nos techniques d'information. Nous les utilisons quotidiennement lorsque nous écoutons un CD Audio ou dès que nous tapons le code de notre carte bancaire sur le clavier d'un distributeur de billets. Le but de cet exposé est de raconter deux applications récentes de l'algèbre : les codes linéaires (et à travers eux les codes BCH) et le désormais célèbre système de chiffrement à clés publiques RSA.

1 Introduction

L'algèbre et l'arithmétique peuvent apparaître comme des constructions théoriques, gratuites et sans aucun impact dans la vie quotidienne. Elles peuvent aussi être considérées comme un passage obligé permettant d'acquérir une certaine maîtrise calculatoire qui sera réinvestie dans d'autres domaines mathématiques ou dans les sciences expérimentales.

- Dès le collège on utilise des rudiments d'algèbre pour calculer des distances dans des problèmes de géométrie, et pour résoudre des équations du premier

[0][ccod0013] v1.05
In *Actes du 5e Colloque de l'IREM des Antilles-Guyane*, du 31 mai 2000 au 3 juin 2000

degré. La réponse à des problèmes pratiques donnée par le choix des inconnues, la mise en équation et la résolution de ces équations, représente une application remarquable de l'algèbre pendant toutes les études du secondaire.

- Il est important de savoir résoudre des systèmes linéaires dans pratiquement toutes les sciences expérimentales, et cela justifie l'apprentissage de la méthode du pivot de Gauss dès le lycée. Par ailleurs, en Astronomie, la prédiction des éclipses représente une application historique du Théorème des restes chinois, et par conséquent du fameux Théorème de Bezout et de l'arithmétique.

- On peut rappeler l'importance du Théorème de Pythagore en architecture. La méthode de la corde à 12 noeuds était utilisée par les bâtisseurs babyloniens et égyptiens pour obtenir des alignements et des angles droits parfaits sur le terrain. Le Théorème de Pythagore, à la fois géométrique et algébrique, trouve ici une application fondamentale.

- Le lien entre l'harmonie architecturale et le célèbre nombre d'or :

$$\varphi = \frac{1+\sqrt{5}}{2}$$

était accepté des anciens qui accordaient tout aussi bien une valeur métaphysique à ce nombre, compte tenu de sa nature irrationnelle. Mais si le nombre d'or est traditionnellement affublé d'un caractère esthétique universel (La Divina Proportione, de Luca Pacioli), cette harmonie peut être refusée : si l'on s'accorde à trouver beau le pentagone régulier étoilé, certains, comme l'architecte-esthéticien Borissavliévitch, considèrent que le pentagone régulier convexe est « lourd et n'a absolument rien à voir avec la proportion et l'harmonie architecturale » ([5] p.26).

- Les devinettes arithmétiques liées aux équations diophantiennes ont joué un rôle important dans le passé et peuvent toujours être utilisées pour mettre en oeuvre des résultats importants tels le Théorème de Gauss ou le Théorème Fondamental de l'Arithmétique. On pourra penser aux problèmes anciens suivants relevés dans le livre de Jean Itard [8] pp. 25-26 :

Problème (Pacioli-Tartaglia-Bachet) : « Il y a 41 personnes en un banquet tant hommes que femmes et enfants, qui en tout dépensent 40 sous, mais chaque homme paie 4 sous, chaque femme 3 sous, chaque enfant 4 deniers (il y a 12 deniers dans un sou). Je demande combien il y a d'hommes, combien de femmes, combien d'enfants ».

Problème (Euler) : « Quelqu'un achète des chevaux et des boeufs ; il paie 31 écus par cheval et 20 écus pour chaque boeuf, et il se trouve que les boeufs lui ont coûté 7 écus de plus que ne lui ont coûté les chevaux : combien cet homme a-t-il acheté de boeufs et de chevaux ? ».

Pour un mathématicien, l'intérêt de l'arithmétique et de l'algèbre ne fait aucun doute. Mais que répondre à un élève de terminale qui demande à quoi peuvent bien servir les polynômes dans la vie de tous les jours ? Et les congruences ? Que répondre à un étudiant de DEUG qui doute sincèrement de l'intérêt d'un apprentissage systématique des matrices et de l'algèbre linéaire, du Théorème de Bezout et de l'indicateur d'Euler ?

Il y a peu de temps, j'ai rencontré une personne qui devait apprendre à multiplier des matrices entre elles et à les inverser. Cette personne préparait un concours interne d'Inspecteur des Impôts, et était persuadée que les matrices n'avaient été inventées que par des matheux désireux de créer des concepts abstraits sans aucun lien avec le réel. L'algèbre apparaît souvent comme une construction gratuite, et la Théorie des Nombres comme un « modèle même d'inutilité » comme le dit M. Demazure dans l'introduction de son Cours d'Algèbre [7].

Mais où vivons-nous ? Qu'utilisons-nous constamment ? Chaque jour, nous écoutons des disques compacts et nous utilisons des cartes bancaires. Et bien l'algèbre est tout à fait présente dans ces instruments : en fait, nous utilisons quotidiennement, sans le savoir, des matrices, des polynômes et des espaces vectoriels pour entendre de la musique. Notre lecteur de salon pourra restituer le morceau de musique malgré les quelques petites rayures sur la surface du CD, le baladeur fonctionnera malgré quelques petits chocs, et nos cartes magnétiques protégeront les informations qu'elles contiennent en faisant appel à des techniques arithmétiques de chiffrement des données. Loin d'être obsolète, l'arithmétique devient inhérente à une grande diversité de systèmes d'information.

Et la tendance n'est pas prête de s'inverser... L'information doit être conservée, transmise et protégée. Cette information devient entièrement numérique : les textes, les images, les sons, les vidéos se résument à une succession des bits (symboles 0 ou 1). Le livre devient électronique avec l'e-book américain et le Cytale français. Tout récemment, le projet PersonaForm, imaginé par l'américain Gregory Peter Panos, se propose de numériser l'individu sous toutes ses coutures pour en générer un double parfait, et ceci en utilisant les technologies d'acquisition les plus sophistiquées (scanning 3D du corps, échantillonnage numérique des expressions faciales, étude des mouvements...). Dans ce cas, le but est de créer et de conserver une trace cohérente d'une personne, et les applications seront nombreuses (applications thérapeutiques pour aider les victimes de traumatisme à reconstruire leur identité à partir de leur archive, création de simulacres qui évolueront dans des espaces virtuels, créations d'hologrammes...) [17].

Décidément, nous entrons dans une nouvelle ère. En suivant Pierre Levy, nous

pouvons distinguer trois âges de l'humanité : celui de l'Oral, celui de l'Ecrit et celui de l'Information ([15] p. 23). L'informatique est régie par des nombres, c'est le domaine du numérique, et les nombres sont étudiés en mathématiques.

Dans cet article, je voudrais motiver les apprentissages abstraits de l'arithmétique et de l'algèbre en montrant les deux domaines d'intervention importants que sont le codage et le cryptage. La Fig. 1 montre les trois grands risques qui concernent un système d'information et la place du codage et du cryptage dans ces risques.

De façon sommaire, on dira que le codage permet de protéger un message contre une perte de donnée due à sa transmission, à la destruction partielle du support de l'information, ou à tout autre cause. Il doit permettre de retrouver des données exactes à partir de données erronées. Le but du cryptage, quant à lui, est de transformer un message en une suite incompréhensible de symboles pour une tierce personne non autorisée. Le cryptage assure la confidentialité d'une communication et permet de signer des messages.

2 Codage

Si nous envoyons un message par un canal de transmission, par exemple le mot 01101 formé de 5 bits valant chacun 0 ou 1, le message obtenu à l'arrivée pourra contenir quelques erreurs. Décidons maintenant de répéter ce message trois fois, et d'envoyer le mot :

011010110101101.

Dans ce cas, une erreur pourra être corrigée et deux erreurs pourront être détectées. Ce code très simple qui consiste à répéter un certain nombre de fois le message à envoyer est appelé **code à répétition**. Il nous montre déjà que, pour protéger l'information, il faut rajouter des caractères redondants et augmenter le poids de cette information.

Le code à répétition est utilisé dans certains lecteurs de CD Audio qui possèdent trois têtes de lecture. Dans ce cas, le signal 0 ou 1 est lu indépendamment par chacune des trois têtes et une erreur de lecture peut être corrigée.

Un deuxième exemple de détection d'erreurs très utilisé en informatique est l'adjonction d'un **bit de parité**. Le message 01101 est alors transformé en 011011 où le dernier bit est égal à la somme de tous les bits du message originel. Il suffit alors de sommer tous les bits du message reçu pour obtenir 1 si une erreur s'est produite, et 0 dans le cas contraire. Ce code, appelé **code de parité**, détecte une erreur sans pouvoir la corriger.

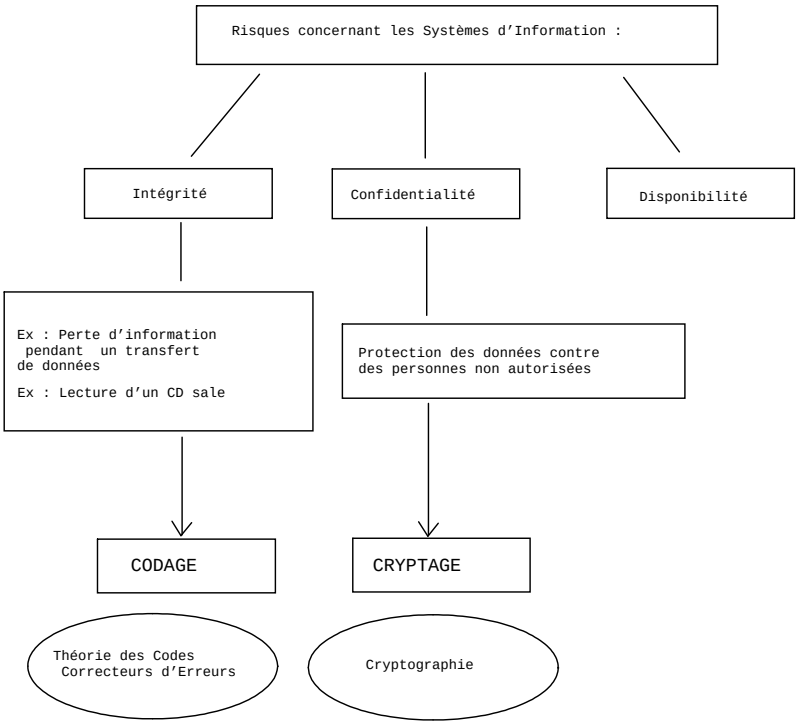


Fig. 1

Ces deux exemples contiennent en germe toute la Théorie des Codes. L'idée fondamentale est de s'apercevoir que l'on peut corriger un message en l'allongeant volontairement avant sa transmission ou sa lecture.

Une Théorie des Codes devra construire des outils qui permettent de :

- savoir si des erreurs se sont produites (problème de la détection),
- retrouver le message correct à partir du message reçu (problème de la correction),
- corriger le plus d'erreurs possible en utilisant le moins de bits supplémentaires possible (problème de la performance du codage).

Achevons ce paragraphe par une visualisation amusante et tout de même assez proche de la réalité. Imaginons un champs de tir où toutes les cibles seraient régulièrement disposées sur un cercle dont le centre serait occupé par un tireur (Fig. 2). Supposons que le tireur vise une cible A (A désigne le centre de la cible sur la circonférence). Notons d la distance curviligne entre les centres de deux cibles consécutives, et supposons que la cible A soit située entre les deux cibles B et C. Notons X le point d'impact de la balle sur le cercle.

• Si X est à une distance curviligne de A strictement inférieur à $d/2$, alors on peut raisonnablement en déduire que le tireur visait la cible A, mais on n'en sera jamais vraiment sûrs (décodage par maximum de vraisemblance).

• Si notre tireur est assez sûr de lui pour ne jamais envoyer sa balle à une distance supérieure ou égale à $d/2$ mètres de son objectif, alors on pourra toujours retrouver la cible qu'il visait en observant l'impact X. L'erreur peut être détectée et corrigée.

• Il sera par contre impossible de déterminer la cible qui était visée si notre tireur est parfois capable d'envoyer sa balle à une distance supérieure ou égale à $d/2$ de son objectif.

• Enfin, si le tireur est certain de toujours placer son point d'impact X à une distance strictement inférieur à d de son objectif, on pourra toujours détecter une erreur de tir mais sans pouvoir la corriger (c'est-à-dire sans pouvoir déterminer la cible qui était visée).

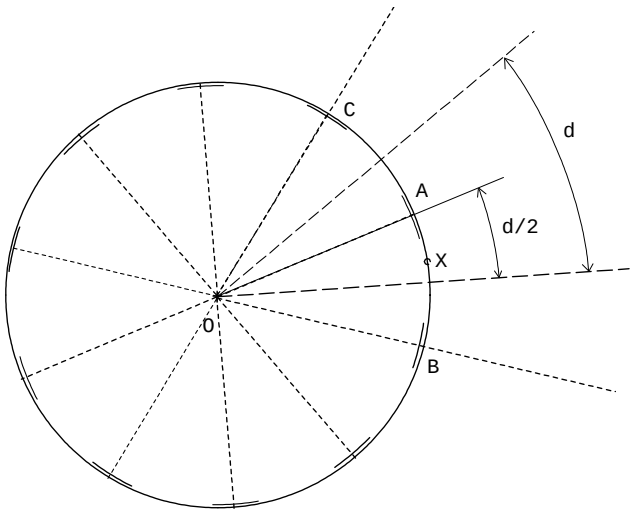


Fig. 2

Nous devrons travailler dans des espaces discrets, et l'adaptation faite à la Figure 3 est encore plus réaliste. Les cibles sont remplacées par des points, et les seuls points d'impact possibles sont les points-cibles A, B... et les points a, b, c... marqués sur la circonférence.

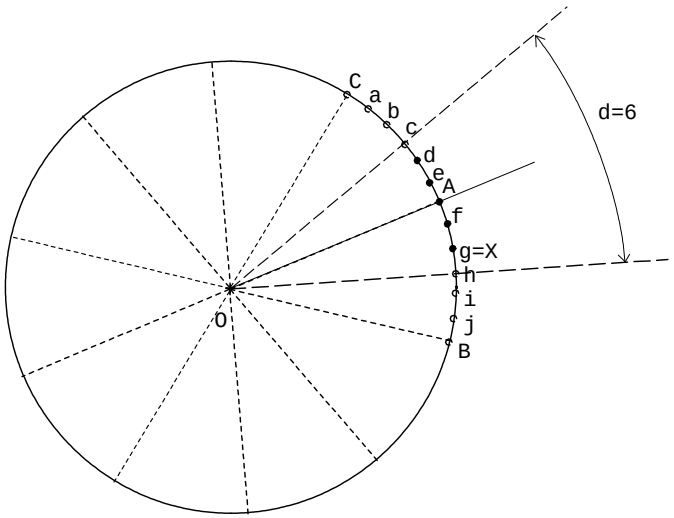


Fig. 3

Dans ce cas, la correction du tir est possible si et seulement si notre tireur est suffisamment régulier pour toujours envoyer sa balle à une distance inférieure ou égale à $\left[\frac{d-1}{2}\right]$ de son objectif ($[t]$ désigne la partie entière de t).

3 Distance de Hamming et distance minimale d'un code

Soit Q un ensemble fini à q éléments. Soient k et n deux entiers naturels non nuls avec $k \leq n$. L'ensemble des messages sera une partie E de Q^k, et l'on introduit une application injective :

$$\begin{array}{rccc} f: & E & \to & Q^n \\ & a = (a_1, a_2, ..., a_k) & \mapsto & c = (c_1, c_2, ..., c_n) \end{array}$$

appelée **application de codage** ou **encodeur**. Le message ou mot a est un élément de E. Il est modifié pour fournir le mot $f(a) = c \in Q^n$. C'est le mot c qui sera transmis et lu par un système quelconque pour donner un message reçu $x = (x_1, ..., x_n)$ qui contient éventuellement quelques erreurs.

Notons $C = f(E)$ l'image de f. Comme f est injective, f réalise une bijection de E sur C et C peut être considéré comme l'ensemble de tous les messages possibles. C est appelé **code de longueur** n, et les éléments de C s'appellent les **mots** du code. Le **cardinal** du code est par définition celui de C.

La **distance de Hamming** d définie par :

$$\forall x, y \in Q^n \quad d(x, y) = \#\{i \in \mathbb{N}_n \,/\, x_i \neq y_i\}$$

permet de mesurer le degré de différence entre deux mots x et y de Q^n. On peut vérifier que d satisfait bien les axiomes classiques d'une distance.

La **distance minimale** du code C est la distance minimum entre deux mots distincts de ce code. On la note :

$$d = \text{Min}\left\{d\left(x,y\right) \ / x,y \in C \text{ et } x \neq y\right\}.$$

Un code C de longueur n, de cardinal M et de distance minimale d est appelé code $[n, M, d]$. Les nombres n, M, d sont les **paramètres** du code.

La distance minimale d permet d'obtenir le nombre maximum d'erreurs que le code peut corriger. Si le message $c = (c_1, ..., c_n)$ a été envoyé avec moins de t erreurs de transmission, le message obtenu $x = (x_1, ..., x_n)$ vérifie $d\left(x, c\right) \leq t$. Ainsi l'on peut retrouver c à partir de x si, et seulement si, il existe un unique mot de code situé à une distance de x inférieure ou égale à t. Cela revient à dire que les boules fermées de rayon t centrées sur les éléments du code C soient disjointes. Un **code corrigera** t **erreurs** si cette condition est vérifiée.

Théorème 1 *Un code C de distance minimale d corrige au plus $e = \left[\frac{d-1}{2}\right]$ erreurs et en détecte $d-1$.*

Preuve : Le code C ne corrige pas t erreurs si et seulement si :

$$\exists x \in Q^n \quad \exists c, c' \in C \quad c \neq c' \quad d\left(x, c\right) \leq t \text{ et } d\left(x, c'\right) \leq t \quad (1)$$

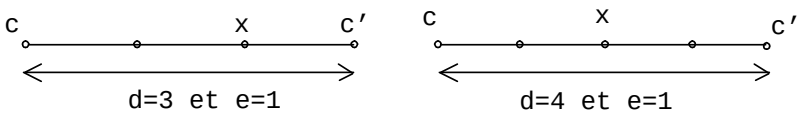
et cela entraîne $d \leq d\left(c, c'\right) \leq d\left(c, x\right) + d\left(x, c'\right) \leq 2t$, soit :

$$d \leq 2t. \quad (2)$$

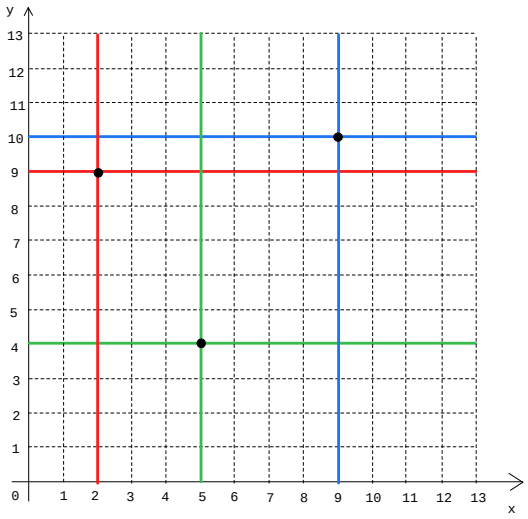
La réciproque est vraie. En effet, si (2) est vérifiée, on peut toujours trouver deux mots de code c et c' situés à la distance d l'un de l'autre, et les noter $c = (c_1, ..., c_n)$ et
$c' = (c'_1, ..., c'_d, c_{d+1}, ..., c_n)$ quitte à permuter les coordonnées. Il existe deux entiers naturels p et q inférieurs ou égaux à t tels que $d = p + q \leq 2t$, et le mot :

$$x = \left(c'_1, ..., c'_p, c_{p+1}, ..., c_{p+q}, c_{d+1}, ..., c_n\right)$$

vérifie bien $d\left(x, c\right) = p \leq t$ et $d\left(x, c'\right) = q \leq t$. Cela prouve (1). En conclusion C corrige t erreurs si et seulement si $2t < d$, et cela équivaut à $t \leq \left[\frac{d-1}{2}\right]$. ■

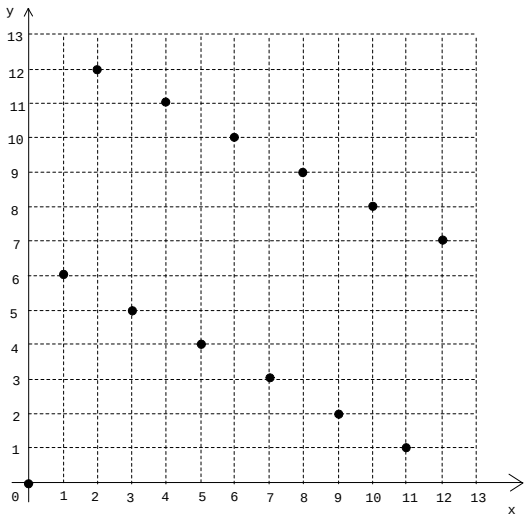


Exemple 1 – Dans $\mathbb{F}_{13}^2$, le code C formé des 3 points ci-dessous est de distance minimal 2. Les boules fermées de rayon 1 centrées sur les mots du code sont représentées en couleur. Ce code détecte une erreur sans pouvoir la corriger.



Exemple 2 – Dans $\mathbb{F}_{13}^2$, la droite vectorielle $5x - 3y = 0$ détermine un code linéaire C'. Sa distance minimale est encore 2. Il est facile de savoir si un mot appartient ou non à C', et d'exhiber l'encodage :

$$\begin{array}{rccc} f: & \mathbb{F}_{13} & \longrightarrow & \mathbb{F}_{13}^2 \\ & t & \longmapsto & (3t, 5t)\,. \end{array}$$



4 Codes linéaires

Si q est une puissance non nulle d'un nombre premier, on sait qu'il existe à isomorphisme près un unique corps fini $\mathbb{F}_q$ de cardinal q ([9], [16]). Choisissons $E = \mathbb{F}_q^k$ comme ensemble des messages. L'ensemble E devient maintenant un espace vectoriel de dimension k sur $\mathbb{F}_q$, et il est naturel de ne considérer que les fonctions d'encodage f linéaires. Le code $C = f\left(\mathbb{F}_q^k\right)$ est alors structuré en sous-espace vectoriel de $\mathbb{F}_q^n$.

Définition 1 *Un* ***code linéaire*** *de dimension k et de longueur n sur $\mathbb{F}_q$ est un sous-espace vectoriel de dimension k de $\mathbb{F}_q^n$. Si la distance minimal de C est d, on dit que C est un code de paramètres $[n, k, d]$ (ou simplement $[n, k]$). Si $q = 2$, on dit que C est un* ***code binaire****.*

Définition 2 *Le* ***poids de Hamming*** *d'un mot x de $\mathbb{F}_q^n$ est, par définition, le nombre de coordonnées non nulles de ce mot. On le note :*

$$w(x) = \#\left\{i \in \mathbb{N}_n \,/\, x_i \neq 0\right\}.$$

Comme $d(x, y) = w(x - y)$, la distance minimale de C s'écrit :

$$d = \operatorname{Min}\left\{w(x) \,/\, x \in C^*\right\}.$$

L'application linéaire $f : \mathbb{F}_q^k \to \mathbb{F}_q^n$ possède une matrice que l'on notera ${}^tG$ dans les bases canoniques. Ici l'écriture ${}^tG$ désigne la transposée de la matrice G. La matrice G possèdera k lignes et n colonnes, et tout mot de C s'écrira sous la forme $c = f(x) = xG$ où $c = (c_1, ..., c_n) \in \mathbb{F}_q^n$ et où $x = (x_1, ..., x_k) \in \mathbb{F}_q^k$ sont des vecteurs-lignes.

Définition 3 *Une* ***matrice génératrice*** *du code C est une matrice G de taille $k \times n$ telle que :*

$$C = \left\{c \in \mathbb{F}_q^n \,/\, \exists x \in \mathbb{F}_q^k \quad c = xG\right\}.$$

Toute matrice H de taille $(n - k) \times n$ vérifiant :

$$c \in C \Leftrightarrow H\,{}^tc = 0$$

est appelée ***matrice de contrôle*** *de C. On a $C = \operatorname{Ker} H$ et $\operatorname{rg} H = n - k$. Toute application linéaire $S : \mathbb{F}_q^n \to \mathbb{F}_q^m$ (où $m \in \mathbb{N}$) de noyau C est appelée* ***syndrome*** *de C. En particulier, $S(x) = H\,{}^tx$ définit un syndrome de C.*

5 Codes systématiques

A chaque mot $x = (x_1, ..., x_k)$ du message on adjoint $n-k$ symboles $c_{k+1}, ..., c_n$ dépendant linéairement des x_i pour obtenir le mot de code $c = f(x)$. Les symboles ajoutés sont appelés **bits de contrôle**. On a :

$$c = (x_1, ..., x_k, c_{k+1}, ..., c_n) = (x_1, ..., x_k)(I_k \mid A)$$

où $(I_k \mid A)$ désigne la matrice $k \times n$ obtenue en écrivant côte à côte la matrice identité I_k de taille k et une matrice quelconque A. Un code C sera dit **systématique** s'il possède une matrice génératrice de la forme $G = (I_k \mid A)$. Comme :

$$c \in C \Leftrightarrow \exists x \in \mathbb{F}_q^k \quad c = xG = x(I_k \mid A)$$

on aura :

$$c \in C \Rightarrow \left(-{}^tA \mid I_{n-k}\right) {}^tc = \left(-{}^tA \mid I_{n-k}\right) \begin{pmatrix} I_k \\ {}^tA \end{pmatrix} {}^tx = -{}^tA\,{}^tx + {}^tA\,{}^tx = 0.$$

Autrement dit C est inclus dans le noyau de l'application linéaire de matrice $H = \left(-{}^tA \mid I_{n-k}\right)$. On notera $C \subset \operatorname{Ker} H$. Comme H est une matrice de rang $n-k$, on aura $\dim C = \dim \operatorname{Ker} H = k$ et $C = \operatorname{Ker} H$. On vient donc de montrer que la matrice $H = \left(-{}^tA \mid I_{n-k}\right)$ est une matrice de contrôle de C. Donnons quelques exemples de codes binaires systématiques :

a) L'encodage $f(x_1, x_2, x_3) = (x_1, x_2, x_3, x_1 + x_3, x_2 + x_3, x_1 + x_2 + x_3, x_3)$ définit un code systématique C de type $[7, 3]$ sur $\mathbb{F}_2$. L'écriture :

$$(c_1, ..., c_6) = (x_1, x_2, x_3) \left(\begin{array}{ccc|cccc} 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{array}\right)$$

met en évidence une matrice génératrice de C, et l'on déduit la matrice de contrôle :

$$H = \left(\begin{array}{ccc|cccc} 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{array}\right).$$

b) La matrice génératrice G du **code de parité binaire** sera :

$$G = \left(\begin{array}{ccc|c} 1 & & 0 & 1 \\ & \ddots & & \vdots \\ 0 & & 1 & 1 \end{array}\right)$$

puisque l'encodage est :

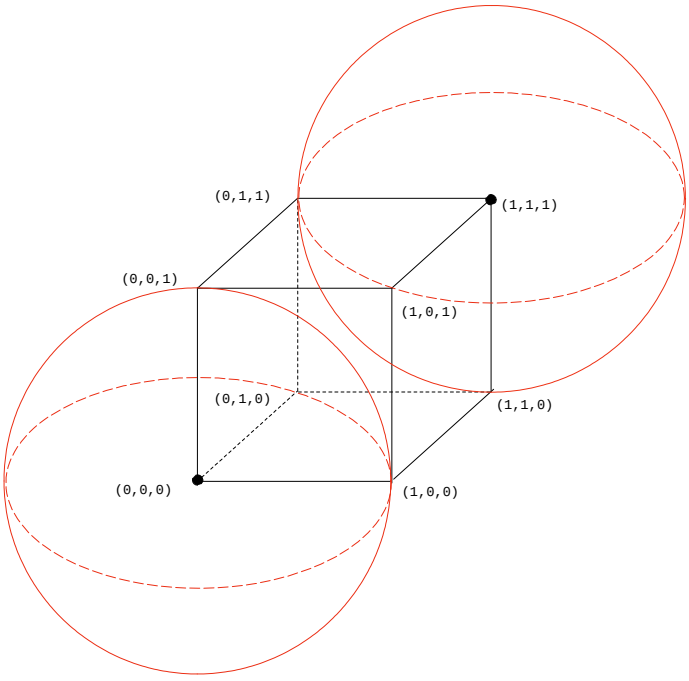
$$(x_1, ..., x_k) \longmapsto (x_1, ..., x_k) \left(\begin{array}{ccc|c} 1 & & 0 & 1 \\ & \ddots & & \vdots \\ 0 & & 1 & 1 \end{array}\right) = \left(x_1, ..., x_k, \sum_{i=1}^{k} x_i\right).$$

La matrice de contrôle H associée est $H = \begin{pmatrix} 1 & \dots & 1 \end{pmatrix}$.

c) Code à répétition : répétons n fois le symbole x_1. On obtient un code $[n, 1]$ pour lequel :

$$G = \begin{pmatrix} 1 & \dots & 1 \end{pmatrix} \quad \text{et} \quad H = \left(\begin{array}{c|cc} 1 & 1 & 0 \\ \vdots & & \ddots \\ 1 & 0 & 1 \end{array}\right).$$

Le code $\{(0,0,0), (1,1,1)\}$ obtenu pour $n = 3$ a été dessiné ci-dessous. De distance minimale 3, ce code corrige une erreur et les boules fermées de rayon 1 centrées sur les mots de code recouvrent $\mathbb{F}_2^3$. On dit que ce code est **parfait**.



6 Codes MDS

Théorème 2 *La distance minimale d'un code linéaire de matrice de contrôle H est égale au nombre minimum de colonnes de H linéairement dépendantes. Autrement dit :*

$$d = \mathrm{Min}\left\{s \in \mathbb{N}^* \,/\, \textit{il existe } s \textit{ colonnes de } H \textit{ linéairement dépendantes}\right\}.$$

Preuve — Soit C un code de paramètres $[n,k,d]$ et de matrice de contrôle $H = [h_1, ..., h_n]$, où h_i désigne la i-ième colonne de H. Le Théorème provient des équivalences :

$$\begin{aligned} x = (x_1, ..., x_n) \in C &\Leftrightarrow H\,{}^t x = 0 \\ &\Leftrightarrow [h_1, ..., h_n] \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} = 0 \\ &\Leftrightarrow \sum_{i=1}^{n} x_i h_i = 0. \end{aligned}$$

Si $x = (x_1, ..., x_n)$ représente un mot de code de poids d, et la relation $\sum_{i=1}^{n} x_i h_i = 0$ montre une relation de dépendance d'exactement d colonnes de H. Il existera donc d colonnes de H linéairement dépendantes. D'autre part, si les s colonnes $h_{i_1}, ..., h_{i_s}$ de H sont linéairement dépendantes, il existe une s-liste $(x_{i_1}, ..., x_{i_s}) \neq (0, ..., 0)$ telle que $\sum_{j=1}^{s} x_{i_j} h_{i_j} = 0$. En posant $x_i = 0$ si $i \notin \{i_1, ..., i_s\}$ et $x = (x_1, ..., x_n)$, on constate que $x \in C$ et $w(x) \leq s$. Cela entraîne $d \leq s$. ■

Corollaire 1 *Les paramètres d'un code vérifient toujours l'inégalité :*

$$k + d \leq n + 1.$$

Preuve — La matrice H est de rang $n-k$, donc $n-k+1$ de ses colonnes seront toujours liées et le Théorème 2 entraîne $d \leq n-k+1$. ■

Définition 4 *La majoration $d \leq n-k+1$ est appelée **borne de Singleton**. Un **code MDS** (Maximum Distance Separable) est un code $[n,k,d]$ tel que $d = n-k+1$.*

Corollaire 2 *Le code C est un code MDS si, et seulement si, $n-k$ colonnes quelconques d'une de ses matrices de contrôle H sont toujours linéairement indépendantes.*

Voici trois exemples de codes MDS :

a) Le sous-espace vectoriel C engendré par un vecteur $a \in \mathbb{F}_q^n$ dont toutes les coordonnées sont non nulles est un code MDS de paramètres $[n, 1, n]$,

b) L'hyperplan d'équation $x_1 + ... + x_n = 0$ est un code MDS de paramètres $[n, n-1, 2]$,

c) Un code de Reed-Solomon (voir plus loin) est un code MDS. C'est le code utilisé dans le minitel.

7 Codes cycliques

Définition 5 *Un code linéaire C est **cyclique** s'il est stable par permutation à droite de ses composantes, c'est-à-dire s'il vérifie :*

$$(a_0, ..., a_{n-2}, a_{n-1}) \in C \Rightarrow (a_{n-1}, a_0, ..., a_{n-2}) \in C.$$

Notons $\mathbb{F}_q[x]$ l'algèbre des polynômes à coefficients dans le corps de Galois $\mathbb{F}_q$, et $(x^n - 1)$ l'idéal engendré par le polynôme $x^n - 1$. L'algèbre quotient $\mathbb{F}_q[x]/(x^n - 1)$ est un $\mathbb{F}_q$-espace vectoriel de dimension n, dont une base est $(\dot{1}, \dot{x}, ..., \dot{x}^{n-1})$. Cela nous permet d'identifier (vectoriellement) un élément $a = (a_0, ..., a_{n-1})$ de $\mathbb{F}_q^n$ au polynôme $a(x) = a_0 + a_1\dot{x} + ... + a_{n-1}\dot{x}^{n-1}$ de $\mathbb{F}_q[x]/(x^n - 1)$. Pour simplifier les notations, on oubliera de mettre le point au-dessus de la classe de x, de sorte que l'on écrira :

$$a(x) = a_0 + a_1 x + ... + a_{n-1}x^{n-1} \in \mathbb{F}_q[x]/(x^n - 1)$$

le polynôme associé au vecteur $a = (a_0, ..., a_{n-1}) \in \mathbb{F}_q^n$. Avec cette identification, le code C devient un sous-espace vectoriel de $\mathbb{F}_q[x]/(x^n - 1)$. Si C est un code cyclique,

$$a(x) = a_0 + a_1 x + ... + a_{n-1}x^{n-1} \in C \Rightarrow x.a(x) = a_{n-1} + a_0 x + ... + a_{n-2}x^{n-1} \in C$$

et, de proche en proche, on obtient $x^i a(x) \in C$ pour tout entier i. Par linéarité le polynôme $b(x)\,a(x)$ appartiendra à C quel que soit le polynôme b de $\mathbb{F}_q[x]/(x^n - 1)$, et C sera un idéal de l'anneau-quotient $\mathbb{F}_q[x]/(x^n - 1)$. Réciproquement, tout idéal de $\mathbb{F}_q[x]/(x^n - 1)$ est un sous-espace vectoriel stable par permutation à droite des coordonnées dans la base $\left(1, x, ..., x^{n-1}\right)$, et l'on peut énoncer :

Théorème 3 *Un code cyclique est un idéal de $\mathbb{F}_q[x]/(x^n - 1)$ dès que l'on identifie les $\mathbb{F}_q$-espaces vectoriels $\mathbb{F}_q^n$ et $\mathbb{F}_q[x]/(x^n - 1)$.*

On peut démontrer le Théorème suivant qui précise la nature des idéaux de $\mathbb{F}_q[x]/(x^n - 1)$.

Théorème 4 *Soit k un corps et $f \in k[x]$. L'anneau $A = k[x]/(f)$ est principal, et tout idéal de A est de la forme $(\dot{g})$ où g est un polynôme unitaire de $k[x]$ qui divise f. De plus un tel polynôme g est unique.*

Définition 6 *Soit C un code cyclique. L'unique polynôme unitaire g de $\mathbb{F}_q[x]$ qui divise $x^n - 1$ et tel que $C = (g)$ est appelé* ***polynôme générateur*** *du code cyclique C.*

Soit C le code cyclique de longueur n sur $\mathbb{F}_q$ et de polynôme générateur g. Ecrivons $x^n - 1 = g(x) h(x)$ et posons $\deg g = n - k$. Le code C est l'image de l'application linéaire :

$$\begin{array}{ccccc} \gamma : & \mathbb{F}_q^k & \to & \mathbb{F}_q^n \simeq \mathbb{F}_q[x]/(x^n-1) \\ & a = (a_0, ..., a_{k-1}) & \mapsto & a(x) g(x) \end{array}$$

puisque tout élément de C est un multiple de g (dans l'identification des espaces vectoriels $\mathbb{F}_q^n$ et $\mathbb{F}_q[x]/(x^n-1)$). Comme $\deg(a(x) g(x)) \leq n-1$, l'application γ sera injective et réalisera un encodage simple de C. On en déduit aussi :

Théorème 5 *La dimension du code cyclique de longueur n sur $\mathbb{F}_q$ et de polynôme générateur g est $n - \deg g$.*

Notons $g(x) = g_0 + g_1 x + ... + g_{n-k} x^{n-k}$ et $a(x) = a_0 + a_1 x + ... + a_{k-1} x^{k-1}$. Alors :

$$a(x) g(x) = a_0 g_0 + (a_0 g_1 + a_1 g_0) x + ... + a_{k-1} g_{n-k} x^{n-1}$$

et $\gamma(a) = (a_0, a_1, ..., a_{k-1}) G$ avec la matrice génératrice :

$$G = \begin{pmatrix} g_0 & g_1 & \cdots & g_{n-k} & 0 \cdots & 0 \\ 0 & g_0 & & & & \vdots \\ \vdots & & \ddots & & \ddots & 0 \\ 0 & \cdots & 0 & g_0 & \cdots & g_{n-k} \end{pmatrix}.$$

L'application $S : a \mapsto S(a) = a(x) h(x)$ définit un syndrome de C puisque :

$$a(x) \in C \Leftrightarrow a(x) h(x) = 0.$$

En effet $a(x) \in C$ entraîne $a(x) = b(x) g(x)$, d'où :

$$a(x) h(x) = b(x) g(x) h(x) = 0.$$

Réciproquement, $a(x)\,h(x) = 0$ se traduit par $(x^n - 1)\,|a(x)\,h(x)$ qui entraîne $g(x)\,|a(x)$ et $a(x) \in C$. On déduit alors les équivalences :

$$\begin{aligned} a(x) \in C &\Leftrightarrow \left(a_0 + a_1 x + ... + a_{n-1}x^{n-1}\right)\left(h_0 + h_1 x + ... + h_k x^k\right) \\ &\Leftrightarrow H\,{}^t a = 0 \end{aligned}$$

avec :

$$H = \begin{pmatrix} h_0 & 0 & \cdots & 0 & h_k & \cdots & h_2 & h_1 \\ h_1 & h_0 & & & \ddots & \ddots & & h_2 \\ h_2 & h_1 & h_0 & & & \ddots & \ddots & \vdots \\ & & h_1 & \ddots & & & \ddots & h_k \\ \vdots & & \ddots & & \ddots & & & 0 \\ h_k & \cdots & & & \ddots & h_0 & & \vdots \\ & \ddots & & & & h_1 & h_0 & 0 \\ 0 & & h_k & & \cdots & & h_1 & h_0 \end{pmatrix}.$$

On a trouvé une matrice de contrôle de C de taille $n \times n$.

8 Codes BCH (Bose-Chaudhuri-Hocquenghem)

Les codes BCH sont des codes cycliques particuliers très utilisés et adaptés à des situations variées. Ainsi le Minitel utilise un code BCH primitif de longueur 127 et de polynôme générateur $g(x) = x^7 + x^3 + 1$. Le code CIRC (Cross Interleaved Reed-Solomon Code) est présent dans les CD Audio et permet de corriger des paquets d'erreurs ou d'effacements de 4096 bits consécutifs correspondant à une rayure d'environ un millimètre (voir [2], [13] ou [21] pour plus de détails). Enfin signalons qu'un code de Reed-Solomon $[255, 223, 33]$ a été adopté comme norme de transmission entre satellites en 1982 par le CCSDS (Consultative Committe for Space Data System) et a été utilisé sur la sonde européenne Giotto, puis sur le satellite d'exploration de Jupiter Galileo ([16] p. 150).

Si q est premier avec n, l'ordre multiplicatif de q modulo n est, par définition, l'ordre de l'élément q dans le groupe multiplicatif des éléments inversibles de $\mathbb{Z}/n\mathbb{Z}$. C'est donc le plus petit entier naturel non nul m tel que $q^m \equiv 1\ (n)$. Notons m l'ordre multiplicatif de q modulo n. Soit α une racine primitive n-ième de l'unité (dans une extension algébriquement close de $\mathbb{F}_q$). On a :

$$\alpha \in \mathbb{F}_{q^t} \ \Leftrightarrow\ \alpha^{q^t-1} = 1 \ \Leftrightarrow\ n|\left(q^t - 1\right) \ \Leftrightarrow\ q^t \equiv 1\ (n) \ \Leftrightarrow\ m|t.$$

La plus petite extension de $\mathbb{F}_q$ contenant α sera donc $\mathbb{F}_{q^m}$, et l'on peut écrire $\mathbb{F}_{q^m} = \mathbb{F}_q(\alpha)$.
Le corps $\mathbb{F}_{q^m}$ sera la plus petite extension de $\mathbb{F}_q$ contenant toutes les racines α^t $(0 \leq t \leq n-1)$ du polynôme $x^n - 1$. C'est le corps de décomposition de $x^n - 1$ sur $\mathbb{F}_q$, encore appelé corps des racines n-ième de l'unité sur $\mathbb{F}_q$.
Soit $p(x)$ le polynôme minimal de α sur $\mathbb{F}_q$. C'est, par définition, le polynôme irréductible unitaire de $\mathbb{F}_q[x]$ tel que $p(\alpha) = 0$, et l'on sait que $p(x)$ divise $x^n - 1$. De :

$$\mathbb{F}_{q^m} = \mathbb{F}_q(\alpha) \simeq \mathbb{F}_q[x] / (p(x))$$

on déduit $m = \dim_{\mathbb{F}_q}(\mathbb{F}_q[x] / (p(x))) = \deg p(x)$. On sait aussi que les racines de $p(x)$ sont exactement α, α^q, ..., $\alpha^{q^{m-1}}$ [9].

Définition 7 *Soient n un entier naturel premier avec q, et d un entier tel que $2 \leq d \leq n$. On appelle* ***code BCH de longueur*** *n* ***et de distance construite*** *d* ***sur*** *$\mathbb{F}_q$ tout code cyclique de longueur n et de polynôme générateur :*

$$g(x) = \text{ppcm}(p_r(x), p_{r+1}(x), ..., p_{r+d-2}(x))$$

où $r \in \mathbb{N}^*$, où $p_i(x)$ désigne le polynôme minimal de α^i sur $\mathbb{F}_q$, et où α est une racine primitive n-ième de l'unité dans $\mathbb{F}_{q^m}$.

Bien entendu g divise $x^n - 1$ et représente le polynôme de $\mathbb{F}_q[x]$ de plus petit degré admettant les éléments distincts $\alpha^r, \alpha^{r+1}, ..., \alpha^{r+d-2}$ comme racines.

Définition 8 *Si $r = 1$, on dit que C est un* ***code BCH au sens strict****. Si $n = q^m - 1$, C est appelé* ***code BCH primitif*** *(dans ce cas α est un élément primitif de $\mathbb{F}_{q^m}$). Enfin un* ***code de Reed-Solomon*** *de longueur $q-1$ est un code BCH de longueur $n = q - 1$ sur le corps $\mathbb{F}_q$ (autrement dit α est un élément primitif de $\mathbb{F}_q$).*

Théorème 6 *La distance minimale d'un code BCH est supérieure à sa distance construite.*

Preuve — Un mot $a = (a_0, a_1, ..., a_{n-1})$ appartient à C si, et seulement si, le polynôme associé $a(x) = a_0 + a_1 x + ... + a_{n-1}x^{n-1}$ est multiple de $g(x)$. Cela revient à dire que $a(x)$ est divisible par chacun des polynômes $p_i(x)$, ou encore que $a(x)$ admet $\alpha^r, \alpha^{r+1}, ..., \alpha^{r+d-2}$ comme racines. Cela s'écrit :

$$\forall j \in \{0, ..., d-2\} \quad a_0 + a_1\alpha^{r+j} + a_2\alpha^{2(r+j)} + ... + a_{n-1}\left(\alpha^{r+j}\right)^{n-1} = 0$$

ou encore $H\,{}^t a = 0$ avec :

$$H = \begin{pmatrix} 1 & \alpha^r & \alpha^{2r} & \alpha^{(n-1)r} \\ 1 & \alpha^{r+1} & \alpha^{2(r+1)} & \alpha^{(n-1)(r+1)} \\ \vdots & & & \vdots \\ 1 & \alpha^{r+d-2} & \alpha^{2(r+d-2)} & \alpha^{(n-1)(r+d-2)} \end{pmatrix}.$$

La matrice H est une matrice de contrôle de C et l'on peut vérifier que $d-1$ colonnes quelconques de cette matrice sont linéairement indépendantes. En effet le déterminant :

$$D = \begin{vmatrix} \alpha^{i_1 r} & \alpha^{i_2 r} & \alpha^{i_{d-1} r} \\ \alpha^{i_1 (r+1)} & \alpha^{i_2 (r+1)} & \alpha^{i_{d-1} (r+1)} \\ & & \vdots \\ \alpha^{i_1 (r+d-2)} & \alpha^{i_2 (r+d-2)} & \alpha^{i_{d-1} (r+d-2)} \end{vmatrix}$$

se calcule en utilisant le déterminant de Vandermonde. On trouve :

$$\begin{aligned} D &= \alpha^{(i_1+\ldots+i_{d-1})r} \begin{vmatrix} 1 & 1 & 1 \\ \alpha^{i_1} & \alpha^{i_2} & \alpha^{i_{d-1}} \\ & & \vdots \\ \alpha^{i_1 (d-2)} & \alpha^{i_2 (d-2)} & \alpha^{i_{d-1} (d-2)} \end{vmatrix} \\ &= \alpha^{(i_1+\ldots+i_{d-1})r} \prod_{l<k} \left(\alpha^{i_k} - \alpha^{i_l} \right) \neq 0. \end{aligned}$$

Il suffit d'appliquer le Théorème 2 pour conclure. ■

Théorème 7 *Un code de Reed-Solomon de longueur $q-1$ et de distance construite d ($2 \leq d \leq q-1$) est un code cyclique de polynôme générateur :*

$$g(x) = (x - \alpha^r)(x - \alpha^{r+1})...(x - \alpha^{r+d-2})$$

où α est un élément primitif de $\mathbb{F}_q$. Ses paramètres sont $[q-1, q-d, d]$ et c'est donc un code MDS.

Preuve : Le polynôme minimal de α^i sur $\mathbb{F}_q$ est $p_i(x) = x - \alpha^i$ de sorte que :

$$g(x) = (x - \alpha^r)(x - \alpha^{r+1})...(x - \alpha^{r+d-2})$$

engendre C et $\dim C = k = n - \deg g = n - d + 1$. Le Théorème 6 et la borne de Singleton permettent d'encadrer la distance minimale $\operatorname{dist} C$ de C pour obtenir $d \leq \operatorname{dist} C \leq n - k + 1 = d$. ■

9 Exemple de décodage matriciel d'un code BCH binaire

Cherchons une racine primitive 15-ième de l'unité sur $\mathbb{F}_2$. Cela revient à connaître un polynôme irréductible de degré 4 et d'ordre 15 appartenant à l'anneau $\mathbb{F}_2[x]$.

Une table (par exemple, la table C du chap. 10 de [9]) répertorie seulement 3 polynômes irréductibles de degré 4. Ce sont les polynômes x^4+x+1 d'ordre 15, x^4+x^3+1 d'ordre 15 et $x^4+x^3+x^2+x+1$ d'ordre 5. Choisissons le polynôme $P(x) = x^4 + x + 1$. Une racine α de $P(x)$ sera d'ordre multiplicatif 15, et c'est elle que nous choisirons pour construire nos exemples.

On sait que les racines de $P(x)$ sont α, α^2, α^4 et α^8. Notons $p_i(x)$ le polynôme minimal de α^i sur $\mathbb{F}_2$. Le code C_d cyclique de longueur 15 engendré par le polynôme :

$$g(x) = \text{ppcm}(p_1(x), p_2(x), ..., p_{d-1}(x))$$

est un code BCH de distance construite d qui corrigera au moins $\left[\frac{d-1}{2}\right]$ erreurs. Considérons le code cyclique C_2 de polynôme générateur $p_1(x) = P(x)$. Comme $C_2 = C_3$, ce code corrige au moins 1 erreur. On a :

$$a = (a_0, a_1, ..., a_{14}) \in C_2 \Leftrightarrow a(\alpha) = 0 \text{ où } a(x) = a_0 + a_1x + ... + a_{14}x^{14}.$$

Une matrice de contrôle de C_2 sera donc :

$$H = \begin{pmatrix} 1 & \alpha & \alpha^2 & \alpha^3 & \alpha^4 & \alpha^5 & \alpha^6 & \alpha^7 & \alpha^8 & \alpha^9 & \alpha^{10} & \alpha^{11} & \alpha^{12} & \alpha^{13} & \alpha^{14} \end{pmatrix}$$

ou encore :

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

en exprimant chacun des α^i dans la base $\left(1, \alpha, \alpha^2, \alpha^3\right)$ du $\mathbb{F}_2$-espace vectoriel $\mathbb{F}_{16}$. Une permutation des colonnes de H nous donne une matrice de contrôle du code $[15, 11]$ de Hamming. Toutes les colonnes de H sont discernables et cela permet de localiser une erreur. En effet, si $x \in \mathbb{F}_2^{15}$ désigne le message envoyé, et si le message reçu y contient au plus une erreur, alors $y = x + e$ où e est un mot de poids 0 ou 1. En posant $S(y) = H\,{}^t y$, on trouve $S(y) = S(e)$. Si $S(y) = 0$, il n'y a pas eu d'erreur. Si le vecteur $S(y)$ n'est pas nul, il sera égal à l'un des vecteurs-colonnes de H et l'indice de cette colonne représentera la place de l'erreur.

Considérons maintenant un code BCH construit sur les racines consécutives $\alpha, \alpha^2, \alpha^3$. Comme α^4 est aussi racine de $p_1(x)$, le code C_4 sera identique au code C_5 et corrigera au moins $\left[\frac{5-1}{2}\right] = 2$ erreurs. Comme $\text{pgcd}(3, 15) = 3$, l'ordre multiplicatif de α^3 sera 5, et le polynôme minimal $p_3(x)$ de α^3 sera un polynôme d'ordre 5 appartenant à la liste des polynômes irréductibles de degré 4 donnée plus haut. Donc :

$$p_3(x) = x^4 + x^3 + x^2 + x + 1.$$

En conclusion le code C_4 est cyclique engendré par le polynôme :

$$g(x) = p_1(x)\, p_3(x).$$

On a :

$a = (a_0, a_1, ..., a_{14}) \in C_4 \Leftrightarrow a(\alpha) = a\left(\alpha^3\right) = 0$ où $a(x) = a_0 + a_1 x + ... + a_{14} x^{14}$.

Une matrice de contrôle de C_2 sera donc :

$$H = \begin{pmatrix} 1 & \alpha & \alpha^2 & \alpha^3 & ... & \alpha^i & ... & \alpha^{12} & \alpha^{13} & \alpha^{14} \\ 1 & \alpha^3 & \alpha^{3\times 2} & \alpha^{3\times 3} & ... & \alpha^{3i} & ... & \alpha^{3\times 12} & \alpha^{3\times 13} & \alpha^{3\times 14} \end{pmatrix}.$$

En exprimant chacune des puissances de α dans la base $\left(1, \alpha, \alpha^2, \alpha^3\right)$, on obtient encore :

$$H = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 \end{pmatrix} = [h_1, ..., h_{15}].$$

Soit $x \in \mathbb{F}_2^{15}$ le message envoyé. Supposons que le message reçu y contient moins de deux erreurs. On a $y = x + e$ où e est un mot de poids 0, 1 ou 2. Posons :

$$S(y) = H\, {}^t y = S(e) = \begin{pmatrix} A \\ B \end{pmatrix}.$$

De trois choses l'une :

- Si $S(y) = 0$, aucune erreur ne s'est produite et $y = x$.

- Si $\mathrm{w}(e) = 1$, un seule erreur s'est produite à la place i et $S(e) = \begin{pmatrix} \alpha^i \\ \alpha^{3i} \end{pmatrix}$ représente la i-ème colonne de H (lorsque les colonnes sont numérotées de 0 à 14). Comme toutes les colonnes de H sont discernables, la position i de l'erreur est facilement repérable et l'on peut corriger le message.

- Si $\mathrm{w}(e) = 2$, deux erreurs se sont produites aux places i et j à déterminer. On connaît parfaitement le vecteur :

$$S(y) = S(e) = \begin{pmatrix} A \\ B \end{pmatrix} = \begin{pmatrix} \alpha^i + \alpha^j \\ \alpha^{3i} + \alpha^{3j} \end{pmatrix}.$$

Le calcul :

$$A^3 = B + 3\alpha^{2i}\alpha^j + 3\alpha^i\alpha^{2j} = B + \alpha^i\alpha^j A$$

montre que A n'est pas nul (sinon $A = B = 0$ entraîne $S(y) = 0$). Alors $\alpha^i\alpha^j = \frac{A^3-B}{A}$ et les nombres α^i et α^j seront les racines de l'équation du second degré $AX^2 - A^2X + A^3 - B = 0$. Il ne reste plus qu'à résoudre cette équation dans $\mathbb{F}_{16}$ par substitution pour obtenir les places i et j.

Une dernière remarque reste à faire pour expliquer comment on peut savoir si $\mathrm{w}(e)$ vaut 1 ou 2 à la seule lecture de $S(y)$. En fait on réagira toujours en calculant $S(y)$ puis en cherchant une colonne de H identique à $S(y)$ dès que $S(y) \neq 0$. Si une telle colonne est découverte, on peut affirmer que $\mathrm{w}(e) = 1$ et qu'une seule erreur s'est produite. Si $S(y)$ ne correspond à aucune des colonnes de H, on peut affirmer que $\mathrm{w}(e) = 2$ et conclure comme ci-dessus. L'explication en est simple : il n'existe pas de triplets (i, j, k) dans $\{0, ..., 14\}^3$ tels que :

$$\begin{pmatrix} \alpha^i + \alpha^j \\ \alpha^{3i} + \alpha^{3j} \end{pmatrix} = \begin{pmatrix} \alpha^k \\ \alpha^{3k} \end{pmatrix}.$$

Dans le cas contraire on aurait $\alpha^{3i}+\alpha^{3j} = \left(\alpha^i + \alpha^j\right)^3$ d'où $\alpha^i\alpha^j\left(\alpha^i + \alpha^j\right) = 0$, ce qui est absurde.

10 Cryptographie à clé révélée

Si la cryptographie existe depuis l'antiquité et était essentiellement utilisée à des fins militaires, les années 1970 ont généralisé son emploi dans les domaines civils, créant par la même de nouveaux besoins. L'utilisation d'un cryptosystème conventionnel pose un certain nombre de problèmes :

1. Multiplication des clés – Pour sécuriser une information entre deux partenaires, ceux-ci doivent posséder la même clé K permettant, pour l'un, d'utiliser sa fonction de chiffrement C_K, et pour l'autre, d'obtenir sa fonction de déchiffrement D_K. Ainsi n abonnés auront besoin de n^2 clés distinctes correspondant à chaque couple (E, R) que l'on peut former en choisissant un émetteur E et un récepteur R dans la même liste des n abonnés.
Toutes ces clés K doivent rester secrètes, et cela peut être difficile à réaliser lorsqu'il s'agit de communications entre des sociétés où de nombreuses personnes peuvent avoir accès aux données confidentielles contenues dans un coffre. Cela nécessite aussi de changer les clés classiques à intervalles réguliers.

2. Communication des clés – Deux interlocuteurs potentiels utilisant un réseau électronique devront se communiquer leur clé commune avant de pouvoir obtenir une communication sécurisée. La seule façon de le faire serait

d'utiliser un courrier classique contenant la clé. C'est insupportable si l'on utilise un moyen rapide de communication comme la poste électronique. Un nouveau dispositif doit être inventé pour permettre le choix de la clé commune entre nos interlocuteurs et interdire sa lecture par une tierce personne.

3. Signature – Un correspondant mal intentionné peut facilement créer un message en clair, puis le crypter en utilisant la fonction C_K. A partir de là, il peut décrypter le message et faire croire qu'il a été obtenu par la voie normale. L'utilisation d'un système conventionnel interdit par conséquent l'envoi d'un chèque électronique à un créancier puisqu'un juge ne pourra jamais affirmer qu'un chèque a effectivement été envoyé par la personne qui l'a signé. L'association $(C_K(M), M)$ du message chiffré et de sa traduction en clair ne constitue en aucun cas la preuve de l'envoi de ce message par la personne indiquée.

Le problème de la signature d'un message est beaucoup plus grave que celui de son authentification. Authentifier un message, c'est créer les conditions pour que le receveur soit assuré que le message qu'il vient de déchiffrer provient bien de la personne indiquée. Signer un message, c'est se donner les moyens de démontrer à une tierce personne que le message reçu provient bien de la personne qui l'a signé. Bien évidemment, la signature d'un message doit dépendre à la fois de l'expéditeur et du message lui-même, pour éviter toute modification du message originel (problème de l'intégrité du message).

C'est pour répondre à ces nouvelles exigences que Diffie et Hellman ont inventé le concept de cryptosystème à clés révélées en 1976 ([4], [3]). La première implémentation pratique utilisant ce concept a été donné en 1978, et il s'agit du système RSA de Rivest, Shamir et Adleman ([1]). Ce système est très populaire et utilisé dans le célèbre PGP (Pretty Good Privacy) de Phil Zimmerman pour l'envoi de lettres électroniques cryptées.

Dans un cryptosystème classique, l'émetteur E et le récepteur R connaissent tous les deux parfaitement la fonction de chiffrement C et la fonction de déchiffrement D.

Dans un cryptosystème à clé révélée, le récepteur est le seul à connaître les fonctions C et D, et permettra à tout le monde d'avoir accès à la fonction C de chiffrement en la publiant, par exemple, sur un bottin. La fonction C est appelée clé publique, ou clé révélée, de chiffrement, et la fonction D est appelée clé secrète de déchiffrement.

Un émetteur E désirant envoyer un message M au récepteur R devra d'abord récupérer la clé publique C correspondant à R sur un bottin. Ensuite l'émetteur E calcule $C(M)$ et envoie ce message à R. A l'arrivée, le récepteur R applique sa clé de déchiffrement D au message pour obtenir $D(C(M)) = M$.

Les fonctions C et D, définies et à valeurs dans l'ensemble $\mathcal{M}$ des messages, doivent vérifier les propriétés suivantes :

P1 : Pour tout message M, on a $D(C(M)) = M$ (en particulier C est injective),

P2 : La connaissance de C et de $M' = C(M)$ interdit le calcul de $D(M')$ en un temps raisonnable,

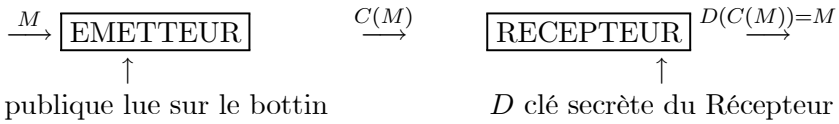
auxquelles on ajoute éventuellement la propriété :

P3 : Pour tout message M, on a $C(D(M)) = M$.

Définition 9 *Une* ***fonction trappe*** *ou* ***fonction à sens unique*** *est une fonction*
$C : \mathcal{M} \to \mathcal{M}$ *pour laquelle il existe une fonction* $D : \mathcal{M} \to \mathcal{M}$ *vérifiant les propriétés* ***P1*** *et* ***P2****. Une* ***permutation trappe*** *ou* ***permutation à sens unique*** *vérifie en outre la propriété* ***P3****.*

L'utilisation d'une fonction trappe C résout immédiatement les problèmes **1** et **2** de la multiplication des clés et de leur communication. En effet, les clés de chiffrement C sont connus de tous les abonnés et n'ont plus besoin d'être transmises. Ensuite, le nombre de clés nécessaires à la communication entre n abonnés devient n au lieu de n^2, et, à elles seules, ces n clés permettront la communication de n'importe quelle personne vers l'un des abonnés, même si celle-ci ne figure pas dans la liste de ces abonnés.

La communication à clé révélée peut se résumer dans le diagramme :



Communication à clé révélée

Dans la pratique l'utilisation d'un cryptosystème à clés publiques est plus lent que celui d'un cryptosystème classique, et le système à clés publiques permet seulement l'échange d'une clé commune secrète permettant à un système conventionnel de fonctionner. Une fois cette clé commune partagée, l'utilisation d'un chiffrement classique représente un gain de temps appréciable surtout pour des messages longs.

Supposons maintenant que l'application C soit une permutation à sens unique, autrement dit qu'elle vérifie **P1**, **P2** et **P3**. Supposons aussi que l'émetteur E désire envoyer et signer un message M au récepteur R, et que à la fois E et R appartiennent à la même liste d'abonnés à notre système de cryptographie

publique. Notons C_E et D_E (resp. C_R et D_R) la clé publique de chiffrement et la clé secrète de déchiffrement de E (resp. R).

L'émetteur utilise sa clé secrète D_E sur son message M pour obtenir $D_E(M)$, puis utilise la clé publique du destinataire R pour obtenir le message crypté $M' = C_R(D_E(M))$. Le destinataire R reçoit ce message M' qui provient normalement de E. Il cherche la clé publique C_E de E, puis calcule $C_E(D_R(M'))$ pour obtenir M, puisque :

$$C_E(D_R(C_R(D_E(M)))) = M.$$

Schématiquement :

$$M \quad \rightsquigarrow \quad M' = C_R(D_E(M)) \quad \rightsquigarrow \quad D_R(M') = M'' \rightsquigarrow \quad C_E(M'') = M$$

Le récepteur R est maintenant assuré que E est bien l'émetteur du message M. Il peut même prouver à une tierce personne que le message M qu'il vient de déchiffrer provient de E. En effet, personne d'autre que E n'aurait pu obtenir un message $M'' = D_E(M)$ tel que $C_E(M'') = M$. Le problème de la signature du message est résolu grâce aux permutations à sens unique. Résumons-nous par le diagramme :

$$\xrightarrow{M} \boxed{\text{EMETTEUR } E} \xrightarrow{M'=C_R(D_E(M))} \boxed{\text{RECEPTEUR } R} \xrightarrow{C_E(D_R(M'))=M}$$

C_R publique, D_E secrète — C_E publique, D_R secrète

Communication à clé révélée avec signature

11 Le système RSA (Rivest, Shamir & Adleman, 1978)

11.1 Une généralisation du Théorème d'Euler

La fonction arithmétique $\varphi : \mathbb{N}^* \to \mathbb{N}$ qui à tout entier naturel non nul n associe le nombre d'entiers naturels de l'intervalle $[1, n]$ premiers avec n, est appelée **fonction indicatrice d'Euler**. Si $n \geq 2$, on sait que $\varphi(n)$ est égal au cardinal du sous-groupe multiplicatif U des éléments inversibles de l'anneau $\mathbb{Z}/n\mathbb{Z}$, et l'on rappelle que si $n = p_1^{\alpha_1}...p_k^{\alpha_k}$ désigne la décomposition de n en produit de facteurs premiers, alors :

$$\varphi(n) = (p_1^{\alpha_1} - p_1^{\alpha_1 - 1})...(p_k^{\alpha_k} - p_k^{\alpha_k - 1}).$$

On sait aussi qu'un élément $\dot{x}$ de $\mathbb{Z}/n\mathbb{Z}$ est inversible si et seulement si x est premier avec n.
D'après le Théorème de Lagrange, tout élément x d'un groupe multiplicatif fini G de cardinal s et d'élément neutre e vérifie $x^s = e$. En appliquant ce résultat à U on obtient $\dot{x}^{\varphi(n)} = \dot{1}$ pour tout $\dot{x} \in U$, soit :

Théorème 8 ***Théorème d'Euler***
Si n est un entier supérieur ou égal à 2, alors $x^{\varphi(n)} \equiv 1\ (n)$ pour tout entier x premier avec n.

On peut améliorer le Théorème d'Euler lorsque l'entier naturel $n \geq 2$ est libre de tout carré, c'est-à-dire sans facteur carré dans sa décomposition (on dit encore quadratfrei ou square free) :

Théorème 9 *Soit n un entier naturel supérieur ou égal à 2 et libre de tout carré. On note $n = p_1...p_k$ où les nombres premiers p_i sont distincts entre eux deux à deux. Soit m un multiple commun aux nombres $p_1 - 1$,... , $p_k - 1$. Soient c et d deux entiers naturels tels que $cd \equiv 1\ (m)$, autrement dit tels qu'il existe un entier k pour lequel $cd = km + 1$. Alors*
1) La congruence $x^{cd} \equiv x\ (n)$ est vraie pour tout entier relatif x.
2) Si $m \geq 0$, alors $x^{m+1} \equiv x\ (n)$ pour tout $x \in \mathbb{Z}$.
3) On a $x^{\varphi(n)+1} \equiv x\ (n)$ pour tout $x \in \mathbb{Z}$.

Preuve — 1) Soit $x \in \mathbb{Z}$. Si p est un facteur premier de n, le petit Théorème de Fermat permet d'écrire $x^{p-1} \equiv 1\ (p)$ pour tout entier x non divisible par p. Comme km est un multiple de $p - 1$, on déduit $x^{km} \equiv 1\ (p)$, ou encore $x^{cd} \equiv x^{km+1} \equiv x\ (p)$. Cette dernière congruence est en fait triviale lorsque x est divisible par p, de sorte que l'on puisse écrire $x^{cd} \equiv x\ (p)$ pour tout $x \in \mathbb{Z}$. La différence $x^{cd} - x$ sera divisible par chacun des nombres premiers p_i, donc aussi par le produit $n = p_1...p_k$, et l'on obtient bien $x^{cd} \equiv x\ (n)$

2) On applique 1) avec $k = 1$, $d = 1$ et $c = m + 1$.

3) On applique 2) avec $m = \varphi(n)$, ce qui est possible puisque :

$$\varphi(n) = \varphi(p_1...p_k) = (p_1 - 1)...(p_k - 1)$$

est bien un multiple commun de $p_1 - 1$, ..., $p_k - 1$. ■

11.2 Principe du système RSA

Soient p et q deux entiers premiers et $n = pq$. Soient c et d deux entiers naturels tels que $cd \equiv 1\ (\varphi(n))$ où $\varphi(n) = (p-1)(q-1)$ (on peut remplacer $\varphi(n)$ par n'importe quel multiple m commun à $p-1$ et $q-1$). On peut écrire

$cd = k\varphi(n) + 1$ où $k \in \mathbb{Z}$. Les messages x seront des entiers appartenant à $\{0, 1, ..., n-1\}$. Le Théorème 9 montre que $x^{cd} \equiv x \ (n)$. Il suffit donc de poser :

$$C(x) \equiv x^c \ (n) \quad \text{et} \quad D(y) \equiv y^d \ (n)$$

pour obtenir :

$$D(C(x)) \equiv D(x^c) \equiv x^{cd} \equiv x \ (n) \text{ et } C(D(y)) \equiv C(y^d) \equiv x^{dc} \equiv x \ (n)$$

pour tout entier x. Les fonctions C et D seront les fonctions de chiffrement et de déchiffrement.

Faisons quelques remarques :

1) Pour chiffrer, on a besoin de C, soit de c et de n, qui appartiendront au domaine public et seront accessibles en consultant un bottin.

2) Pour déchiffrer, il faut connaître d et n.

3) On peut abusivement écrire $C = (c, n)$ et $D = (d, n)$. Dans ce cas C représente la clé publique de chiffrement et D la clé secrète de déchiffrement, et il s'agit de vérifier que C est une permutation à sens unique.

Si l'entier $n = pq$ est connu de tout le monde, les nombres premiers p et q seront conservés secrets car leur connaissance entraîne celle de $\varphi(n) = (p-1)(q-1)$, puis celle de d en résolvant l'équation de Bezout $cd - k\varphi(n) = 1$, ce qui est possible puisque c est dans l'annuaire. Pour que le système puisse fonctionner de manière sûre, il faut choisir un entier n très grand possédant au moins 200 chiffres et rendre ainsi le calcul de p et q impossible en un temps raisonnable même si l'on a recours à des ordinateurs puissants.
L'existence et la sécurité du système RSA reposent sur la facilité d'obtenir des entiers premiers p et q de plus de 100 chiffres, et sur la difficulté de retrouver la décomposition de l'entier $n = pq$ en un temps raisonnable. Le tableau suivant, relevé dans [1], donne une idée du temps nécessaire pour obtenir la décomposition de n.

Nombre de chiffres	Nombre d'opérations	Temps de calcul
50	$1,4 \times 10^{10}$	3,9 h
75	$9,0 \times 10^{12}$	104 jours
100	$2,3 \times 10^{15}$	74 ans
200	$1,2 \times 10^{23}$	$3,8 \times 10^{9}$ ans
300	$1,5 \times 10^{29}$	$4,9 \times 10^{15}$ ans
500	$1,3 \times 10^{39}$	$4,2 \times 10^{25}$ ans.

En 1983, le temps moyen nécessaire à un gros ordinateur pour tester la primalité d'un entier dans les cas les plus défavorables est donné dans le tableau

ci-dessous dû à Pomerance ([18], [3], [10]).

nombre de chiffres :	20	50	100	200	1000
temps de calcul :	10 sec	15 sec	40 sec	10 min	1 semaine.

11.3 Exemple numérique

Prenons $p = 163$, $q = 359$ et $n = 163 \times 359 = 58517$. Ici :

$$\varphi(n) = 162 \times 358 = 57996 = 2^2 \times 3^4 \times 179$$

et l'on peut choisir $c = 5 \times 17 \times 59 = 5015$ qui est premier avec $\varphi(n)$. On résout l'équation de Bezout :

$$5015d = 1 + 57996k \quad (*)$$

en utilisant l'algorithme d'Euclide étendu. On obtient :

$$57996 = 5015 \times 11 + 2831 \; ; \quad 5015 = 2831 + 2184 \; ; \quad 2831 = 2184 + 647 \; ;$$
$$2184 = 647 \times 3 + 243 \; ; \quad 647 = 243 \times 2 + 161 \; ; \quad 243 = 161 + 82 \; ;$$
$$161 = 82 + 79 \; ; \quad 82 = 79 + 3 \; ; \quad 79 = 3 \times 26 + 1.$$

D'où :

$$\begin{aligned}
1 &= 79 - 3 \times 26 \\
&= 79 - (82 - 79) \times 26 = 27 \times 79 - 82 \times 26 \\
&= 27 \times (161 - 82) - 82 \times 26 = 27 \times 161 - 53 \times 82 \\
&= 27 \times 161 - 53 \times (243 - 161) = 80 \times 161 - 53 \times 243 \\
&= 80 \times (647 - 243 \times 2) - 53 \times 243 = 80 \times 647 - 213 \times 243 \\
&= 80 \times 647 - 213 \times (2184 - 647 \times 3) = 719 \times 647 - 213 \times 2184 \\
&= 719 \times (2831 - 2184) - 213 \times 2184 = 719 \times 2831 - 932 \times 2184 \\
&= 719 \times 2831 - 932 \times (5015 - 2831) = 1651 \times 2831 - 932 \times 5015 \\
&= 1651 \times (57996 - 5015 \times 11) - 932 \times 5015 = 1651 \times 57996 - 19093 \times 5015.
\end{aligned}$$

On a les équivalences :

$$\begin{aligned}
5015d - 57996k = 1 &\Leftrightarrow 5015d - 57996k = 1651 \times 57996 - 19093 \times 5015 \\
&\Leftrightarrow 5015\,(d + 19093) = 57996\,(1651 + k)\,.
\end{aligned}$$

Comme 5015 et 57996 sont premiers entre eux, le Théorème de Gauss montre l'existence de $t \in \mathbb{Z}$ tel que $1651 + k = 5015t$, d'où $d + 19093 = 57996t$.

La réciproque étant évidente, les solutions entières de l'équation $(*)$ seront données par :

$$(d, k) = (57996t - 19093, 5015t - 1651)$$

où $t \in \mathbb{Z}$.

Choisissons $d = 57996 - 19093 = 38903$. Si $x \in \{0, 1, ..., 58516\}$, la clé de chiffrement sera $C(x) = x^{5015} \bmod 58517$ et la clé de déchiffrement sera $D(x) = x^{38903} \bmod 58517$. Choisissons un tableau de correspondance permettant d'écrire les 26 lettres de l'alphabet usuel et quelques caractères spéciaux sous la forme de nombres. Par exemple :

espace	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15

P	Q	R	S	T	U	V	W	X	Y	Z	.	,	’
16	17	18	19	20	21	22	23	24	25	26	27	28	29

Le message IL FAIT BEAU s'écrit 09-12-00-06-01-09-20-00-02-05-01-21. Quitte à rajouter des zéros à la fin, on peut créer des blocs de 5 chiffres pour obtenir 09120-00601-09200-00205-01210. Le tableau ci-dessous donne la valeur de $x^c \bmod n$ en fonction de x :

x	:	09120	00601	09200	00205	01210
$x^{5015} \bmod 58517$	:	36974	30760	55559	47231	55755.

Le message chiffré sera 36974-30760-55559-47231-55755. On le déchiffrera en calculant $y^{38903} \bmod 58517$ pour chaque bloc y de 5 chiffres. Par exemple $36974^{38903} \bmod 58517 = 09120$ nous redonne la combinaison de lettres IL.

References

[1] L. Adleman, R.L. Rivest & A. Shamir, A method for Obtaining Digital Signatures and Public-Key Cryptosystems, Communications of the ACM, vol.**21**, Number **2**, 1978, pp. 120-126.

[2] P. Arnoux, Minitel, Codage de l'Information et Corps Finis, Pour la Science n^0**125**, mars 1988.

[3] A. Bouvier, Cryptographie Publique, dans "Didactique des Mathématiques, le Dire et le Faire", Cedic-Nathan, 1986, pp. 77-83.

[4] W. Diffie & M. Hellman, New Directions in Cryptography, IEEE Transactions on Information Theory, IT-**22**, **6**, 1976, p. 644-654.

[5] M. Cleyet-Michaud, Le Nombre d'Or, Coll. Que sais-je ? n°1536, PUF, 1978.

[6] G. Cohen, J.-L. Dornstetter , P. Godlewski, Codes Correcteurs d'Erreurs, Une Introduction au Codage Algébrique, Coll. Technique & Scientifique des Télécom, Masson, 1992.

[7] M. Demazure, Cours d'Algèbre, Primalité, Divisibilité, Codes, Editions Cassini, 1997.

[8] J. Itard, Arithmétique et Théorie des Nombres, Coll. Que sais-je ? n°1093, PUF, 1973.

[9] R. Lidl & H. Niederreiter, Finite Fields, Encyclopedia of Mathematics and Its Applications, vol. **20**, Addison-Weysley Publishing Company, 1983.

[10] D. Leglu : La chasse aux nombres premiers, Sciences et Avenir n°422, pp. 70-76.

[11] F.J. Mac William & N.J.A. Sloane, The theory of Error-Correcting Codes, North-Holland Mathematical Library, vol. **16**, 2nd reprint, 1983.

[12] D-J. Mercier, Cryptographie Classique et Cryptographie publique à clé révélée, A.P.M.E.P. **406**, 1996.

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

[14] D-J. Mercier, Codage & Cryptage, APMEP **421**, pp.219-232, 1999.

[15] P. Monot & M. Simon, Habiter le Cybermonde, Les Editions de l'Atelier, 1998.

[16] O. Papini & J. Wolfmann, Algèbre Discrète et Codes Correcteurs, Springer-Verlag, Mathématiques & Applications n°**20**, 1995.

[17] http://www.personaform.com/index.html.old pour le projet PersonaForm et http://www.avatarme.com pour la créatrion d'avatars humains capables d'évoluer dans un monde virtuel.

[18] C. Pomerance, Recent Developments in primality testing, The Mathematical Intelligencer, vol **3**, n°**3**, pp. 47-105, 1981.

[19] J. Querré, Cours d'Algèbre, Maîtrise de Mathématiques, Masson, 1976.

[20] http://www.rocket-library.com/ pour des renseignements sur l'e-Book; et http://www.cytale.com/index.htm pour le Cytale.

[21] J.P. Zanotti, Codage d'un Signal Audionumérique sur un Support à Lecture Optique, Erreurs au décodage et Codes MDS, Mémoire de DEA, Université d'Aix-Marseille II, Faculté des Sciences de Luminy, 1992.