REPERES - IREM . N° 37 - octobre 1999

ARITHMÉTIQUE ET CRYPTOGRAPHIE

Robert NOIRFALISE
IREM de Clermont-Ferrand

I. INTRODUCTION

Depuis les temps historiques les plus reculés, les hommes ont utilisé diverses méthodes pour coder des messages et rendre ceux-ci inintelligibles à des secteurs indésirables. C'est essentiellement dans les domaines militaires et diplomatiques que le cryptage de messages était utilisé. Les Anglais, avec des mathématiciens comme Alan Turing, se sont illustrés pendant la seconde guerre mondiale, en décodant les messages que la marine allemande chiffrait avec la machine Enigma dont le principe remontait à la fin de la première guerre mondiale. Aujourd'hui, le champ d'application de la cryptologie s'est élargi et a trouvé un regain d'actualité avec les problèmes posés par la sécurité des transactions bancaires, des transmissions de fichiers et de bases de données sous forme électronique.

Quelques points de vocabulaire : les termes de *cryptologie* et de *cryptographie* sont souvent utilisés de façon équivalente. Cependant, on réserve parfois le second pour désigner l'ensemble des méthodes de cryptage, le premier ayant alors un sens plus général et pouvant se comprendre comme "science du cryptage". Un autre terme utilisé est celui de *cryptanalyse :* il désigne le travail et les méthodes de ceux qui essaient de casser les codes secrets, de rendre clair ce qui est codé.

Nous entrons, en effet, dans un jeu qui se joue avec trois personnages : le premier envoie un message au second, le problème étant alors d'interdire l'accès de ce message au troisième. On peut, bien sûr, envisager des parades physiques visant à faire que le troisième personnage ne puisse accéder au message, mais l'expérience montre que ces parades sont trop souvent déjouées par l'adversaire. Une façon de procéder est de coder le message de façon à ce que, même intercepté, il ne puisse être lu par le troisième personnage.

Les deux premiers joueurs qui jouent ensemble contre le troisième auront gagné :

– s'ils peuvent coder et décoder facilement des messages et ce dans un temps humainement raisonnable,
– si le troisème personnage, le cryptanalyste, ne peut décoder le message, même avec la puissance calculatoire des ordinateurs, en un temps raisonnable.

Aujourd'hui de tels systèmes existent et sont quasiment inviolables : les temps de calcul pour casser les codes sont astronomiques. Cependant, les cryptanalystes utilisent aujourd'hui le réseau Internet pour faire fonctionner des milliers d'ordinateurs pendant les plages de temps libérés par leurs utilisateurs : si la partie est gagnée provisoirement par les codeurs, les cryptanalystes n'ont pas dit leur dernier mot.

L'arithmétique joue un rôle essentiel dans les méthodes modernes de chiffrement et de cryptanalyse. Le texte qui suit vise à illustrer quelques-unes de ces méthodes en supposant du lecteur seulement les rudiments d'arithmétique d programme de spécialité des classes de T (programme en vigueur à la rentrée 1998) En empruntant un premier exemple d codage aux troupes de Jules César, nou tenterons de montrer la nature des "jeu arithmétiques" du codeur, du décodeur e du cryptanalyste. Le dernier systèm présenté, le RSA, date de 1978 et aujour d'hui son succès fait qu'il tend à supplante les autres systèmes ; on peut montrer san trop de difficultés les principes arithméti ques utilisés pour effectuer codages e décodages. Il serait plus difficile de présen ter une preuve exhaustive de son inviola bilté dans les quelques pages qui suivent e nous renverrons le lecteur qui voudrait e savoir plus vers la littérature spécialisée.

Nous ne pouvons achever cette intro duction sans dire que nous nous somme largement inspiré, dans les lignes qu suivent du chapitre 7 "Cryptologie" d manuel en langue anglaise de Kennet H. Rosen *Elementary number theory an its applications* publié chez Addisson Wesley.

II. LE CHIFFREMENT DE CARACTÈRES ALPHABÉTIQUES

Nous présentons des modes de chiffrement basés sur des principes de congruence arithmétique (n étant un entier naturel, on dit que deux nombres a et b sont congruents modulo n si et seulement si leur différence est un multiple de n, ou encore s'ils ont même reste dans la division euclidienne par n).

Le premier des systèmes de codage que nous présenterons est emprunté à Jules César (100-44 avant Jésus Christ), le dernier a été conçu dans les années 1970 et est encore utilisé aujourd'hui.

Un principe commun à tous ces mode de codage est de transformer chaqu lettre de l'alphabet ou chaque signe d système symbolique utilisé en u nombre ; c'est d'ailleurs ce qui justifi l'usage du terme "chiffrement" pou désigner les techniques de codage. Ains on peut faire correspondre à chaqu lettre de l'alphabet français un nombr compris entre 0 et 25, comme le fait l table n°1 *(page suivante)*.

Bien sûr, si nous voulions coder u message écrit en Russe, en Grec ou e

Lettre	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Équivalent numérique	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

Table n° 1

Hébreu, ou dans n'importe quelle langue utilisant un autre alphabet que le nôtre, nous utiliserions une correspondance analogue entre lettres et nombres. On pourrait aussi inclure les signes de ponctuations, un symbole pour marquer les blancs et aussi ces signes que sont les chiffres eux-mêmes pour représenter des nombres. Pour des raisons de simplicité, nous ne le ferons pas : nous nous restreindrons donc aux vingt-six lettres de notre alphabet : cela est suffisant pour montrer les principes de codage et de décodage usant de congruences arithmétiques. Cela suffit également pour montrer comment un cryptanalyste peut tenter de découvrir la signification du message codé.

1. Le système de codage de Jules César

C'est un système qui, à une lettre du message en clair, associe une lettre le plus souvent différente pour former le message codé. De tels systèmes de codage sont parfois appelés systèmes de codage *"monographique"* : chaque lettre de l'alphabet est tranformée en une autre par substitution.

[Cela peut paraître rudimentaire, mais il y a cependant "26 !" façons d'envisager une telle correspondance entre lettres, puisqu'il y a 26 ! permutations des lettres de l'alphabet.

Or 26 ! = 403291461126605635584000000

$26 ! > 4.10^{26}$: cela montre qu'un cryptanalyste ne pourrait sans doute sérieusement envisager toutes les permutations possibles.

Remarquons, toutefois, qu'une exigence est souvent qu'à une lettre correspond une lettre différente : toutes les permutations en l'occurence, ne conviennent pas. Il convient de ne considérer que celles que l'on appelle des "dérangements" (1). (Si on appelle φ la permutation, on exige pour toute lettre X de l'alphabet que $\varphi(X) \neq X$). Le nombre de dérangements possibles avec 26 caractères reste supérieur à 10^{26}.]

Une exigence du chiffrement, toujours actuelle, est que le codage comme le décodage puissent se faire rapidement.

(1) C'est une notion qui fait l'objet de quelques exercices de combinatoire en Terminale. 26 ! est un nombre "astronomique". On peut se demander légitimement si le fait de ne considérer que des dérangements diminue "beaucoup" le nombre de permutations à envisager pour un cryptanalyste. On peut démontrer que le nombre de dérangements de l'ensemble {1, ... n} est égal à $n! \sum_{p=0}^{n} \frac{(-1)^p}{p!}$.
Pour n = 26, on obtient :
148362637348470135821287825 dérangements, ce qui reste supérieur à 10^{26}.
Les Anglais (avec au sein de leur équipe de cryptanalystes, le mathématicien Alain Turing), pendant la seconde guerre mondiale, se sont servis de cette particularité du système de codage "Enigma" utilisé par les Allemands. Le codage Enigma était bien sûr plus complexe que ceux évoqués dans ce paragraphe ; il ne s'agissait pas de bijections, une même lettre pouvait se transformer en diverses lettres selon son rang dans le message en clair. Cependant une lettre de l'alphabet ne pouvait être codée par elle-même. Les Anglais se servaient de mots probables : ainsi tel centre d'émission spécialisé dans les bulletins de météo, utilisait fréquemment le mot météo. Supposons alors que le message codé WARETOLERZ contienne sous forme codée le mot METEO : la première lettre M ne peut être un W car cela conduirait à chiffrer le second E de météo par un E, ce qui est exclu. Un rapide examen montre qu'il n'y a que deux positions possibles (exemple cité *in* : STERN J : *La science du secret,* Ed. O. Jacob, p. 61).

Alphabet en clair	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
Alphabet codé	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	0	1	2
	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C

Table n° 2

Jules César, qui à l'époque ne devait pas craindre les cryptanalystes, avait un système simple : chaque lettre de l'alphabet était remplacé par la lettre située trois rangs plus loin dans l'alphabet (A → D ; B → E...) quand cela était possible ; aux trois dernières lettres X, Y, Z correspondaient les trois premières A, B, C. Aujourd'hui, on pourrait dire qu'il opérait une permutation circulaire.

Cela peut se décrire facilement en terme de congruence : soit P l'équivalent numérique d'une lettre de l'alphabet en clair et C l'équivalent numérique de la lettre codée correspondante :

On a $\boxed{C \equiv P + 3\ (26)}$ avec $0 \leq C \leq 25$.

La correspondance obtenue ainsi entre l'alphabet et son code est donné dans la table n° 2.

Illustrons la procédure de codage de César avec un exemple. Soit à coder le message :

CE MESSAGE EST TOP SECRET

(Dans les quelques exemples que nous utilisons, nous laissons les blancs des textes pour en faciliter la lecture, mais il conviendrait de les supprimer ! Les blancs seraient autant d'indices mis à la disposition des cryptanalystes)

En passant par les équivalents numériques, on obtient la table ci-dessous.

Le message codé, le "cryptogramme" est donc :

FH PHVVDJH HVW WRS VHFUHW

Certes, les "officiers du chiffre" de Jules César n'usaient sûrement pas de congruence modulo 26 : ici, on voit que le codage peut se faire directement sans passer par les équivalents numériques des lettres de l'alphabet. L'exemple a simplement le mérite de nous rappeler un fait historique et de nous montrer le principe de l'usage de congruences arithmétiques pour élaborer des codages.

Comment décoder ?

Pour coder, on décale le rang des lettres de trois vers le bas, pour décoder on décale les lettres du cryptogramme de trois rangs vers le haut de l'alphabet : les lettres A, B, C, quant à elles sont transformées en X, Y, Z.

Traduisons cette façon de faire

Lettre	C	E		M	E	S	S	A	G	E		E	S	T		T	O	P		S	E	C	R	E	T
Équivalent numérique P	2	4		12	4	18	18	0	6	4		4	18	19		19	14	15		18	4	2	17	4	19
P + 3 (26)	5	7		15	7	21	21	3	9	7		7	21	22		22	17	18		21	7	5	20	7	22

"arithmétiquement" avec les équivalents numériques.

Si C est le symbole d'une lettre codée, P le symbole de la lettre en clair lui correspondant, on a :

$$P \equiv C - 3\ (26)$$

L'identité arithmétique "$P \equiv (P - 3) + 3\ (26)$" nous assure que le décodage fonctionne bien, y compris pour les lettres X, Y, Z et A, B, C !

Exemple :

Soit à décoder le cryptogramme suivant :

OH FUBSWDJH HVW ODUW GH FRGHU XQ PHVVDJH (2)

Faisons-le en utilisant les équivalents numériques (table ci-dessous).

Le message en clair est : "Le cryptage est l'art de coder un message".

Le système de codage de César est un exemple de chiffrement que l'on peut inclure dans une famille de codage par *translation* arithmétique :

$$C \equiv P + K(26), \quad 0 \leq C \leq 25$$

où K est la *clé* du codage. Il y a 26 transformation de ce type incluant la "transformation identique" correspondant à K = 0.

2. Codages par transformations affines

On peut généraliser les transformations précédentes en considérant des transformations du type suivant, que l'on appelle affines pour des raisons évidentes.

$$C \equiv aP + b\ (26), \quad 0 \leq C \leq 25$$

C'est bien une généralisation, car on retrouve les transformations précédentes avec a = 1.

Comme l'on travaille "modulo 26", on peut se contenter de choisir des entiers a et b compris entre 0 et 25.

Q : *Peut-on choisir n'importe quelles valeurs pour a et b ? Sinon, quelles conditions doit-on leur imposer ?*

Il est assez immédiat que toutes valeurs de a et b ne vont pas convenir : si l'on

Message codé	O	H		F	U	B	S	W	D	J	H		H	V	W		O	D	U	W	
P	14	7		5	20	1	18	22	3	9	7		7	21	22		14	3	20	22	
C = P – 3 (26)	11	4		2	17	24	15	19	0	6	4		4	18	19		11	0	17	19	
	L	E		C	R	Y	P	T	A	G	E		E	S	T		L	A	R	T	

(suite)

Message codé	G	H		F	R	G	H	U		X	Q		P	H	V	V	D	J	H
P	6	7		5	17	6	7	20		23	16		15	7	21	21	3	9	7
C = P – 3 (26)	3	4		2	14	3	4	17		20	13		12	4	18	18	0	6	4
	D	E		C	O	D	E	R		U	N		M	E	S	S	A	G	E

(2) Rappelons que que nous laissons les blancs du texte seulement pour faciliter la lecture de la mise en œuvre des algorithmes de codage ou de décodage.

choisit a = 0, toutes les lettres seront codées de la même façon, ce qui rendra le décodage impossible.

On peut aussi voir que si l'on choisit a = 13, alors toutes les lettres ayant un équivalent numérique pair seront codées de la même façon :

Si P = 2k
$C \equiv 13 \times 2 \times k + b\ (26),\ 0 \leq C \leq 25$.
On obtient
$C \equiv b\ (26)$

Un tel système de codage ne convient pas non plus. *On doit exiger qu'à deux lettres différentes au départ correspondent deux lettres codées différentes.* Qu'est-ce que cette condition implique pour les valeurs possibles de a et b ?

[Montrons qu'une condition nécessaire et suffisante pour satisfaire à cette exigence est *"a premier avec 26"*.

Supposons que "a soit premier avec 26" : Soient C et C' tels que $C \equiv C'(26)$. On a donc "$aP + b \equiv aP' + b(26)$", ou encore, après un petit calcul congruentiel, $a(P - P)' \equiv 0(26)$. Cela signifie que 26 divise a(P – P') et a étant premier avec 26, que 26 divise P – P' (d'après le théorème de Gauss). Comme l'on $a - 26 < P - P' < 26$, on en déduit que P = P'. "a premier avec 26" est donc bien une condition suffisante pour qu'à deux lettres distinctes correspondent deux lettres codées distinctes.

Réciproquement supposons que a ne soit pas premier avec 26.

Soit d le pgcd de a et 26. On a d > 1. Soit k l'entier tel que 26 = kd. k est un entier non nul et strictement inférieur à 26. Soit P la lettre dont l'équivalent numérique est k. Montrons alors que A et P, qui sont deux lettres distinctes, sont codées de la même façon : il suffit de montrer que :

$ak + b \equiv a.0 + b(26)$ (l'équivalent numérique de A est 0)

Or il existe k' tel que : a = k'd et ainsi, on a : $ak + b = k'dk + b = 26.k' + b \equiv b\ (26)$, cqfd.]

En définitive, les couples (a, b) satisfaisants sont ceux qui sont tels que "a soit premier avec 26". On remarque, et on pouvait s'y attendre, qu'il n'y a pas de *condition restrictive* sur b.

Q : *Combien y a-t-il de tranformations affines permettant de réaliser un codage ?*

a est un entier compris entre 0 et 25 premier avec 26 : on peut dénombrer (3) directement par exploration systématique les entiers satisfaisants : on trouve 1, 3, 5, 7, 9, 11, 15, 17, 19, 21, 23, 25. Il y en a 12. Comme il y a 26 valeurs possibles pour b, on obtient $12 \times 26 = 312$ transformations affines, ou 311 si on enlève la transformation identique correspondant à a = 1 et b = 0.

Q : *Peut-on décoder ?*

On sait, après avoir fait un choix de a satisfaisant (par exemple a = 7), coder un texte en clair grâce à une transformation affine. Peut-on décoder ? Oui, bien sûr ! On est assuré de pouvoir décoder car la transformation affine assure une correspondance terme à terme entre toutes les lettres de l'alphabet, c'est une bijection et on sait ainsi qu'il existe une transformation réciproque, laquelle d'ailleurs peut se lire dans un tableau exhaustif donnant la transformation (le tableau se lit aussi bien pour coder que pour décoder).

Cependant, on peut se demander si l'opération de décodage peut se faire via l'utilisation d'une opération arithmétique modulo 26. La réponse est oui ; on le montre en utilisant Bézout :

(3) Il existe des méthodes calculatoires, n étant un entier pour déterminer φ(n), le nombre d'entiers premiers avec n qui lui soient inférieurs : la fonction φ qui à n associe φ(n) s'appelle l'indicatrice l'Euler.

a est premier avec 26, donc, d'après Bezout, il existe a' et v tel que :

$$a\,a' + 26v = 1$$

ou, modulo 26,

"Il existe a' tel que a a' ≡ 1(26)"

On dit que *a' est un inverse de a modulo 26.*

Soit C une lettre codée, correspondant en clair à P. On connait C, on veut retrouver P. On sait (quand on est dans le secret du codage) que :

$$C \equiv aP + b\ (26)$$

On a donc

$$C - b \equiv aP\ (26)$$

et en multipliant par a' :

$$a'(C - b) \equiv P\ (26)$$

ou encore $\boxed{P \equiv a'C - a'b\ (26)}$

ce qui montre l'existence d'une transformation affine de décodage, réciproque de la transformation affine de codage.

Q : *La transformation affine de décodage est-elle unique ?* (Si elle ne l'était pas, on pourrait se demander s'il y en a une plus simple que les autres ? Ou s'il y en a une plus rapide que les autres ?)

[Supposons qu'il y en ait deux, la première celle trouvée ci-dessus et une seconde de la forme

$P \equiv a''\,C + b''\ (26)$

On doit, car elle doit décoder, avoir pour tout caractère :

$P \equiv a''\,(aP + b) + b''\ (26)$

Pour P = 0

$0 \equiv a''\,b + b''\ (26)$ soit $b'' \equiv -a''\,b\ (26)$

Pour P = 1

$1 \equiv a''\,(a + b) + b''\ (26)$

$1 \equiv a''\,a + a''\,b + b''\ (26)$

$1 \equiv a''\,a\ (26)$

Or :

$1 \equiv a\,a'\ (26)$

Donc

$a\,a' \equiv a\,a''\ (26)$

ou encore

$a\,(a' - a'') \equiv 0\ (26)$

et comme a est premier avec 26, (Gauss) on a bien alors

$a' \equiv a''\ (26)$

et en conséquence

$b'' \equiv -a''\,b \equiv -a'\,b \equiv b'\ (26).$]

Remarque :

On peut reprendre la démonstration précédente pour montrer aussi qu'une transformation affine de codage est unique (il n'en existe pas d'autre produisant le même codage).

Exemple : a = 7 et b = 10

$$\boxed{C \equiv 7P + 10\ (26)}$$

Pour trouver a', une des clés du décodage, on peut soit utiliser "l'algorithme d'Euclide étendu" permettant de trouver des valeurs u et v de la formule de Bézout telle que :

$$7u + 26v = 1$$

soit considérer les multiples de 7, modulo 26 ; on sait qu'il en existe un et un seul satisfaisant à la relation

$$7.\,a' \equiv 1\ (26).$$

[Recherchons a' en utilisant l'algorithme d'Euclide étendu : le principe est simple ; on procède classiquement, avec l'algorithme d'Euclide, à la recherche du P.G.C.D. de deux nombres a et b, et parallèlement (on étend l'algorithme) on écrit les

restes successifs en fonction de a et b :

$26 = 3 \times 7 + 5$

$5 = 26 - 3 \times 7$

$7 = 5 + 2$

$2 = 7 - 5 = 7 - (26 - 3 \times 7) = 4 \times 7 - 26$

$5 = 2 \times 2 + 1$

$1 = 5 - 2 \times 2 = (26 - 3 \times 7) - 2(4 \times 7 - 26)$

$= -11 \times 7 + 3 \times 26$

On obtient une relation, à la "Bézout" :

$1 = -11 \times 7 + 3 \times 26$

ce qui donne, modulo 26 :

$-11 \times 7 \equiv 1\ (26)$

ou encore $15 \times 7 \equiv 1\ (26)$

On obtient a' = 15.]

La transformation de décodage est donc

$P \equiv 15(C - 10)\ (26)$

ou $P \equiv 15C + 6\ (26)$ (car $-150 \equiv 6\ (26)$)

La correspondance entre lettre (avec a = 7 et b = 10) est donnée dans la table 3.

On peut remarquer que la transformation affine opérée n'est pas "un dérangement" car la lettre H est sa propre transformée. Une question que nous ne traiterons pas ici, serait de savoir quelles conditions supplémentaires il conviendrait d'imposer aux paramètres a et b pour que la transformation soit bien un dérangement. On peut remarquer que pour le codage, cela n'a pas grande importance, bien que cette condition, historiquement, ait souvent été requise (le chiffrage allemand, par exemple, pendant la seconde guerre mondiale, procédait ainsi, ce qui, on l'a vu, était en fait une faiblesse du système).

Nous allons, dans le paragraphe suivant, nous servir de ce mode de chiffrement pour voir comment pouvaient procéder ceux qui, sans connaître les clés de codage et de décodage, tentaient de décrypter ou de déchiffrer les textes codés, ceux que la littérature moderne désigne par le terme de "cryptanalystes".

3. Un exemple de cryptanalyse

La cryptanalyse désigne l'ensemble des procédés pouvant être mis en œuvre pour percer à jour un texte codé, sans connaître, *a priori*, la ou les clés de codage et de décodage.

Plaçons-nous dans le cas où le cryptanalyste sait, cependant, que le mode de codage est une transformation affine du genre de celles vues dans le paragraphe précédent. Il lui reste à découvrir les paramètres a et b ayant servis à faire la transformation. Il pourra alors, sans difficulté, procéder au décodage.

On l'a vu, il y a trois cent onze transformations affines possibles : il est toujours possible "théoriquement" d'envisager chacune de ces transformations et d'examiner, avec le texte à décoder ce que chacune d'entre elles produit comme texte supposé en clair. On le devine, cela va être bien fastidieux ! C'est encore un principe retenu aujourd'hui ; la recherche d'une clé de décodage est toujours réalisable théoriquement, mais les "codeurs" choisissent des systèmes de code tels que la découverte

Alphabet en clair	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
Alphabet codé	10	17	24	5	12	19	0	7	14	21	2	9	16	23	4	11	18	25	6	13	20	1	8	15	22	3
	K	R	Y	F	M	T	A	H	O	V	C	J	Q	X	E	L	S	Z	G	N	U	B	I	P	W	D

Table n° 3

de la clé demande plusieurs années de calculs à des ordinateurs puissants.
Prenons un exemple : supposons que nous ayons à rendre clair le message suivant :

YM QMGGKAM MGN NEL GMYZMN

en sachant qu'il a été codé au moyen d'une tranformation affine.

A ce stade, le cryptanalyste dispose d'une arme redoutable qui peut lui éviter bien des heures de labeur fastidieux : *les lettres de l'alphabet n'apparaissent pas avec la même fréquence dans une langue donnée !* Certaines sont rares, d'autres plus fréquentes. C'est ainsi qu'en français, la lettre la plus fréquente est le E suivi du S puis du A.

Avec un peu de chance, cet ordre "fréquentiel" va être suivi, à quelque chose près, par les lettres du texte à décoder : ceci sera d'autant plus vrai que le texte sera long et, aussi, d'autant plus que le texte à décoder appartiendra à une famille analogue aux textes ayant servi à établir la table des fréquences.

Ici, dans le court message que nous avons à décrypter, la lettre la plus fréquente est le M qui apparaît six fois, suivi du G qui apparaît quatre fois.

Faisons alors l'hypothèse que le M correspond au E et le G au S.

Passons aux équivalents numériques :

M → 12 E → 4
G → 6S → 18

Les paramètres de codage, a et b doivent alors vérifier les deux équations (deux comme le nombre de paramètres à déterminer)

$12 \equiv 4a + b\ (26)$
(équation traduisant la transformation du E en M)

$6 \equiv 18a + b\ (26)$
(équation traduisant la transformation du S en G)

Nous sommes donc ramenés à résoudre un système de 2 équations à 2 inconnues modulo (26). Ces équations sont souvent désignées par le terme de Diophantiennes (du nom du mathématicien grec Diophante qui a beaucoup travaillé à la résolution de ce genre d'équations).

Les règles élémentaires de calcul sur des congruences de même module nous permettent d'opérer quasiment comme pour la résolution d'équations classiques : nous ne pouvons faire de divisions, mais nous pouvons faire des multiplications équivalentes à des divisions.

[$12 \equiv 4a + b\ (26)$ (1)
$6 \equiv 18a + b\ (26)$ (2)
En soustrayant (1) de (2), nous obtenons :
$14a \equiv -6 \equiv 20\ (26)$
Si a vérifie cette équation, il existe k Z tel que $14a = 20 + 26k$ soit en divisant par 2 :
$7a \equiv 10\ (13)$ (Ici, nous avons pris la précaution de passer par un calcul dans Z pour assurer la légitimité d'une division).
Multiplions par 2 qui est un inverse de 7 modulo 13 : $14a \equiv 20\ (13)$, soit $a \equiv 7\ (13)$
Modulo 26, cela nous donne a = 7 ou a = 7 + 13 = 20.
Comme de plus a doit être premier avec 26 pour être admissible, cela nous donne comme seule solution possible a = 7. De (1) ou (2), on déduit que b = 10.]

On retrouve ainsi une transformation de codage que nous savons déjà décoder : le message crypté devient en clair :

CE MESSAGE EST TOP SECRET

L'exemple précédent montre qu'une

analyse statistique des fréquences de lettres permet facilement de briser un cryptogramme quand on sait que celui-ci est le fruit d'une transformation affine monographique.

De fait, si le travail est très simple pour le cryptanalyste, dès lors qu'il sait que le codage est monographique et affine, son travail n'est pas beaucoup plus complexe s'il ne dispose pas de ce type d'hypothèses. N'importe quel message, pas trop court, codé de façon monographique ne résiste pas longtemps à l'analyse statistique des fréqences de lettres : le nombre de bijections possibles, bien qu'astronomique, ne protège pas les codeurs de ce type d'analyse.

La lutte se trouve engagée entre "codeurs" et "cryptanalystes". Les premiers doivent anticiper le travail des seconds pour essayer de leur rendre la tâche mal-aisée, voire impossible.

On peut citer, avant d'envisager des moyens plus récents, un système de codage proposé par un Auvergnat de Saint-Pourçain sur Sioule, Blaise de Vigenère (4) (1523-1596) qui a publié en particulier un *Traité des chiffres* (1586) qui est à la fois un manuel d'épigraphie et un véritable livre de cryptographie diplomatique.

Son procédé emprunte à celui de César en le complexifiant : on opère des translations sur les lettres du texte à coder en fonction d'un mot-clé indiquant les translations à opérer selon le rang des lettres dans le texte. Illustrons-en le principe avec un exemple : le mot-clé est SECRET. Cela veut dire, il y a 6 lettres dans le mot secret, que les lettres de rang 1 modulo 6 dans le texte vont subir une translation modulo 18 (car l'équivalent numérique de S est 18), puis les lettres de rang 2 modulo 6 vont subir une tranlation modulo 4 (car l'équivalent numérique de E est 4) etc. Le mot-clé est simple à transmettre, les modes de codage et de décodage sont simples. C'est encore ce principe de codage qui était utilisé par les Allemands pendant la seconde guerre mondiale et qui était automatisé avec la machine Enigma. Le cryptanalyste est en difficulté pour utiliser la fréquence des lettres, du moins tant qu'il n'a pas d'autres indications sur la longueur du mot-clé. Il semble que d'autres régularités de la langue ou la recherche de mots probables dans un texte permettent de deviner cette longueur... c'est ce que les Anglais ont réussi à faire à plusieurs reprises.

III. LE CODAGE POLYGRAPHIQUE OU PAR BLOC

Nous avons vu que les codages monographiques résistent mal aux cryptanalystes, lesquels peuvent disposer de données sur les fréquences d'apparition des lettres d'une langue donnée. Pour pallier cette faiblesse, des systèmes de codage ont été développés, qui substituent à un bloc de lettres, un autre bloc de lettres de même longueur. De tels systèmes de codage sont dits *polygraphiques* ou désignés par le terme de *codages par blocs.*

Nous allons présenter un tel système,

(4) Le lycée de Saint-Pourçain porte son nom.

basé sur le calcul congruentiel et développé par Hill (5) dans les années 1930.

N'oublions pas que si un système doit résister aux efforts des cryptanalystes, il doit rester simple d'usage pour le codeur comme pour le décodeur.

1. Le système de codage

Pour illustrer simplement le système de Hill, prenons le cas le plus simple (on pourra le complexifier en augmentant la longueur des blocs), celui où les blocs sont formés de 2 lettres : on parle de *codage digraphique.*

La première étape est tout d'abord de répartir les lettres du message à coder en blocs de 2 lettres (en ajoutant éventuellement une lettre supplémentaire, un X, à la dernière lettre pour former un bloc adéquat).

Illustrons le procédé avec un exemple :

ENVOYEZ L'ARGENT

est scindé en bloc :

EN VO YE ZL AR GE NT

Ensuite chaque lettre est transformée en son équivalent numérique

4.13 21.14 24.4 25.11 0.17 6.4 13.19

Chaque bloc de 2 nombres $P_1\ P_2$ est alors converti en un bloc de 2 lettres codé $C_1\ C_2$ selon un procédé arithmétique comme le suivant :

$$C_1 \equiv 5P_1 + 17P_2\ (26)$$
$$C_2 \equiv 4P_1 + 15P_2\ (26)$$

Par exemple, le bloc 4.13 est converti en HD car :

$$C_1 \equiv 5 \times 4 + 17 \times 13 \equiv 7\ (26)$$
$$C_2 \equiv 4 \times 4 + 15 \times 13 \equiv 3\ (26)$$

On obtient le cryptogramme suivant :

HD FI GA AF DV UG YZ

2. Le décodage

On peut voir apparaître un phénomène qui sera exploité par les systèmes de codage les plus récents : ce n'est pas parce que l'on sait coder que l'on sait automatiquement décoder. En effet, contrairement au cas des transformations affines ou des translations à la César, un simple tableau ne saurait suffire à obtenir le moyen de décoder les messages cryptés à la mode Hill ! (6).

Ici, cependant, l'arithmétique vient à notre secours ; le théorème suivant nous délivre un système de décodage arithmétique :

[*Théorème :*
Soient a, b, c, d, e, f et m des entiers avec $m > 0$, tels que PGCD $(\Delta, m) = 1$ avec $\Delta = ad - bc$.
Alors, le système d'équations congruentes (ou diophantiennes)

$$ax + by \equiv e\ (m)$$
$$cx + dy \equiv f\ (m)$$

admet une unique solution modulo m donnée par :

(5) HILL L.S. : "Concerning certain linear transformation apparatus of cryptography. *American mathematical monthly* - Vol. 38 (1931) pp. 135-154.

(6) On pourrait dresser un tableau des correspondances, mais dans le cas les plus simple du codage digraphique ce serait déjà un tableau avec $676 = 26^2$ entrées, ce qui le rend d'un usage mal aisé.

$x \equiv \overline{\Delta}(de - bf)\ (m)$

$y \equiv \overline{\Delta}(af - ce)\ (m)$

où $\overline{\Delta}$ est un inverse de Δ modulo m.

Preuve :

Multiplions la première congruence par d et la seconde par b, pour obtenir

$adx + bdy \equiv de\ (m)$

$bcx + bdy \equiv bf\ (m)$

Puis, ôtant la seconde congruence de la première, nous trouvons :

$(ad - bc)x \equiv de - bf\ (m)$

Ou, puisque $\Delta = ad - bc$,

$\Delta x \equiv de - bf\ (m)$

Δ est premier avec m, donc inversible modulo (m) : en effet, d'après Bézout, il existe $\overline{\Delta}$ et v tel que $\Delta.\overline{\Delta} + mv = 1$ ce qui donne bien $\Delta\overline{\Delta} \equiv 1\ (m)$. $\overline{\Delta}$ est un inverse de Δ modulo m.

Multiplions les deux membres de la dernière congruence obtenue par $\overline{\Delta}$: nous avons

$x \equiv \overline{\Delta}(de - bf)\ (m)$

De façon identique, on obtiendrait

$y \equiv \overline{\Delta}(af - ce)\ (m)$

Nous avons montré que si (x, y) est une solution du système étudié, alors :

$x \equiv \overline{\Delta}(de - bf)\ (m)$ et $y \equiv \overline{\Delta}(af - ce)\ (m)$

On peut vérifier (démarche de synthèse, *i.e.* de condition suffisante) qu'un tel couple est bien solution du système :

$ax + by \equiv a\overline{\Delta}(de - bf) + b\overline{\Delta}(af - ce)\ (26)$

$\equiv \overline{\Delta}(ade - abf + abf - bce)\ (26)$

$\equiv \overline{\Delta}e(ad - bc)\ (26)$

$\equiv \overline{\Delta}.\Delta.e\ (26)$

$\equiv e\ (26)$

De la même façon, on vérifierait que

$cx + dy \equiv f\ (26)$ CQFD.]

Revenons au décodage lié à la méthode de Hill ;

On applique le théorème précédent avec a = 5 b = 17 c = 4 d = 15, m = 26.

On a $\Delta = ad - bc = 5 \times 15 - 17 \times 4 = 7$. Δ et 26 sont bien premiers entre eux.

Considérons un bloc codé $C_1\ C_2$: l'équivalent numérique de C_1 joue le rôle de e, et celui de C_2, le rôle de f.

Le problème est de trouver P_1 et P_2, connaissant C_1 et C_2 et sachant que :

$5P_1 + 17P_2 \equiv C_1\ (26)$
$4P_1 + 15P_2 \equiv C_2\ (26)$

Cela revient à résoudre un système de 2 équations à 2 inconnues dont le théorème précédent donne la solution unique, à savoir :

$P_1 \equiv 17C_1 + 5C_2\ (26)$
$P_2 \equiv 18C_1 + 23C_2\ (26)$

En effet : $\Delta = 7$ d'où $\overline{\Delta} = 15$ et

$P1 \equiv \overline{\Delta}(15C_1 - 17C_2)\ (26)$
$\equiv 15(15C_1 - 17C_2)\ (26)$
$\equiv 17C_1 - 21C_2\ (26)$
$\equiv 17C_1 + 5C_2\ (26)$

$P2 \equiv \overline{\Delta}(5C_2 - 4C_1)\ (26)$
$\equiv 15(5C_2 - 4C_1)\ (26)$
$\equiv 23C_2 - 8\ C_1\ (26)$
$\equiv 18C_1 + 23C_2\ (26)$

3. La riposte du cryptanalyste

Malheureusement pour le système de Hill, les particularités statistiques de la langue vont là aussi venir au secours des cryptanalystes : les couples de lettres n'apparaissent pas avec les mêmes fréquences. C'est ainsi qu'en Anglais (7) les

(7) Ne disposant pas de l'équivalent français, nous nous en tenons à un exemple du texte anglais de Kenneth Rosen.

couples les plus fréquents sont dans l'ordre TH et HE. Supposons alors que l'on dispose d'un texte codé et que l'on sache que le mode de codage est un mode digraphique arithmétique dû à Hill, et de plus que les couples de lettres les plus fréquents soient KX et VZ. On peut alors faire l'hypothèse qu'ils correspondent respectivement aux couples TH et HE. En passant aux équivalents numériques, cela signifie que les couples 19.7 et 7.4 sont transformés en 10.23 et 21.25 respectivement.

En conséquence, les paramètres du codage a, b, c, d doivent remplir les 4 équations :

$$(S)\begin{cases}19a+7b\equiv 10(26)\,(1)\\19c+7d\equiv 23(26)(2)\\7a+4b\equiv 21\,(26)(3)\\7c+4d\equiv 25\,(26)(4)\end{cases}$$

C'est un système que l'on peut séparer en deux sous-systèmes en isolant les équations relatives aux paramètres a et b d'une part et celles relatives à c et d d'autre part.

On obtient ainsi :

$$(S_1)\begin{cases}19a+7b\equiv 10(26)\,(1)\\7a+4b\equiv 21\,(26)(3)\end{cases}$$

$$\text{et } (S_2)\begin{cases}19c+7d\equiv 23(26)(2)\\7c+4d\equiv 25\,(26)(4)\end{cases}$$

Ces deux systèmes peuvent se résoudre classiquement ou en utilisant le théorème vu ci-dessus : on obtient a = 23 ; b =17 ; c = 21 ; d = 2.

On peut vérifier aussi que $\Delta = ad - bc$ est bien premier avec 26.

Ainsi, on a une clé de codage possible : on sait déterminer la clé de décodage.

Si le texte décodé n'est pas un texte cohérent, on procède alors à d'autres hypothèses de correspondance entre couples de lettres, toujours en faisant des comparaisons de fréquences entre couples de lettres dans le cryptogramme et couples dans la langue du texte.

IV. CODAGES PAR EXPONENTIATION ARITHMÉTIQUE

Nous présentons un mode de codage basé sur des calculs d'exponentiels modulo un nombre premier. Nous verrons que ce procédé de chiffrement résiste aux efforts des cryptanalystes tout en autorisant des codages et décodages rapides.

1. Le principe du codage

On utilise un entier p impair et premier, et e un autre entier naturel, la clé du codage, qui soit premier avec p – 1. (PGCD (e, p – 1) = 1).

Pour coder un message, on traduit tout d'abord, comme on l'a fait jusqu'ici, chaque lettre de l'alphabet en son équivalent numérique : on impose cependant que chaque lettre soit codée par deux chiffres comme l'indique la table 4 *(page suivante)*.

On reprend le principe de regroupement en bloc utilisé par Hill. On groupe les chiffres obtenus par blocs de 2 m chiffres.

On choisit m de telle sorte que deux nombres distincts formés de 2 m chiffres associés à m lettres correspondent à deux nombres distincts modulo p. De plus on choisit m le plus grand possible.

Lettre	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Équivalent numérique	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

Table n° 4

En, d'autres termes, pour coder, on va coder des blocs de 2m chiffres considérés comme des nombres en passant "modulo p". Une façon de s'assurer qu'à 2 blocs distincts correspondent bien 2 équivalents numériques distincts, est que le plus grand nombre possible formé avec un bloc de 2m chiffres associé à m lettres soit inférieur à p.

Ainsi

si $25 < p < 2525$, on a $m = 1$
si $2525 < p < 252525$, on a $m = 2$
si $252525 < p < 25252525$, on a $m = 3$ etc.

(Dans les inégalités précédentes, on ne saurait avoir une inégalité au sens large car p est premier).

Ensuite pour chaque bloc P, qui est donc un nombre formé de 2m chiffres, on forme un bloc "codé" C en utilisant la relation :

$$C \equiv P^e \ (p), \ 0 \leq C < p.$$

Remarquons que la condition $0 \leq C < p$ n'implique pas nécessairement que C soit un nombre pouvant s'écrire avec 2m chiffres. Le choix de m, en fonction de p, impose cependant que C s'écrive au plus avec 2(m + 1) chiffres (ceci dans le cas où p s'écrit lui-même avec 2(m + 1) chiffres).

Illustrons cette technique de codage à l'aide d'un exemple :

Exemple :

Choisissons p = 2633 et e = 29. Des tables existent permettant de vérifier que 2633 est bien un nombre premier ; 29 étant aussi un nombre premier, on a bien 2632 et 29 premiers entre eux.

Soit alors à coder le message :

CECI EST UN EXEMPLE DE
CODAGE EXPONENTIEL

Commençons par transformer chaque lettre en son équivalent numérique et groupons les chiffres obtenus par blocs de 4. (Ici, le choix de p donne m = 2).

0204 0208 0418 1920 1304 2304
1215 1104 0304 0214 0300 0604
0423 1514 1304 1319 0804 1123

Notons que nous avons, pour avoir un nombre pair de lettres, ajouté X au dernier bloc de lettres.

Ensuite, nous transformons chaque bloc numérique, en un bloc codé C en utilisant l'opération :

$$C \equiv P^{29} \ (2633) \qquad 0 \leq C < 2633$$

Par exemple, pour transformer le premier bloc, 0204, nous calculons :

$$C \equiv 0204^{29} \equiv 1566 \ (2633)$$

En procédant de même, avec les autres blocs, on obtient le cryptogramme suivant :

1566 1846	0177	0071	0960	2241
2059 1684	2435	0356	1144	2441
2437 1759	0960	0815	0951	1133

Q : *Le codage exponentiel n'est-il pas trop coûteux en calculs ?*

La réponse est non (8) dès lors que l'on dispose de machines électroniques ; contrairement à ce que l'exponentiel pourrait laisser supposer, le nombre de calculs n'est pas si grand que cela :

Reprenons l'exemple du calcul de 0204^{29} (2633).

On procède à des élévations successives au carré modulo 2633

$204^2 \equiv 2121\ (2633)$
$204^4 \equiv (2121)^2 \equiv 1477\ (2633)$
$204^8 \equiv (1477)^2 \equiv 1405\ (2633)$
$204^{16} \equiv (1405)^2 \equiv 1908\ (2633)$

Et en remarquant que

$$204^{29} = 204^{16+8+4+1} = 204^{16} \times 204^8 \times 204^4 \times 204,$$

il reste à opérer 3 multiplications modulo (2633), ce qui donne :

$$1908 \times 1405 \times 1477 \times 204 \equiv 1566\ (2633)$$

Pour élever à la puissance 29, il suffit comme nous l'avons fait d'effectuer 7 multiplications (4 pour les élévations successives au carré et 3 pour la multiplication finale).

Q : *On sait donc coder sans trop de difficulté, peut-on décoder de même ?*

(8) Nous avons opéré avec une TI92 pour effectuer le codage du message présenté ci-dessus : la réponse de la TI92 pour une opération est instantanée !

2. Le décodage

Pour comprendre le principe du décodage, on a besoin du "petit théorème de Fermat" qui s'énonce ainsi :

> Soit p un entier premier et a un entier positif et inférieur à p, alors
> $a^{p-1} \equiv 1\ (p)$

[*Preuve* :
considérons les mutliples de a : a, 2a,...,(p – 1)a et leurs restes r_1, r_2... r_{p-1} modulo p.
Montrons que tous les restes r_i pour i = 1...p-1 sont distincts 2 à 2.
Supposons le contraire, c'est-à-dire qu'il existe i et j avec $i \neq j$ et tels que $r_i = r_j$.
On a en conséquence $ia \equiv ja\ (p)$ et donc $(j - i)a \equiv 0(p)$.
On peut toujours supposer, sans perte de généralité, que j > i ; j-i est donc un entier compris entre 1 et p – 1, donc premier avec p. Or, a est aussi premier avec p (p étant premier, il en est ainsi de tous les entiers naturels non nuls qui lui sont inférieurs).
D'après Gauss, ceci est impossible, et donc en nécessité, on doit avoir $ia \neq ja(p)$ et donc $r_i \neq r_j$.
Or, il y a p – 1, restes possibles, non nuls, modulo p, à savoir 1, 2... p – 1.
Les différents restes ri étant distincts et leur nombre étant p-1, ils forment donc l'ensemble 1,2... p – 1 et on a :

$$\prod_{i=1}^{p-1} r_i \equiv 1 \times 2 \times ... \times p - 1 = (p - 1)!\ (p)$$

et donc

$$\prod_{i=1}^{p-1} ia \equiv (p - 1)!\ (p)$$

Or :

$$\prod_{i=1}^{p-1} ia = (p - 1)!\ a^{p-1}$$

On a donc $(p-1)!\, a^{p-1} \equiv (p-1)!\ (p)$
et ainsi $(p-1)!\,(a^{p-1}-1) \equiv 0\ (p)$
Il en découle que p divise $(p-1)!\,(a^{p-1}-1)$, mais c'est une conséquence du théorème de Gauss, p est premier avec $(p-1)!$ et donc, toujours selon Gauss, il divise $a^{p-1}-1$, ce qui se traduit par le résultat annoncé :
$a^{p-1} \equiv 1\ (p)$ CQFD.]

La clé de codage e est un entier premier avec $p-1$, et donc d'après Bézout, cette fois-ci, il existe d et k entiers tels que :

$$ed + k(p-1) = 1$$

On peut toujours choisir d tel que $0 < d < p-1$. Dans ce cas, il est assez évident que k doit être négatif et qu'on peut alors écrire, en posant $k' = -k$

$$\boxed{ed = 1 + k'(p-1)} \quad \text{avec } k' > 0.$$

On remarque que d est un inverse de e modulo $(p-1)$

C étant un bloc codé, il suffit alors pour retrouver le bloc P initial, d'élever C à la puissance d modulo p.

d est la clé de décodage : on a

$$\boxed{C^d \equiv P\ (p)}$$

Montrons qu'il en est bien ainsi :

On sait que $C \equiv P^e\ (p)$

et donc

$$C^d \equiv (P^e)^d \equiv P^{ed} \equiv P^{(1+k'(p-1))}\ (p)$$
$$\equiv P.\ P^{k'(p-1)}\ (p)$$
$$\equiv P.\ (P^{(p-1)})^{k'}\ (p)$$

Or $P^{p-1} \equiv 1\ (p)$ d'après Fermat.

d'où en conclusion de ces quelques calculs modulo p :

$$C^d \equiv P\ (p)$$

ce qui assure du décodage ; on retrouve le bloc de chiffres initial, ce qui permet alors de reconstituer le texte en clair.

Reprenons l'exemple numérique que nous avons commencé à traiter avec $p = 2633$ et $e = 29$. Quelle est la clé de décodage ?

On peut appliquer l'algorithme d'Euclide étendu pour la trouver. Faisons-le car cela montre que même si l'on a à opérer un tel calcul pour le décodage, le calcul peut être rapide :

$2632 = 90 \times 29 + 22$
$22 = 2632 - 90 \times 29$
$29 = 22 + 7$
$7 = 29 - 22 = 91 \times 29 - 1 \times 2632$
$22 = 3 \times 7 + 1$
$1 = (2632 - 90 \times 29) - 3(91 \times 29 - 2632)$

d'où $1 = 4 \times 2632 - 363 \times 29$

soit modulo (2632) : $1 \equiv -363 \times 29\ (2632)$

ou $1 \equiv 2269 \times 29\ (2632)$

la clé d est : $\boxed{d = 2269.}$

Q : *On peut se demander, la clé de décodage étant connue, si le coût en calculs, du décodage est raisonnable ?*

La réponse est oui ! Le coût est raisonnable et, ce, pour les mêmes raisons que celles nous ayant conduit à déclarer que le coût du codage était raisonnable. Le procédé de calcul pour décoder est du même type : on pratique une exponentiation modulo p. Cependant, d étant, relativement à p, nettement plus grand que e, on peut se demander si le nombre de multiplications à opérer modulo p est beaucoup plus grand.

Ici, nous avons :

$$2269 = 2^{11} + 2^7 + 2^6 + 2^4 + 2^3 + 2^2 + 1$$

et donc pour tout bloc de chiffres C codé, nous avons :

$$C^{2269} \quad C^{2^{11}}.C^{2^7}.C^{2^6}.C^{2^4}.C^{2^3}.C^{2^2}.C \ (p)$$

Il y a 6 multiplications à opérer, celles indiquées dans l'écriture de la ligne ci-dessus une fois calculé modulo p, les quantités C^2, C^{2^2}... $C^{2^{11}}$. Or, pour avoir celles-ci, il suffit de procéder à 11 élévations au carré modulo p. Pour comprendre cela, il suffit de remarquer que :

Pour tout n $(C^{2^n})^2 = C^{2^n} \cdot C^{2^n} = C^{2^n + 2^n} = C^{2^{n+1}}$

On a donc

$C^{2^{11}} = (C^{2^{10}})^2$, $C^{2^{10}} = (C^{2^9})^2$ etc.

Malgré la valeur "élevée" de d, il n'y a que 17 multiplications à opérer (multiplications modulo p) : 11 élévations au carré suivies de 6 multiplications.

[On peut d'ailleurs, assez facilement, déterminer un majorant du nombre de multiplications impliquées par une exponentiation modulo un entier.

Considérons le cas général d'une exponentiation du type P^N (p) avec $0 < N < p$.

Soit k l'entier vérifiant $2^{k-1} \leq N < 2^k$ (1)

Cela signifie que, au maximum, N prend la valeur $2^{k-1} + 2^{k-2} + ... + 2^i + ... + 2 + 1$ (ie $2^k - 1$).

$P^N = p^{2^{k-1}}.p^{2^{k-2}} ... p^{2^i} ... p^2.p$ correspond au cas le plus défavorable, celui où il y a le maximum d'opérations à faire (sachant que $2^{k-1} \leq N < 2^k$).

Comme nous l'avons vu tout à l'heure, pour déterminer les diverses puissances du type p^{2^i} il suffit d'opérer (k – 1) élévations au carré, donc il suffit de faire (k – 1) multiplications.

Pour terminer le calcul, il reste à multiplier les diverses puissances... p^{2^i}...entre elles. Il y a au plus (k – 1) multiplications à faire.

En tout, au plus 2k – 2 multiplications à opérer.

Nous avons un majorant exprimé en fonction de k ; il convient de l'exprimer en fonction de N et p. Or, on peut, en passant au logarithme à base 2, modifier l'inégalité

$2^{k-1} \leq N < 2^k$

qui devient

$k - 1 \leq \log2\ N < k$.

Nous pouvons donc majorer le nombre de multiplications nécessaires par "2log2 N" (9).

Plus précisément, une détermination du nombre d'opérations élémentaires électroniques à effectuer pour faire une multiplication modulo p, nous donnant un majorant de "l'ordre" de $(\log2\ p)^2$, nous pouvons ici majorer le nombre d'opérations élémentaires nécessaires à une exponentiation par un nombre de l'ordre de $(\log2\ p)^2\ \log2\ N$ ou encore par $(\log2\ p)^3$ (puisque $N < p$). Si le nombre p s'écrit en base décimale avec n chiffres, nous avons $p < 10^n$, et donc, on peut majorer le nombre d'opérations à faire par un nombre de l'ordre de $(\log2\ 10^n)^3$ soit par un nombre de l'ordre de n^3.]

3. La résistance de ce type de codage à la cryptanalyse

Si le cryptanalyste connaît le type de codage (par exponentiation) et le nombre premier p, il lui reste à découvrir une des clés e ou d.

Peut-il le faire en utilisant les mêmes ressources que celles vues précédemment, c'est-à-dire des données statistiques qui

(9) Ce passage au logarithme permet de déterminer le nombre de chiffres avec lequel est écrit un nombre dans une base donnée. Or, le nombre d'opérations dans un calcul dépend souvent, non pas du nombre, mais du nombre de chiffres servant à l'écrire, ceci explique que des expressions donnant des appréciations de coût en calcul contiennent le plus souvent des logarithmes.

porteront, ici, sur des fréquences de blocs de lettres. Plaçons-nous dans un cas favorable (mais tout à fait possible) et supposons que le cryptanalyste peut faire l'hypothèse qu'un bloc codé C est la traduction d'un bloc P en clair.

Il lui reste à résoudre l'équation $C \equiv P^e$ (p) où l'inconnue est e alors que C, P et p sont connus !

Banalement, pourrait-on dire, il lui suffit de faire varier e de 1 à p – 1. C'est un algorithme grossier qui en moyenne va nécessiter p/2 fois plus de calculs que le calcul nécessité par un codage. Il existe des algorithmes permettant de trouver e qui sont, certes, plus efficaces, mais dont le coût reste prohibitif en temps dès lors que p est assez grand. Si l'on se réfère, non plus à p mais à n le nombre de chiffres nécessaires pour écrire p en base 10, le coût en calcul pour découvrir e est de l'ordre de 10^n. Là est toute la différence ; alors qu'il faut quelques centièmes de seconde pour coder ou décoder un bloc de lettres, il faudrait un temps considérable pour découvrir la clé "e" : C'est ainsi qu'il faudrait approximativement 74 ans de calcul à un ordinateur puissant pour trouver e lorsque p est un nombre écrit avec 100 chiffres décimaux, ou encore, approximativement 4.10^9 années pour un nombre p écrit avec 200 chiffres, et ce, avec les algorithmes les plus performants connus à ce jour.

Rappelons que le coût du codage ou du décodage en calcul est de l'ordre de n^3, n étant le nombre de chiffres en base décimale de p. Si donc, n = 100, à une constante multiplicative près, il convient d'opérer $100^3 = 10^6$ opérations pour effectuer un codage ou un décodage, ce qui peut se faire en 1/100 seconde par les ordinateurs classiques actuels.

Si l'on augmente le nombre de chiffres de p et que l'on passe à 200, on multiplie le coût du codage par $2^3 = 8$, ce qui reste tout à fait accessible.

Il convient, bien sûr, de faire une remarque sur le fait que l'on "raisonne" sur un ordre de grandeur, ce qui est légitime lorsqu'on considère une variable mais n'a pas de sens, en toute rigueur, lorsqu'on donne, comme nous venons de le faire, une valeur spécifique à n ($n = 10^6$). Il se trouve que, *dans la pratique,* les constantes multiplicatives qu'il conviendrait de faire intervenir sont suffisamment petites pour que les comparaisons effectuées en terme de coût demeurent exactes.

En résumé : n étant le nombre de chiffres de p, le coût d'un codage ou d'un décodage est de l'ordre de n^3. On dit que ce coût est polynômial en n, alors que le coût de recherche de la clé est de l'ordre de 10^n, il est exponentiel (10) en n.

[Pour tenter de saisir plus avant la signification de cette comparaison, résolvons le petit problème suivant :

Supposons que demain, on construise des ordinateurs plus puissants, travaillant 100 fois plus vite que ceux d'aujourd'hui. De combien de chiffres faut-il augmenter, en écriture décimale, le nombre de chiffres de p pour avoir la même sécurité que celle obtenue aujourd'hui. En d'autres termes, on veut modifier p de telle sorte que le temps requis pour "casser" le code reste le même.

Faisons comme si le temps est proportionnel à 10^n, n étant le nombre de chiffres de p.

(10) En vérité, certains algorithmes permettent de faire mieux, mais fondamentalement cela ne change pas la nature des coûts : La fonction logarithme modulo p, réciproque de l'exponentielle reste, à ce jour, difficile à calculer : les calculs sont coûteux en temps !

On a donc, le temps To, étant le temps nécessaire pour casser le code aujourd'hui,

$T_0 = k10^n$.

Pour que le temps T, temps nécessaire demain pour casser le code, reste le même sachant que l'ordinateur travaille cent fois plus vite (cela revient à dire que k a été divisé par 100), il convient donc que le nombre d'opérations à exécuter soit multiplié par 100 :

$T_1 = T_0 = k/100 \times 10^n \times 10^2 = k' \, 10^{n+2}$

La lecture du petit calcul précédent indique que pour obtenir la même sécurité, il suffit d'augmenter le nombre de chiffres de p de 2 unités.

En revanche, les temps de codage et de décodage de l'ordre de n^3, si le nombre de chiffres de p est grand et est augmenté de deux unités, sont quasiment divisés par 100.]

On a ainsi un système de crytage efficace, le codage comme le décodage des textes sont rapides, il est résistant aux efforts des cryptanalystes. Un défaut cependant, que nous n'avons pas encore envisagé, demeure : si l'on veut que les messages circulent dans un réseau, et non plus seulement entre deux personnes, il est nécessaire que les différentes personnes du réseau possèdent les clés de codage et de décodage. La prolifération des possesseurs de clés augmente les risques de fuites et il suffit d'une indiscrétion pour mettre à bas tout le système de communication. Le dernier système que nous allons présenter permet de pallier ce défaut.

IV. LA CRYPTOGRAPHIE À CLÉS PUBLIQUES : LE SYSTÈME RSA

Considéré par certains comme une petite révolution dans le monde de la cryptographie, apparaît dans les années 1970 un système de codage à *clés publiques* (elles sont, par exemple, publiées dans un annuaire et ainsi elles sont rendues publiques !)

Le concept de "clés publiques" est dû, semble-t-il, à W. Diffie et M. Hellmann qui en ont publié l'idée en 1976. Cependant, ce sont trois autres auteurs R. Rivest, A. Shamir et L. Adleman qui ont les premiers, en 1978, concrètement proposé un tel système, connu aujourd'hui sous le sigle RSA (initiales des trois concepteurs).

Le principe de base en est le suivant : les clés de codage sont publiques, seules les clés de décodage sont gardées secrètes. Le système est tel que le temps nécessaire pour découvrir une clé de décodage à partir de la clé de codage est astronomique, impraticable de façon pratique.

Le système RSA, comme celui exposé dans le paragraphe précédent, est basé sur des exponentiations arithmétiques. Examinons-en le principe.

La clé de codage est un couple d'entiers (e, n) ; e est l'exposant et n le module. n est le produit de deux nombres premiers très grands (dans la pratique qui ont une centaine de chiffres en base décimale). On a donc n = pq, avec p et q premiers.

L'exposant e est un entier premier avec $\varphi(n)$ où $\varphi(n)$ (11) désigne le nombre d'entiers inférieurs à n premiers avec celui-ci. Nous verrons plus loin les raisons de ce choix.

Le codage se fait alors de la manière suivante :

On découpe le texte en blocs et on fait correspondre à chaque lettre son équivalent numérique, selon un procédé analogue à celui vu dans le paragraphe précédent.

(11) $\varphi(n)$ est la fonction indicatrice d'Euler.

Puis on procède à une exponentiation avec e modulo n. On code un bloc P et on obtient le bloc codé C correspondant par :

$$C \equiv P^e\ (n). \qquad 0 \leq C < n$$

Le principe du décodage : e étant premier avec $\varphi(n)$, on sait, en utilisant le théorème de Bézout, que e possède un inverse d modulo $\varphi(n)$.

Il existe donc d tel que
$ed \equiv 1\ (\varphi(n))$
et donc il existe k (k > 0) tel que
$ed = k\ \varphi(n) + 1.$

Montrons que d donne, par exponentiation, une clé de décodage :

Euler a montré le théorème suivant :

> Si n est un nombre positif, et a un entier premier avec n, alors :
> $a^{\varphi(n)} \equiv 1\ (n)$

[*Preuve :*

Elle est analogue, avec une variante bien sûr, au petit théorème de Fermat (ce dernier est un cas particulier du théorème d'Euler, car si n est premier, on a $\varphi(n) = n - 1$).

Considérons $r_1, r_2 \ldots r_{\varphi(n)}$ les entiers inférieurs à n et premiers avec n, puis les produits $ar_1, ar_2 \ldots a\,r_{\varphi(n)}$ et les restes $r'_1 \ldots r'_{\varphi(n)}$, de ceux-ci modulo (n). Comme a est premier avec n, les différents restes r'_i pour $i = 1 \ldots \varphi(n)$ sont distincts, et ainsi ils sont égaux dans leur ensemble, à l'ordre près, aux r_i pour $i = 1 \ldots \varphi(n)$.

On a donc :

$$ar_1 \times ar_2 \times \ldots \times ar_{\varphi(n)} \equiv r'_1 \times r'_2 \times \ldots \times r'_{\varphi(n)}$$
$$\equiv r_1 \times r_2 \ldots r_{\varphi(n)}\ (n)$$

d'où

$$r_1 \times r_2 \ldots \times r_{\varphi(n)}\ (a^{\varphi(n)} - 1) \equiv 0\ (n).$$

n étant premier avec chacun des r_i, est premier avec leur produit et ainsi (c'est une conséquence du théorème de Gauss), n divise $a^{\varphi(n)} - 1$, ce qui peut encore s'écrire :

$a^{\varphi(n)} \equiv 1\ (n)$ CQFD.]

Considérons le code C et élevons le à la puissance d modulo n :

$$C^d \equiv P^{ed}\ (n)$$
$$C^d \equiv P^{(k\varphi(n)+1)} \equiv (P^{\varphi(n)})^k . P\ (n)$$

P petit devant n, et aussi devant p et q, aussi est-on assuré que P soit premier avec n (car p et q premiers) et on peut appliquer le théorème d'Euler, on obtient :

$C^d \equiv P\ (n)$
On a bien décodé le message C.

On a vu, dans le paragraphe précédent, que le coût de codage, comme celui de décodage est raisonnable "informatiquement parlant".

Il existe aussi des techniques permettant de déterminer assez rapidement si un entier est premier ou non : ainsi pour choisir p et q, on peut choisir au hasard deux nombres de 100 chiffres en base décimale, puis leur appliquer des tests dits de "primalité" qui peuvent assurer, avec un risque d'erreur infime ($\leq 10^{-60}$ par exemple) car ce sont des tests basés sur des propriétés probabilistes des nombres, que p et q sont bien premiers ou non.

Sachant qu'il y a à peu près un nombre de 100 chiffres sur 115 qui est effectivement premier, on voit qu'il convient, certes, de réitérer les choix de p et q jusqu'à ce que l'on ait des nombres premiers, mais la réitération est raisonnablement coûteuse en temps.

L'impossible travail du cryptanalyste

La clé (e, n) est publique, donc connue de quiconque veut en prendre connaissance. Il "suffit" au cryptanalyste de déterminer d, travail facile et rapidement exécutable dès

lors que l'on connaît $\varphi(n)$: en effet, il convient de déterminer d un inverse de e modulo $\varphi(n)$, et on a vu qu'il était facile de faire cela avec l'algorithme d'Euclide étendu. *Mais là est la force du système RSA : partant de la clé publique (e, n), il est quasi impossible de calculer $\varphi(n)$;* le temps de calcul nécessaire est prohibitif et dissuasif. En effet, le problème revient à trouver p et q : si l'on connaît p et q, alors on sait que $\varphi(n) = \varphi(p).\varphi(q) = (p-1)(q-1)$. Trouver $\varphi(n)$ revient à factoriser n et à ce jour, on ne dispose pas d'algorithmes rapides pour ce faire. (Si p et q ont une centaine de chiffres chacun, n a environ 200 chiffres, et la factorisation d'un tel nombre nécessiterait, avec les meilleurs ordinateurs d'aujourd'hui, quelques milliards d'années de temps de calcul !).

Intérêt du système RSA

L'intérêt d'un tel système à clés publiques est qu'il permet la confidentialité des communications dans un réseau formé de multiples personnes. Chacune des personnes P_i du réseau a une clé publique e_i connue de tous et une clé secrète d_i. Supposons que P_i veuille envoyer un message à une autre personne P_j du réseau. La clé e_j étant publique, P_i peut coder son message à l'aide de celle-ci. Comme P_j est le seul à posséder la clé d_j, il est le seul à pouvoir décoder le message envoyé par P_i. Le système assure la confidentialité des transmissions. Le caractère public des clés permet ainsi à tout membre du réseau de communiquer, en sécurité, avec tous les autres.

Mais le système RSA possède une autre propriété intéressante. Il permet de résoudre un problème que nous n'avons pas abordé jusqu'ici, celui de l'authenticité du message. En effet, que ce soit lors de transmissions diplomatiques, militaires ou lors de transactions bancaires encore faut-il s'assurer de l'idendité de celui qui envoie le message (c'est un problème quotidien avec l'usage de cartes bancaires servant à retirer de l'argent ou servant à donner des ordres de versements).

Le système RSA permet à la fois d'assurer la confidentialité et l'authenticité des messages de la manière suivante :

Supposons que P_i veuille envoyer un message M à P_j.

P_i commence par coder le message M à l'aide de la clé publique e_j de P_j.

Le message M est transformé en un message codé M_1 avec :

$$M_1 \equiv M^{e_j}(n)$$

Puis P_i code le message M_1 à l'aide de sa clé secrète d_i ; M_1 devient M_2 avec :

$$M_2 \equiv M_1^{d_i} \equiv M^{d_i e_j}(n)$$

P_j reçoie donc le message M_2 et le décode en appliquant successivement la clé publique e_i de P_i, puis sa propre clé secrète d_j :

$$M_3 \equiv M^{e_i d_i e_j} \equiv M^{e_j}(n)$$

puis

$$M_4 \equiv M^{d_j e_j}(n).$$

Avec un tel système un personnage mal intentionné s'infiltrant dans le réseau ne peut envoyer des messages en se faisant passer pour une des personnes du réseau sans connaître la clé secrète de celle-ci.

V. CONCLUSION

L'avènement de l'informatique a donné un regain d'actualité à la cryptologie : longtemps entourée de mystères car il convenait que les modes et les clés de cryptages restent secrets, aujourd'hui l'utilisation conjuguée de l'arithmétique et de la puissance calculatoire des ordinateurs fait que l'on peut rendre public, comme on l'a vu, une grande partie du procédé de codage. Il n'en reste pas moins que le domaine du cryptage reste sensible ! Aux Etats-Unis, la National Security Agency a demandé, sans succès, au gouvernement d'interdire les publications scientifiques, ainsi que l'invitation de chercheurs étrangers aux colloques portant sur le sujet. Le Pentagone a obtenu l'interdiction de l'exportation de certains logiciels de cryptage. En France, il existe un texte de loi interdisant, sauf autorisation des autorités compétentes, de crypter quelque information que ce soit, mais aussi la diffusion des algorithmes de cryptage. En revanche, les mêmes méthodes sont utilisées, en particulier dans le domaine des transactions bancaires, pour authentifier les donneurs d'ordre : on parle de signature électronique.

Dans les pages qui précèdent, on a pu montrer le rôle de l'arithmétique et en particulier l'usage de résultats anciens : nous avons cité des théorèmes de Fermat (1602-1665), d'Euler (1707-1783), nous avons fréquemment utilisé la formule de Bézout (1730-1783) et le théorème de Gauss (1777-1855). Il n'en reste pas moins que la recherche en arithmétique demeure vive aujourd'hui et les problèmes posés par le cryptage via des modes informatiques ont contribué à redynamiser ce secteur de recherche. Par exemple, on ne sait pas aujourd'hui décomposer un nombre en ses facteurs premiers avec un algorithme rapide (en un temps polynomial), mais on ne sait pas non plus s'il n'existe pas un tel algoritme ! On peut comprendre que la sécurité d'un système comme le RSA dépende fortement des réponses qui seront apportées à ces questions.

BIBLIOGRAPHIE

DEMAZURE M. : (1997) *Cours d'Algèbre ; primalité, divisibilité, codes*, Ed. Cassini.

GROUPE DE TRAVAIL LYCÉES-UNIVERSITÉS (1998) : *Cours et activités en arithmétique pour les classes de terminales*, Ed. IREM de Marseille.

HELMANN M. (1979) : "Les mathématiques de la cryptographie à clé révélée", *in Pour la Science* n°24.

MERCIER D.J. (1996) : "Cryptographie classique et cryptograpie publique à clé révélée", *in Bulletin de l'APMEP* n°406.

ROBIN G. : (1994) : "La cryptographie moderne", *in Tangente* n°37.
— (1991) : *Algorithmique et cryptographie*, Ed. Ellipses.

ROSEN H.R. : *Elementary number theory and its applications*, Ed. Addisson-Wesley.

STERN J. (1998) : *La science du secret*, Ed. O.-Jacob.