DU CHIFFREMENT DE CESAR A LA MATHEMATIQUE DE LA CARTE BANCAIRE

Dany-Jack MERCIER
Iufm Antilles-Guyane
Centre Guadeloupe

Résumé : *Cet article est issu d'une conférence donnée à «La Science en Fête». Il propose un panorama des méthodes cryptographiques à travers le temps. Les chiffres de César et de Vigenère sont décrits et analysés, puis la mécanisation du chiffrement fait son apparition avec le cylindre de Jefferson et la célèbre machine ENIGMA utilisée pendant la seconde guerre mondiale. Au vingtième siècle, la nécessité de communiquer sur un réseau informatique entraîne l'invention de systèmes à clés publiques, dont le RSA utilisé actuellement dans plus de 85% des communications chiffrées. La part toujours actuelle des systèmes classiques est présentée avec la description du Digital Encryption Standard (DES). Enfin, la dernière Section explique l'utilisation du RSA et du DES dans le processus d'authentification des CB à puces.*

1. Introduction.

Cela fait des millénaires que l'homme s'est aperçu qu'il était nécessaire de protéger ses messages de ses ennemis. Le recours à l'écriture cachée était vital pour les rois et les généraux.

Les premières références écrites à l'utilisation de messages cachés sont données par Hérodote, archétype de l'historien, dès le 5ème siècle avant J-C. Dans son recueil intitulé «Histoires», Hérodote raconte comment Demaratus, sujet perse d'origine grecque et vivant à Suse, décida d'informer Sparte que Xerxès rassemblait la plus grande armée jamais connue dans le but d'envahir la Grèce. Il gratta la cire de deux tablettes de bois pliantes, grava le texte directement sur le bois, puis recouvrit le tout d'une nouvelle nappe de cire vierge. Son messager put rejoindre Sparte avec les tablettes sans être inquiété à chacun des barrages qu'il rencontra. A l'arrivée, Cléomène, épouse de Léonidas, eut l'idée d'enlever la cire et dévoila le message.

Dans un autre passage, Hérodote raconte aussi la ruse d'Histaïaeus qui décida le roi de Milet à se soulever contre les Perses. Celui-ci rasa les cheveux d'un messager, écrivit le message directement sur son crâne, puis attendit la repousse.

On peut dire, en plaisantant, que Histaïaeus employait déjà une technique de macro-points. Plus récemment, c'est aux alentours des années 1920 qu'Emanuel Goldberg inventa la

technique des micro-points qui devait être très utilisée par les espions soviétiques et allemands. Pendant la seconde guerre mondiale, les ambassades allemandes d'Amérique du Sud arrivaient à créer des micro-points suffisamment petits pour les placer à l'intérieur d'un point de ponctuation d'un document dactylographié. Les messages cachés se trouvaient dans la ponctuation, mais furent assez vite découverts par le contre-espionnage américain.

Cette technique est toujours d'actualité avec la dématérialisation de l'information. Une image en format bmp peut être exagérément agrandie dans un logiciel de dessin qui permet en outre de changer légèrement la nuance de certains points. Ces points peuvent former un message qui deviendra invisible lorsque l'image retrouvera sa taille normale. Rappelons aussi l'histoire récente de ce hacker qui avait envoyé un virus par mail, et qui fut dénoncé bien malgré lui par les filigranes présents dans chacun des documents produits avec le logiciel de traitement de texte qu'il utilisait...

En fait, les exemples que nous venons de décrire ne sont pas des exemples de cryptographie pure. Ce sont des procédés de stéganographie (en grec, stéganos = couvert, et graphein = écriture). La stéganographie propose de cacher la présence d'un message, tandis que la cryptographie (kruptos = caché) tente de cacher le sens du message en le rendant inintelligible à toute personne non autorisée. La stéganographie a un grand défaut : elle ne demeure sûre que tant qu'elle reste insoupçonnée. S'il y a présomption d'existence d'un message, il y a de forte chances pour que celui-ci finisse par tomber entre les mains ennemies après des recherches particulièrement fines. La cryptographie «au sens strict» constitue une seconde barrière qui se veut infranchissable. Historiquement, cette seconde barrière a été développée en même temps que la stéganographie. Et bien entendu rien n'empêche de protéger deux fois son message en le cryptant et en le dissimulant.

La Fig. 1 donne un autre exemple de stéganographie linguistique et a été cité dans [2]. Le sémagramme est extrait d'un polycopié de logique combinatoire traitant du problème des ponts de Königsberg. Il a été tapé par un mathématicien de RDA pour être expédié à un collègue d'Allemagne de l'Ouest. Si l'on fait bien attention, on s'aperçoit que certaines lettres ont été légèrement décalées pour former le message : «nieder mit dem sowjetimperialismus» («A bas l'impérialisme soviétique»).

L'un des tous premiers dispositifs militaires de chiffrement connus est la scytale que les spartiates utilisèrent au siècle de Péri-

In Königsberg i. Pr. gabelt sich der Pregel und umfließt eine Insel, die *Kneiphof* heißt. In den dreißiger Jahren des achtzehnten Jahrhunderts wurde das Problem gestellt, ob es wohl möglich wäre, in einem Spaziergang jede der sieben Königsberger Brücken genau einmal zu überschreiten.

Fig. 1

clès (5ème siècle Av. J.-C.). L'envoyeur et le receveur étaient munis d'un bâton de bois cylindrique. L'envoyeur enroulait une lanière de cuir autour du bâton, puis écrivait les lettres du message sur la lanière et sur toute la longueur de la scytale. Il enroulait ensuite la lanière autour de la taille d'un messager qui n'avait plus qu'à rejoindre le receveur (Fig. 2).

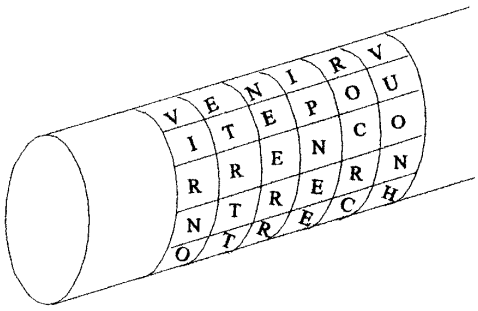


Fig. 2 : Scytale spartiate

Il y a deux grandes familles de méthodes en cryptographie stricte (c'est-à-dire hors stéganographie).

Un système de cryptographie *symétrique* (on dit encore *classique*, ou *à clé unique*) utilise une clé secrète unique. Les systèmes de César, de Vigenère, le chiffre ENIGMA ainsi que les récents standards DES, Triple DES et IDEA, font partie des systèmes symétriques et seront étudiés dans cet article.

Un système de cryptographie est dit *asymétrique* (ou encore *à clé publique*, ou *à clé révélée*) s'il utilise deux clés : une clé secrète de déchiffrement et une clé publique de chiffrement connue de tous. L'exemple actuel est le célèbre RSA largement utilisé dans plus de 80% des échanges électroniques sécurisés, et qui sera décrit à la Section 6.

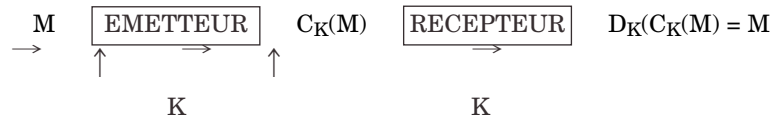
2. Codes de César et de Vigenère

2.1. Structure générale d'un cryptosystème classique

Les ingrédients d'un cryptosytème classique sont une fonction de chiffrement C_K, une fonction de déchiffrement D_K et une clé unique K. L'émetteur désire envoyer un message M. Pour cela il utilise une clé secrète K et produit le message codé $C_K(M)$. A l'aide de cette clé K, le receveur crée une fonction de déchiffrement D_K et l'utilise sur le message $C_K(M)$ pour reconstituer M. On doit donc avoir :

$D_K(C_K(M)) = M$, pour tout message M.

Cette égalité entraîne l'injectivité de C_K, autrement dit C_K doit associer des textes distincts à des messages distincts ! La clé K doit être connue et tenue secrète par l'émetteur et le récepteur. La communication est schématisée selon le dessin ci-dessous :



«Cryptosystème à clé simple»

2.2. Code de César (101 – 44 av. J.-C.)

Pendant la guerre des Gaules, Jules César envoyait des messages chiffrés à Cicéron qui était resté en poste au Sénat à Rome. L'historien Suétone, archiviste de l'Empereur Hadrien au I-II ème siècle, rapporte dans les «Vies des douzes Césars» que le célèbre Jules avait l'habitude de remplacer chaque lettre par celle située trois places plus loin dans l'alphabet.

Ecrivons le message M = *lumiere*, en utilisant l'alphabet occidental contemporain :

$\mathcal{A}$ = {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}

On aura :

Message clair : l u m i e r e
Message crypté : O X P L H U H

et l'on dit que ce chiffrement est une *substitution monoalphabétique*.

Le code de César fut très solide jusque dans les années 800, date où l'islam vivait son âge d'or. De façon étonnante, c'est l'étude des textes du Coran qui développa beaucoup de recherches sur les lettres, le but étant de déterminer si un texte donné avait bien été écrit par la main du prophète.

On découvrit alors l'importance de l'étude des fréquences d'apparition des lettres dans un message, et c'est le «Philosophe des Arabes», Al-Kindi, auteur de 290 livres, qui expliqua la méthode pour décrypter un message obtenu avec un alphabet de substitution. Son «Manuscrit sur le déchiffrement des messages cryptographiques» a été retrouvé en 1987 dans les archives d'Istanbul.

Si l'on sait maintenant que l'étude des fréquences d'apparition des lettres (ou des couples de lettres) dans les messages permet d'initier la cryptanalyse de César, on peut signaler que la découverte d'Al Kindi mit énormément de temps avant d'être connue en Occident. Le livre de Singh [13] explique en détails comment la cryptanalyse des messages entre Marie Stuart et ses partisans a permis l'inculpation puis l'exécution de la reine pour tentative de régicide.

2.3. Code de Vigenère (1523-1596)

Choisissons un mot pour clé, par exemple le mot LOIRE. La lettre L est la 12ème lettre de l'alphabet, et la première lettre du message sera décalée de 11 lettres vers la droite. La deuxième lettre sera décalée de 14 lettres puisque O est la 15ème lettre de l'alphabet. Et ainsi de suite... les décalages seront :

L ~ 11 ; O ~ 14 ; I ~ 8 ; R ~ 17 ; E ~ 4

puisque :

a	*b*	*c*	*d*	*e*	*f*	*g*	*h*	*i*	*j*	*k*	*l*	*m*
0	1	2	3	4	5	6	7	8	9	10	11	12
n	*o*	*p*	*q*	*r*	*s*	*t*	*u*	*v*	*w*	*x*	*y*	*z*
13	14	15	16	17	18	19	20	21	22	23	24	25

Voici un exemple de chiffrement :

Message : l u m i e r e
Clé : L O I R E L O
Message crypté : W I U Z I C S

Dans cet exemple, on constate que la lettre « e » est chiffrée différemment suivant sa position dans le texte (en I, puis en S), de sorte que l'attaque de Vigenère ne puisse pas

	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
1	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	A
2	C	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
3	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
4	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	D
5	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
6	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
7	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
8	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
9	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
10	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
11	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
12	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
13	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
14	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
15	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
16	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
17	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
18	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
19	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
20	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
21	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
22	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
23	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
24	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
25	Z	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
26	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

Fig.. 3 : Carré de Vigenère

utiliser l'analyse des fréquences comme dans le cas de César.

Le chiffrement de Vigenère est un chiffrement polyalphabétique par simple substitution. C'est un chiffrement par blocs.

De façon plus générale, si **A** désigne un alphabet et si l'on choisit m permutations σ_1, ... , σ_m de **A**, un bloc $M = (t_1, ... , t_m)$ de m lettres sera chiffré en $M' = (\sigma_1(t_1) , ... , \sigma_m(t_m))$.

Dans le cas de Vigenère et en identifiant **A** à l'anneau **Z**/n**Z**, on peut écrire :

$$M' = (t_1, ... , t_M) + D ,$$

où $D = (d_1, ... , d_M)$ est un vecteur de décalage. Dans l'exemple ci-dessus :

$$M' = M + (11,14,8,17,4).$$

La Fig. 3 de la page précédente représente le célèbre carré de Vigenère. Ce carré facilite le chiffrement : en conservant l'exemple donné plus haut, on chiffre la lettre l en utilisant la première lettre L de la clé en lisant la lettre située à l'intersection de la colonne l et de la ligne L. On obtient un W. Puis on recommence avec la seconde lettre u du message en clair. Celle-ci sera transformée en I situé à l'intersection de la colonne u et de la ligne O. Et ainsi de suite.

Le code de Vigenère fut très sûr pendant plus de 200 ans, mais il fut un peu boudé par les diplomates et les militaires parce que trop contraignant. Le chiffre de Vigenère était trop long à mettre en œuvre, le chiffre monoalphabétique était trop facile à casser, si bien que, pendant deux siècles, on préféra utiliser des chiffres de sécurité intermédiaire, tels le tableau de substitutions homophoniques ci-dessous ([13] p. 69). Une lettre (par exemple le e) est chiffrée par l'un quelconque des nombres à deux chiffres placés au dessous d'elle dans le tableau. Ces nombres ont été choisis pour que les

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
09	81	13	01	06	31	25	39	32	15	04	26	22	18	00	38	94	29	11	17	02	34	60	28	24	01
12		41	03	10				50			37	27	58	05	90		35	19	20	08	52				
33		62	45	14				56			51	68	59	07	95		40	21	30	61					
47				16				70			65		66	54			42	36	43	63					
48				23				73			84		71	72			77	76	49	85					
53				24				83					91				80	86	69	90					
67				44				88					99					96	75						
78				46				93										97							
92				54																					
				55																					
				57																					
				74																					
				79																					
				82																					
				87																					
				98																					

chiffres du message secret soient équirépartis. En particulier le e, lettre la plus fréquente dans un texte écrit en français, est représenté par 16 nombres différents.

2.4 Attaques sur Vigenère

Pour tester la sécurité d'un chiffre, il est habituel de faire les hypothèses de Kerckhoff ([9] p. 225) :

a) L'adversaire a accès à toute l'information chiffrée (il peut lire autant de messages chiffrés qu'il le désire),

b) L'adversaire connaît parfaitement les détails de la méthode de cryptographie employée, à l'exception de la clé.

Donnons un bref aperçu de trois attaques possibles sur Vigenère (pour plus de détails et pour une mise en œuvre sur des exemples, voir [13]).

1) Première attaque : méthode de Kasiski.

C'est Charles Babbage qui trouva le premier cette méthode en 1854, mais ne publia jamais sa découverte. Wilhelm Kasiski décrivit la même méthode en 1863.

La première phase consiste à avoir une idée sur les longueurs possibles de la clé. Pour cela, on cherche les répétitions dans le message chiffré. Ces répétitions doivent exister puisque des mots courants comme « les » ou « que » finiront par être traduits de la même façon. Si, par exemple, des répétitions sont repérées avec des intervalles de longueurs 42, 63 et 105, on écrit :

$42 = 2 \times 3 \times 7 \quad 63 = 3 \times 3 \times 7 \quad 105 = 3 \times 5 \times 7$

et l'on déduit que la longueur de la clé est un diviseur de 21, c'est-à-dire 1, 3, 7 ou 21.

Pour chacune des longueurs de clés possibles, on débute une analyse de fréquences comme dans le cas de César. Par exemple, pour une longueur de clé de 7, on calcule la fréquence d'apparition des lettres situées à la 1ère, 8ème place, 15ème place, etc.

2) Seconde attaque : cas d'un seul message et d'une très longue clé.

On utilise un chiffre jetable et une technique de « va-et-vient ». On fait comme si le texte en clair était une succession d'un même mot courant, par exemple « lesleslesles... ». On cherche « à rebours » (par exemple avec le carré de Vigenère) un mot-clé possible et de longueur égale à celle du texte chiffré. Ensuite on repère les significations éventuelles des mots-clés trouvés. Si le tronçon « kgw » apparaît dans la clé, il a peu de chance de faire partie du mot-clé car aucun mot français n'est construit avec ce tronçon. Par contre, si le tronçon « mir » est reconnu, on peut raisonnablement supposer qu'il fait partie d'un mot-clé plus long comme « miroir », « mirabelle » ou « émirat »... On envisage ces hypothèses une à une, et cela permet de transcrire une partie plus importante du message chiffré en clair. Si le message obtenu a un sens, on le complète et on met la main sur une partie plus importante de la clé. Il ne reste plus qu'à recommencer un peu partout dans le texte en espérant que le processus ne s'interrompe pas.

3) Troisième attaque : cas d'une même clé utilisée deux fois.

a) On utilise un chiffre jetable et la technique du « va-et-vient » expliquée en 2). On obtient ainsi des morceaux probables de clés.

b) On utilise immédiatement ces clés jetables sur le second message, et l'on repère si des parties du second message ont un sens. Le cas échéant, on complète les mots reconnus et on découvre une partie plus grande de la clé.

c) On retourne sur le premier message avec la nouvelle clé plus complète. Et ainsi de suite.

2.5. La Grande Guerre

Les impératifs stratégiques imposent qu'un chiffre militaire soit facile d'emploi, permette d'envoyer de nombreux messages chaque jour, et assure la confidentialité pendant un laps de temps relativement long. Pendant la Grande Guerre, les chiffres utilisés était formés de transpositions et de substitutions ([4], [7]).

C'est en 1918 que le major Joseph Mauborgne (USA) vante les mérites de Vigenère utilisé avec une *clé aléatoire à usage unique*. Il distribue des livres d'environ 100 pages, chacune d'elle contenant une clé aléatoire de 100 lettres. L'envoyeur et le receveur possèdent un exemplaire du même livre, et utilisent une page par message à transmettre.

La clé aléatoire à usage unique offre une sécurité absolue : il y a impossibilité de retrouver la clé pour deux raisons :

a) Il existe $26^{100} \approx 10^{141}$ possibilités pour une clé de longueur 100,

b) Si M désigne le message chiffré, alors quelle que soit la phrase L contenant 100 lettres, il y aura une clé K de longueur 100 qui permette de chiffrer L en M. Autrement dit le message crypté prendra toutes les significations possibles et imaginables suivant la clé que l'on utilise. On comprend alors qu'il devienne absolument impossible de retrouver « la » phrase envoyée.

Les trois attaques de Vigenère envisagées précédemment ne suffisent plus. La méthode de Kasiski ne peut plus être employée puisque la clé est aussi longue que le message. La seconde méthode nécessitait une clé qui ait un sens en français, ce qui n'est plus le cas avec une clé aléatoire. Enfin la clé utilisée une seule fois empêche le va-et-vient entre deux messages.

Tout en étant inconditionnellement sûre, cette méthode fut à peine utilisée à cause de ses inconvénients :

a) Le système est bien lourd pour être mis en œuvre sur le champ de bataille où il y a énormément de messages à envoyer,

b) Il y a trop de difficultés à créer des clés aléatoires «à la main»,

c) Il faut distribuer et synchroniser les clés pour pouvoir communiquer,

d) La protection des clés doit être assurée bien que des livres secrets de 100 pages circulent sur le front.

3. Enigma

L'utilisation de rouages mécaniques, de l'électricité, ou plus récemment de composants électroniques, facilite les substitutions polyalphabétiques et permet l'emploi des clés très longues et aléatoires. Les premiers disques à chiffrer furent créés au 15ème siècle par l'architecte italien Léon Alberti, puis furent utilisés pendant la guerre de sécession américaine.

A titre d'exemple de « mécanisation du chiffrement », citons le cylindre de Jefferson

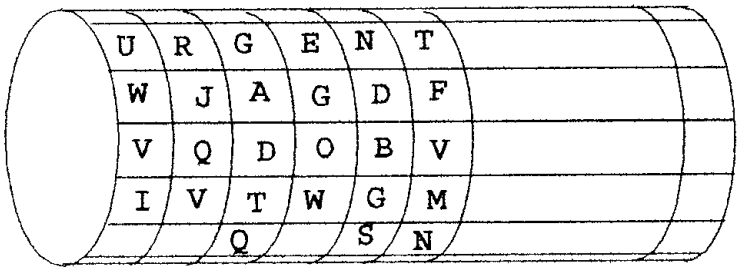


Fig. 4 : Cylindre de Jefferson

(fin 18ème siècle) qui comportait 36 disques de 26 lettres chacun. Une phrase de 36 lettres était chiffrée de 25 façons différentes (et les chiffres étaient associés aux lettres a, e, i, o, u, y, f, l, r, s). Sur la Fig. 4, on a tourné les disques de façon à obtenir le message « urgent ». Ce message est crypté en « WJAGDF » ou « VQDOBV », etc.

La célèbre machine ENIGMA a été inventée par Arthur Scherbius en 1918, et constitue un exemple de machine à rotors (comme la machine du suédois Hagelin). Elle fut l'arme cryptographique de l'armée allemande durant la seconde guerre mondiale [13]. ENIGMA avait l'apparence d'une machine à écrire, et le fait d'appuyer sur une touche du clavier faisait allumer une lampe qui éclairait la lettre à employer dans le message chiffré.

Une batterie électrique alimentait un rotor sur lequel étaient placées des connexions. Le rotor pouvait prendre 26 positions différentes, bien que la Fig. 5 (voir page suivante) ne symbolise que quatre positions possibles.

En fait ENIGMA comportait trois rotors de 26 lettres chacun, ce qui fournissaient déjà $26^3 = 17\,576$ substitutions différentes. Les mouvements des rotors étaient ceux d'un compteur kilométrique : lorsque le premier rotor terminait un tour complet, un engrenage faisait tourner le second rotor d'un cran, et ainsi de suite (Fig. 6).

En position de départ, le rotor R_i crée une substitution σ_i. Si R_i tourne de j crans, il crée une substitution $\sigma_i^{(j)}$ définie par :

$$\sigma_i^{(j)}(x) = \sigma_i(x - j) + j$$

où chacune des 26 lettres est associée à un entier modulo 26. Dans la position de départ, la substitution utilisée est

$$f = \sigma_3 \circ \sigma_2 \circ \sigma_1 \ .$$

Scherbius eut l'idée géniale de placer un réflecteur après les rotors. Celui-ci agissait comme une permutation g d'ordre 2 sur l'alphabet, et renvoyait le signal électrique à travers les rotors en suivant un autre circuit du même type qu'à l'aller (Fig. 7).

Avec ce réflecteur, la même machine pouvait servir tant au chiffrement qu'au dé-

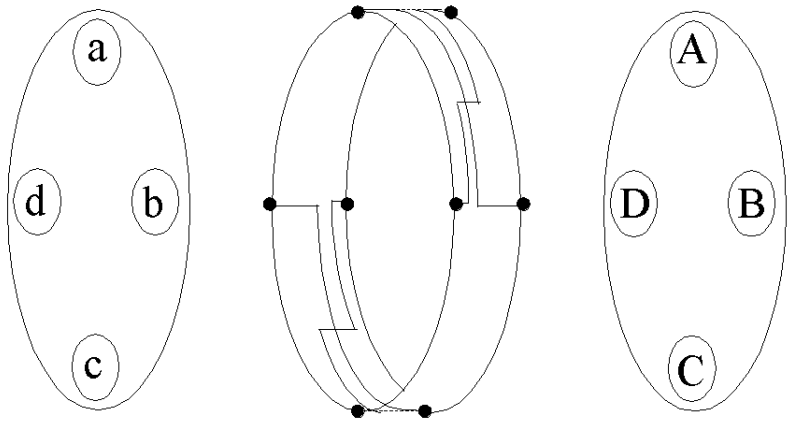

a
d
b
c
A
D
B
C
Clavier
Rotor
Lampes

Fig. 5

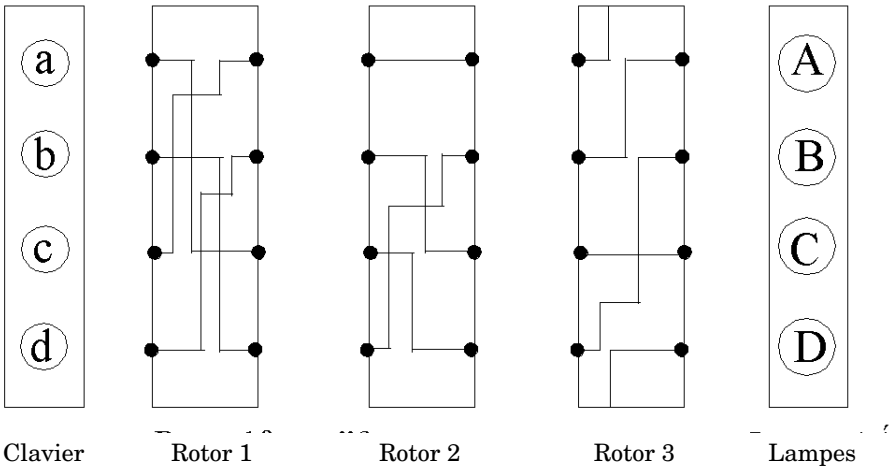

a
b
c
d
A
B
C
D
Clavier
Rotor 1
Rotor 2
Rotor 3
Lampes

Fig. 6

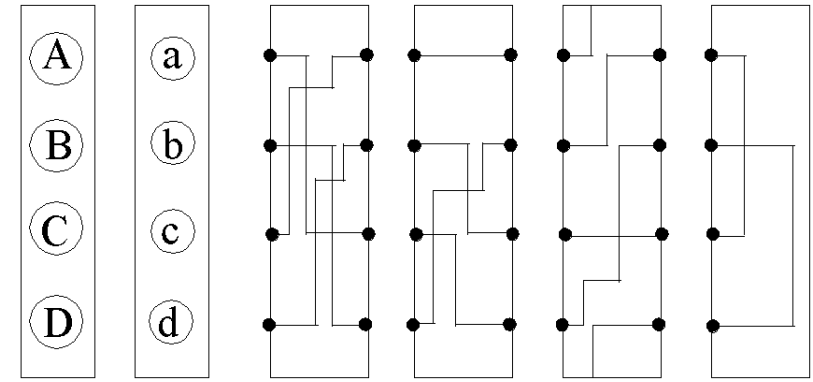

A
B
C
D
a
b
c
d
Lampes
Clavier
Rotor 1
Rotor 2
Rotor 3
Réflecteur

Fig. 7

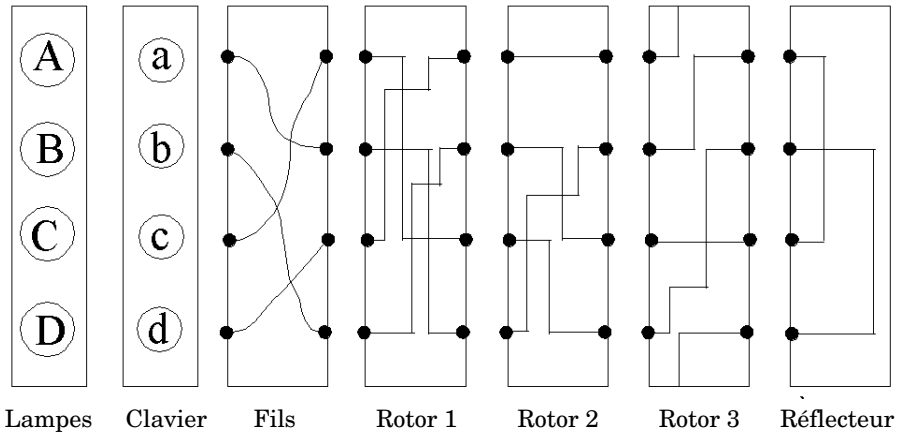

A
B
C
D
a
b
c
d
Lampes
Clavier
Fils
Rotor 1
Rotor 2
Rotor 3
Réflecteur

Fig. 8

chiffrement, et cela sans qu'il soit fait de manipulations sur les rotors. L'écriture

$$f = \sigma_1^{-1} \circ \sigma_2^{-1} \circ \sigma_3^{-1} \circ g \circ \sigma_3 \circ \sigma_2 \circ \sigma_1$$

de la permutation utilisée permet en effet d'obtenir $f(f(x)) = x$.

Pour finir, l'inventeur ajouta un tableau de connexions à fiches à la sortie du clavier. La machine était livrée avec 6 câbles permettant de relier $6 \times 2 = 12$ lettres entre elles. La Fig. 8 ne symbolise que quatre fils (cf. page précédente).

En guise de conclusion, on peut chercher le nombre total de clés possibles :

— L'initialisation des rotors (encore appelés brouilleurs) donne $I = 26 \times 26 \times 26 = 17576$ positions.

— On peut disposer les rotors de $D = 6$ manières.

— Enfin le tableau de connexions offre $T = C_{26}^{6} \times A_{26}^{6} \approx 8{,}654 \times 10^{10}$ possibilités.

Le nombre total de clés sera :

$$I \times D \times T \approx 10^{16}$$

soit 10 000 000 000 000 000 clés.

C'est clairement le tableau de connexions qui offre le plus grand nombre de clés. Dans ce cas pourquoi installer des rotors ? Eh bien tout simplement pour allonger la longueur de la clé permettant la substitution polyalphabétique et rendre le chiffre inattaquable par la méthode de Kasiski.

Essayons de bien comprendre l'utilité des rotors, puisqu'il s'agit là de la clé de la sécurité d'ENIGMA. Imaginons d'abord une machine sans rotor. Une fois la clé choisie — c'est-à-dire une fois les fils du tableau de connexions branchés et les rotors placés — la machine chiffrerait une lettre donnée toujours de la même façon, et le chiffrement qui en résulterait serait monoalphabétique, du type César, mais avec une permutation σ à la place du décalage à droite.

Imaginons maintenant que les rotors fonctionnent. Chaque fois qu'une lettre est chiffrée, le Rotor 1 tourne d'un cran. Lorsque le Rotor 1 achève un tour complet — soit 26 crans — le Rotor 2 tourne d'un cran, puis le Rotor 1 tourne à nouveau d'un cran à chaque chiffrement d'une lettre. Et le processus continue ainsi, l'ensemble des trois Rotors s'enclenchant comme ceux du compteur kilométrique d'une automobile. Utiliser des Rotors revient donc à utiliser des permutations différentes suivant la place de la lettre dans le message. Avec ses Rotors, la machine ENIGMA nous donne un chiffrement de Vigenère défini à l'aide d'une clé longue : la longueur de la clé, égale au nombre de positions des rotors prises pendant le processus de chiffrement, sera de $26^3 = 17576$, et permet déjà l'envoi de messages suffisamment longs.

Que nous offre ENIGMA au niveau de la sécurité ? ENIGMA ne propose, somme toute, qu'un chiffrement de Vigenère associé à une clé longue et aléatoire, et, à ce titre, on peut tester sa sécurité en utilisant les attaques décrites à la Section 2.4.

Premièrement, la longueur de la clé permet d'éviter l'attaque par la méthode de Kasiski si les messages envoyés sont moins longs que la clé, autrement dit comportent moins de 17576 caractères. Deuxièmement, le caractère aléatoire de la clé — celle-ci n'a aucune signification intelligible à priori — empêche l'utilisation de la seconde attaque menée à la

Section 2.4. Finalement seule la troisième attaque reste possible, et une façon de la contrer est de ne jamais utiliser la même clé pour chiffrer deux messages différents dans un laps de temps donné. Cet objectif est atteint lorsque l'émetteur E et le récepteur R se sont préalablement entendus sur une méthode de choix de clés. Ainsi E et R se mettent d'accord pour chiffrer le premier message du jour J en utilisant une certaine clé. Le second message envoyé le jour J sera chiffré à l'aide d'une clé déduite de la première clé utilisée suivant une méthode connue — seulement — de E et de R. Et ainsi de suite jusqu'au dernier message de la journée.

La sécurité d'ENIGMA est donc basée à la fois sur le secret concernant la première clé qui doit être utilisée le jour J — c'est-à-dire le tableau de connexions à fiches et les positions initiales des trois rotors — et sur le secret concernant le processus de changement de clés utilisées le jour J pour chiffrer les messages suivants. Un tel secret est difficile à garder.

Si le code d'ENIGMA a pu être découvert par les alliés, c'est dans une grande mesure parce que les services de renseignement franco-britanniques obtinrent des indications précises sur le fonctionnement de la machine bien avant la déclaration de guerre en 1939, et c'est aussi parce qu'une machine ENIGMA a pu être saisie sur un sous-marin allemand en 1941. A ce sujet, citons l'encart paru dans [9] à l'occasion de la sortie du film U-571 :

« Enigma, la machine de cryptage allemand durant la Seconde Guerre mondiale, est à l'affiche, à travers le film de Jonathan Mostow U-571 (...). Les précisions d'un témoin de l'époque, envoyé en France pour le compte du M16, l'agence de renseignement britannique. C'est en effet la Royal Navy qui a saisi l'Enigma à bord d'un sous-marin allemand. Cet événement date du 8 mai 1941. Sept mois avant que l'Allemagne et l'Italie ne déclarent la guerre aux Etats-Unis. Les Américains n'eurent aucune part dans le décryptage des messages même s'ils en furent, aussi, les bénéficiaires. La tentative de construire une machine Enigma puis de décrypter ses messages avait été entreprise bien avant la guerre. Un ingénieur qui travaillait dans l'usine qui fabriquait les Enigma s'était présenté à l'ambassade de France à Varsovie. Il se proposait d'expliquer son fonctionnement.

Les Anglais et les Français envoyèrent des spécialistes en Pologne. Parmi eux se trouvait le fameux mathématicien anglais Alan Turing. Ils proposèrent à l'ingénieur de venir s'installer en Grande-Bretagne. Il préféra la France.

Dès lors un officier du Deuxième Bureau français, le commandant Bertrand, futur général, fut l'un des deux officiers qui suivirent les travaux de l'ingénieur. Le second fut le commandant Wilfred Dunderdale, représentant des Services secrets britanniques (M16) à Paris. Ils furent de ceux qui apportèrent à Londres la première machine Enigma fabriquée en France. »

4. Limites d'un cryptosystème conventionnel

Différents problèmes surviennent lorsqu'on utilise un cryptosystème classique.

4.1. Problème de la multiplication des clés

Si n personnes désirent communiquer entre elles, elles auront besoin de n^2 clés, et ces nom-

breuses clés devront être gardées secrètes. Cela est très difficile à réaliser lorsque l'on a une communication «B to B» (Business to Business, autrement dit entre sociétés) puisqu'alors un certain nombre de personnes ont l'accès aux clés. De plus, ces nombreuses clés doivent être changées à intervalles réguliers.

4.2. Problème de la communication des clés

Comment deux interlocuteurs utilisant le même réseau de communication peuvent-ils s'entendre pour choisir une clé commune et secrète, alors même que les communications peuvent être piratées sur le réseau ? Le problème est crucial, puisque la communication d'une clé secrète est un préalable indispensable à toute communication sécurisée utilisant un cryptosystème classique.

On peut aussi poser la question de savoir comment obtenir la clé d'un correspondant dont le nom figure pourtant sur un annuaire.

4.3. Problème de l'authentification

Il s'agit de garantir l'origine d'une information. Prenons un exemple, et supposons que la personne A envoie le message M = « prélevez mille francs sur mon compte » à la personne B en utilisant le code de César. Le message chiffré est M' = $C_K(M)$ = « SUHOHYHC PLOOH IUDQFV VXU PRQ FRPSWH ». Quand B reçoit le message chiffré, il applique la clé de décalage de trois lettres vers la gauche et obtient M. Le message M a bien été chiffré avec la clé convenue et B peut considérer que ce message provient bien de A (qui a priori est le seul à posséder cette clé de chiffrement et à savoir quand l'utiliser).

Si B débite le compte de A, rien ne pourra empêcher A d'affirmer qu'il y a eu tromperie et que l'autorisation M n'a jamais été envoyée par ses soins. A peut même expliquer que B a agit sans aucun ordre dans le but de l'escroquer, et que les messages M et M' produits par B pour sa défense ont été créés de toutes pièces par B, ce qui, au demeurant, est tout à fait plausible. Et malheureusement aucune tierce personne (juge) non impliquée dans l'affaire ne pourra décider qui dit la vérité.

Ainsi l'association ($C_K(M)$, M) du message chiffré et de sa traduction en clair ne constitue en aucune manière la preuve de l'envoi du message par la personne qui l'a signé. L'usage simple d'un système conventionnel interdit ainsi l'envoi d'un chèque électronique à un créancier, et empêche toute transaction sur un réseau.

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, mais aussi se donner les moyens de démontrer à une tierce personne que le message reçu provient bien de la personne qui l'a signé.

Le processus d'authentification fait en général appel à une signature numérique obtenue à l'aide d'un couple de clés dont celle permettant de créer les signatures est conservée secrète. Bien entendu, le problème de la protection de l'intégrité du message est lié à celui de la signature. La signature devra dépendre à la fois de l'expéditeur et du message pour éviter que le message ne soit modifié, en chemin, par une tierce personne. Des exemples de signatures seront donnés à la Section suivante et à l'occasion de RSA en 6.1.

5. La cryptographie à clé révélée (PKC)

Pour la première fois et en 1976, Diffie et Hellman définissent les principes d'une communication sécurisée utilisant une clé publique [6] (PKC ou Public Key Cryptosystem). Deux ans plus tard, Rivest, Shamir et Adleman proposent un exemple de système de cryptographie basé sur ces considérations : ce sera le système RSA décrit dans la Section suivante, et largement utilisé actuellement [1]. Il représente plus de 85 % des échanges sécurisés et on le retrouve dans le célèbre PGP (Pretty Good Privacy) de Phil Zimmerman [12].

Dans un système à clé révélée, le récepteur est le seul à connaître les fonctions C et D de chiffrement et de déchiffrement. L'accès à la fonction C est libre. Notons C la clé publique (ou clé révélée) de chiffrement, et D la clé secrète de déchiffrement.

L'émetteur E lit la clé publique C du récepteur R sur un bottin. Il l'utilise pour créer le message chiffré C(M) à partir de son message en clair M, puis envoie C(M) au récepteur. Celui-ci déchiffre en appliquant sa clé secrète D. Il calcule ainsi D(C(M)) et doit retrouver M.

Le schéma de communication est donné par le dessin ci-dessous.

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

P1 : Pour tout message M, on a $D(C(M)) = M$,

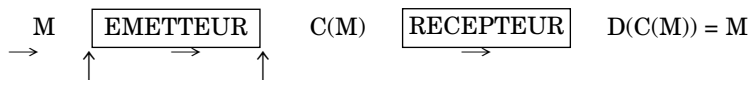
P2 : La connaissance de C et de $M' = C(M)$ interdit néanmoins le calcul de D(M') en un temps raisonnable,

et éventuellement :

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

Definition 1 : Une *fonction trappe*, ou *fonction à sens unique*, est une fonction $C : M \longrightarrow M'$ pour laquelle il existe une fonction $D : M' \longrightarrow M$ vérifiant les propriétés P1 et P2. Une *permutation trappe* ou *permutation à sens unique* vérifie en outre la propriété P3.

Avec une fonction trappe, le problème de la multiplication des clés est résolu puisque le nombre de clés nécessaires à la communication entre n abonnés passe de n^2 à n. Le problème de la communication des clés est aussi résolu puisque toutes les clés nécessaires au chiffrement des messages sont connues du public. De plus toute personne «hors liste» peut envoyer un message chiffré vers l'un des abonnés, pourvu que cet abonné ait inscrit sa clé



C clé publique lue sur le bottin D clé secrète du récepteur

«Cryptosystème à clé simple»

publique dans un bottin. Si C est une permutation trappe, alors le problème de la signature est aussi résolu. Si E et R désignent un émetteur et un récepteur appartenant à la même liste d'abonnés, notons

C_E = clé publique de chiffrement de E,

D_E = clé secrète de déchiffrement de E,

C_R = clé publique de chiffrement de R,

D_R = clé secrète de déchiffrement de R.

L'émetteur E applique sa clé secrète D_E au message M qu'il veut transmettre. Il applique ensuite la clé publique du récepteur R pour obtenir $M' = C_R(D_E(M))$. Dès que le récepteur reçoit le message M', il calcule $D_R(M') = M''$ et peut lire une entête qui lui indique que ce message provient de l'abonné E, et qu'il est signé. Le récepteur lit alors la clé publique C_E de E sur le bottin, puis calcule $C_E(M'')$ pour obtenir M et la totalité de l'envoi. Comme D_E est connue seulement de E, il est certain que le message M provient de E, et peut le prouver à une tierce personne.

Le transfert de données s'établit suivant le schéma suivant :

$$M \longrightarrow M' = C_R(D_E(M)) \longrightarrow D_R(M') = M'' \longrightarrow C_E(M'') = M$$

qui utilise la propriété :

$$\forall\, M \in \mathcal{M} \quad C_E \circ D_R \circ C_R \circ D_E\,(M) = M\ .$$

Ce qui est résumé par le schéma du bas de la page.

Dans la pratique, un PKC est beaucoup plus lent qu'un cryptosystème classique, mais permet l'échange d'une clé commune secrète permettant à un système conventionnel de fonctionner, comme dans PGP.

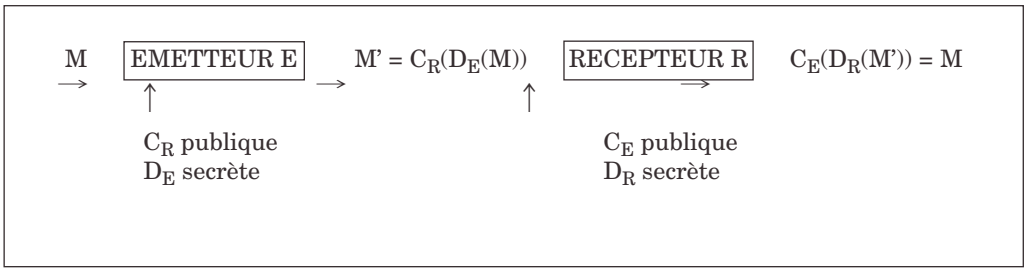
6. Système RSA

6.1. Principe

La mathématique du système RSA est contenue dans le résultat suivant :

Théorème : Si $n = pq$ est le produit de deux nombres premiers, si $m = (p-1)(q-1)$, et si c et d sont deux entiers naturels liés par la relation $cd = km + 1$, où k est un entier, alors la congruence $x^{cd} \equiv x\ [\,n\,]$ est vraie pour tout entier relatif x .

Preuve : Si $x \in \mathbf{Z}$ n'est pas divisible par p, le petit Théorème de Fermat permet d'écrire $x^{p-1} \equiv 1\ [\,p\,] \Rightarrow x^{km} \equiv 1\ [\,p\,] \Rightarrow x^{cd} \equiv x\ [\,p\,]$. Cette dernière congruence est triviale si x



est divisible par p, donc $x^{cd} \equiv x\ [p]$ pour tout $x \in \mathbf{Z}$. De même, on démontrerait que $x^{cd} \equiv x\ [q]$ pour tout $x \in \mathbf{Z}$. Comme p et q sont premiers et distincts, on déduit $x^{cd} \equiv x\ [n]$.

Pour chiffrer un message, on choisit deux nombres premiers p et q très grands et l'on calcule $n = pq$. On pose $m = (p-1)(q-1)$, et l'on cherche deux entiers naturels c et d tels que $cd \equiv 1\ [m]$. D'après le théorème, il existe donc $k \in \mathbf{Z}$ tel que $cd = km + 1$.

Les messages x seront des entiers appartenant à $\{0,1, \ldots, n-1\}$. Les fonctions de chiffrement et de déchiffrement seront :

$$C(x) \equiv x^c\ [n]\ ,$$

$$D(y) \equiv y^d\ [n]\ .$$

On a bien :

$$D(C(x)) \equiv D(x^c) \equiv x^{cd} \equiv x\ [n]$$

et

$$C(D(y)) \equiv C(y^d) \equiv x^{dc} \equiv x\ [n]$$

pour tout $x \in \{0,1, \ldots, n-1\}$. Ce procédé appelle quelques remarques :

1) Pour chiffrer, on a besoin de C, c'est-à-dire 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 écrire abusivement $C = (c, n)$ et $D = (d, n)$. Dans ce cas C représente la clé publique de chiffrement et D la clé secrète de déchiffrement.

Il reste à vérifier que C est une permutation à sens unique, ce qui revient à tester la sécurité du système RSA.

Si l'entier $n = pq$ est connu de tout le monde, les nombres premiers p et q doivent demeurer cachés car leur connaissance entraîne celle de $m = (p-1)(q-1)$, puis celle de d en résolvant l'équation de Bezout :

$$cd - km = 1,$$

ce qui est possible puisque c est dans l'annuaire.

Pour que le système puisse fonctionner de manière sûre, il faut choisir deux nombres premiers p et q très grands, possédant au moins 100 chiffres, et rendre ainsi leur calcul « impossible en un temps raisonnable » à partir de la seule connaissance de leur produit n, même en ayant recours à des ordinateurs puissants.

La sécurité du système RSA repose sur la facilité d'obtenir des nombres premiers très grands — il existe des tests qui certifient la primalité d'un entier et qui sont rapides — et sur la difficulté d'obtenir la décomposition d'un grand nombre en produit de facteurs premiers. Le tableau suivant donne une estimation du temps nécessaire pour obtenir la décomposition de n (en 1978, [1]).

Nb. 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 [13] :

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

Quelques précisions complémentaires sur les algorithmes mis en œuvre dans le système RSA sont données en Annexe à la Section 10.

6.2. Exemple numérique

Prenons $p = 163$, $q = 359$ et

$$n = 163 \times 359 = 58517.$$

On a $m = 162 \times 358 = 57996 = 2^2 \times 3^4 \times 179$ et l'on peut choisir $c = 5 \times 17 \times 59 = 5015$ qui est premier avec m.

On résout l'équation de Bezout :

$$5015d = 1 + 57996k$$

et l'on obtient :

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

où $t \in \mathbf{Z}$.

Choisissons $d = 57996 - 19093 = 38903$. Si $x \in \{ 0, 1, \ldots , 58516 \}$, la clé de chiffrement sera $C(x) = x^{5015}$ modulo 58517 et la clé de déchiffrement $D(x) = x^{38903}$ modulo 58517.

Un tableau de correspondance permet 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
00	01	02	03	04	05	06	07	08
I	J	K	L	M	N	O	P	Q
09	10	11	12	13	14	15	16	17
R	S	T	U	V	W	X	Y	Z
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, 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 modulo n en fonction de x :

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

Le message chiffré sera donc :

36974-30760-55559-47231-55755.

On le déchiffrera en calculant y^{38903} modulo 58517 pour chaque bloc y de cinq chiffres. Par exemple $36974^{38903} \bmod 58517 = 09120$ nous redonne la combinaison de lettres IL.

6.3. Longueurs des clés RSA

— RSA 320 bits est présente dans l'ancienne version de la CB, et devrait disparaître complètement d'ici à 2002. Il correspond à un nombre $n = pq$ de l'ordre de $2^{320} \approx 10^{96}$ qui s'écrit avec environ 97 chiffres décimaux.

— RSA 512 bits correspond à un nombre $n = pq$ de l'ordre de $2^{512} \approx 10^{154}$ s'écrivant avec 155 chiffres décimaux. Cette clé était jusqu'ici très utilisée, mais devient insuffisante puisque l'on a réussi à factoriser un nombre n de 512 bits (record de factorisation du 22 août 1999 cité par [3]).

— RSA 792 bits devait être utilisée dans la nouvelle version de la CB courant 2000, et offrir vraisemblablement une bonne protection pour les 4 à 5 ans à venir. Le nombre $n = pq$, de l'ordre de $2^{792} \approx 10^{238}$, s'écrit avec 239 chiffres décimaux.

L'idéal aurait été de passer directement à RSA 1024 bits, mais cela aurait augmenté le temps d'attente pour l'authentification de 10 secondes, tout en rendant nécessaire le renouvellement du parc de machine électroniques de vérification dans la distribution.

— RSA 1024 bits offre actuellement toute la sécurité nécessaire [3]. Il correspond à un nombre $n = pq$ de l'ordre de $2^{1024} \approx 10^{308}$ s'écrivant avec 309 chiffres décimaux.

7. Le Digital Encryption Standard (DES)

Le Data Encryption Standard est le plus connu des chiffres à blocs et à clé symétrique. Il a été retenu en 1977 par le U.S. National Bureau of Standards (American Standard FIPS 46-2).

7.1. Description

Un message de 64 bits est chiffré en un message de même longueur, en utilisant des opérations basiques et le schéma de Feistel. On parle de *schéma de Feistel* lorsque le message initial M est partagé en deux parties de même longueur, la partie gauche L_0 et la partie droite R_0 , et lorsqu'on adopte une structure en « échelle ».

La Fig. 9 (voir page suivante) montre les 16 opérations (ou rondes) successives du DES. L'échelle utilisée est croisée sauf à la dernière ronde.

La clé K de longueur 64 bits contient 8 bits de contrôles destinés à assurer l'intégrité de l'information. Sa partie utile mesure donc 56 bits. La clé K permet d'obtenir 16 clés partielles K_1 , ... , K_{16} de 48 bits chacunes qui seront utilisées dans chacune des rondes.

On commence par une permutation initiale IP et l'on termine par la permutation inverse IP^{-1}. Le tableau ci-dessous explicite IP. Il signifie que :

$$IP(m_1 m_2 m_3 \ \ldots m_{64}) = m_{58} m_{50} m_{42} \ldots m_7.$$

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

La $i$ème ronde transforme (L_{i-1}, R_{i-1}) en (L_i , R_i) en posant $L_i = R_{i-1}$ et $R_i = L_{i-1} \oplus f(R_{i-1},K_i)$, où $f(R_{i-1},K_i) = P(S(E(R_{i-1}) \oplus K_i))$.

Les 16 opérations du DES (Digital Encryption Standard)

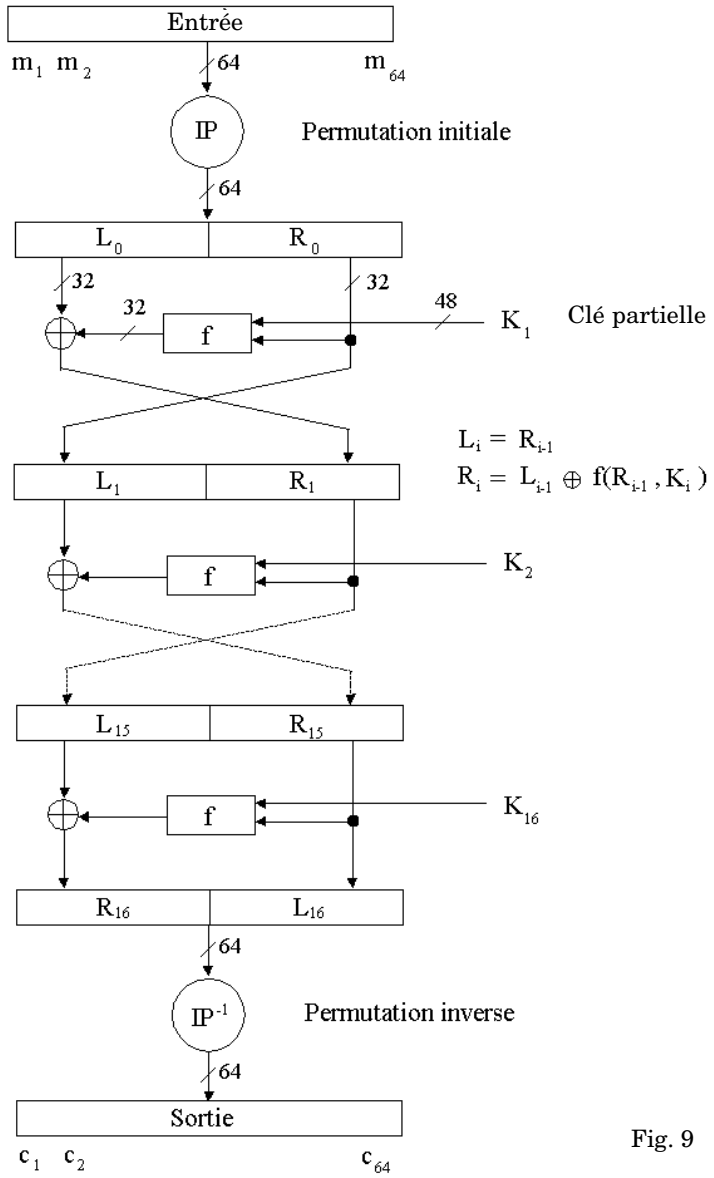


Fig. 9

Dans cette écriture, $\oplus$ désigne la somme bit à bit (c'est la somme dans $\mathbf{Z}/2\mathbf{Z}$, aussi connue sous le nom de XOR ou «ou exclusif»). L'expansion E de 32 à 48 bits et la permutation P de 32 bits sont données par :

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

La réduction S transforme un mot de 48 en un mot de 32 bits. Le mot de 48 bits est découpé en 8 blocs de 6 bits chacun, et le codage s'effectue à l'aide de 8 tables appelée S-boxes (données dans [10] p. 260). La S_1-box est représentée dans le cadre ci-dessous

Notons :

$S(B_1, B_2, \dots, B_8) = (S_1(B_1), S_2(B_2), \dots, S_8(B_8))$.

Si $B_i = b_1b_2b_3b_4b_5b_6$ alors $S_i(B_i)$ est calculé ainsi :

a) $r = 2b_1 + b_6$,

b) $b_2b_3b_4b_5$ est la représentation 2-adique de c,

c) Le nombre N lu dans la S_i-box à la r-ième ligne et la c-ième colonne est écrit en base 2 pour donner les 4 bits de sortie formant $S_i(B_i)$.

Exemple : $B_1 = 111101$; $r = 3$;

$c = \overline{1110} = 2^3 + 2^2 + 2 = 14$,

$N = 6 = 2^2 + 2 = \overline{0110}$, donc $S_1(B_1) = 0110$.

7.2. Calcul des clés partielles

Notons $K = k_1k_2 \dots k_{64}$ la clé du DES. Les 8 bits $k_8, k_{16}, k_{24}, k_{32}, k_{40}, k_{48}, k_{56}, k_{64}$ sont des bits de parité.

1) Les 56 bits d'information de K servent de variables à la permutation PC1 définie par le tableau :

C_0

57	49	41	33	25	17	9
1	58	50	42	34	26	18
10	2	59	51	43	35	27
19	11	3	60	52	44	36

D_0

63	55	47	39	31	23	15
7	62	54	46	38	30	22
14	6	61	53	45	37	29
21	13	5	28	20	12	4

Autrement dit, on calcule $PC1(K) = (C_0, D_0)$, où

$C_0 = k_{57}k_{49} \dots k_{36}$ et $D_0 = k_{63}k_{55} \dots k_4$.

2) Définissons v_i ($1 \le i \le 16$) par $v_i = 1$ si $i = 1, 2, 9$ ou 16 et $v_i = 2$ sinon. Alors en utilisant la notation : $C_i \hookleftarrow v_i$ pour indiquer que

S_1-box	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**
0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

l'on opère un décalage à gauche de v_i crans sur la chaine C_i , on effectue, pour i = 1 à 16 :

$$C_i \leftarrow (C_{i-1} \hookleftarrow v_i)$$
$$D_i \leftarrow (D_{i-1} \hookleftarrow v_i)$$
$$K_i \leftarrow PC2 (C_i , D_i)$$

où PC2 est l'application qui sélectionne 48 bits $b_{14}b_{17} ... b_{32}$ sur les 56 bits $b_1b_2 ... b_{56}$ de (C_i, D_i) suivant le tableau :

PC2

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

7.3. Déchiffrement

Le déchiffrement s'effectue en conservant l'algorithme de chiffrement mais en prenant soin d'échanger les permutations IP et IP^{-1}, et d'inverser l'ordre des clés (c'est d'ailleurs tout l'intérêt d'avoir choisi un schéma de Feistel). Si l'on fait abstraction des permutations IP et IP^{-1}, le bloc clair (L_0 , R_0) est chiffré en (R_{16} , L_{16}) . Pendant l'opération de déchiffrement, la première ronde appliquée avec la clé K_{16} donne (L_1', R_1') avec :

$$L_1' = L_{16} = R_{15}$$
$$R_1' = R_{16} \oplus f(L_{16}, K_{16}) =$$
$$= [L_{15} \oplus f(R_{15}, K_{16})] \oplus f(R_{15}, K_{16}) = L_{15}$$

soit $(L_1', R_1') = (R_{15}, L_{15})$. Et ainsi de suite jusqu'à (R_1, L_1) , puis (L_0, R_0).

7.4. Quelques points forts du DES

Le DES est un chiffrement symétrique par blocs, et, à ce titre, il est commode de le considérer comme une simple substitution monoalphabétique sur un alphabet exceptionnellement grand. Le DES agit en effet sur des blocs de 64 bits, de sorte que l'alphabet utilisé ici comporte quelques

$$2^{64} = 18\,446\,744\,073\,709\,551\,616 \approx 10^{19}$$

symboles ! Troquer notre petit alphabet de 26 lettres pour un alphabet aussi large constitue déjà un avantage réel.

La fonction de chiffrement C_K du DES est paramétrée par une clé K. Cette fonction est une bijection de $\mathbf{F}_2^{64}$ dans $\mathbf{F}_2^{64}$ pour permettre le déchiffrement, et l'inverse C_K^{-1} a été explicitée par un algorithme « en tour » à la Section 7.3. Cela nous dévoile deux nouveaux avantages du DES. Tout d'abord le nombre de fonctions de chiffrement est astronomique puisque la bijection C_K est choisie parmi l'une des $(2^{64})!$ permutations de $\mathbf{F}_2^{64}$. Ensuite l'algorithme de déchiffrement — permettant d'appliquer la fonction C_K^{-1} — est simple et en tout point identique à l'algorithme de chiffrement si l'on prend soin d'échanger les permutations du début et de la fin et d'utiliser les clés partielles dans l'ordre inverse. Ce dernier point est aussi important car un système de cryptage doit toujours trouver un équilibre entre la sécurité (qui exige beaucoup de moyens comme de la mémoire ou des microprocesseurs puissants) et les possibilités offertes par les supports (l'électronique embarquée dans la puce d'une carte bancaire est nécessairement limitée).

Un autre intérêt du DES réside dans la « diffusion » des modifications. L'application

successive des rondes où l'on échange grosso modo la partie droite et la partie gauche du mot de 64 bits tout en les modifiant à l'aide d'une clé partielle aléatoire, permet au DES de « diffuser » convenablement les modifications. Cela signifie que la modification d'un unique bit d'entrée du DES entraîne la modification du message chiffré sur plusieurs bits différents et bien répartis parmi les 64 bits. On comprend dès lors l'utilité des rondes.

Pour conclure, rappelons qu'un algorithme faiblit toujours avec le temps à mesure que la puissance de calcul augmente. Ainsi une attaque en force brutale — c'est-à-dire en essayant successivement toutes les (2^{64}) ! permutations de $\mathbf{F}_2{}^{64}$ jusqu'à obtenir un message lisible — est devenue possible et a déjà permis de casser le code DES en utilisant plus de 1500 ordinateurs spécialisés dans le cassage de codes et reliés en réseau. Lors d'une telle attaque, la clé DES ne résiste que quelques heures [11]. Il a donc bien fallu allonger la longueur de la clé pour continuer à garantir l'inviolabilité du système. Le successeur du DES, qui fonctionne sur le même principe, est connu sous le nom de triple DES, et équipe déjà différentes cartes à puces (un tableau fait le point sur les cartes à puces utilisant le triple DES en [7] pp. 43-44).

8. Carte Bancaire Française

La procédure d'authentification d'une CB est différente suivant qu'on utilise un terminal relié au réseau bancaire ou pas.

Si le terminal est relié au réseau CB (ex : DAB), il utilise un DES (et maintenant un triple-DES) avec une clé secrète de 56 bits. Chaque transaction donne lieu à la délivrance d'un Certificat DES (rôle de juge de paix) inscrit sur le ticket, sur la CB et dans le journal du terminal. Ces trois traces permettent de résoudre les litiges.

Si le terminal n'est pas relié, il effectue deux opérations successives :

1) L'identification du porteur, par le code confidentiel.

2) L'authentification de la carte, en utilisant deux niveaux de RSA 320 bits (portés à RSA 792 bits courant année 2000, et RSA 1024 bits très bientôt).

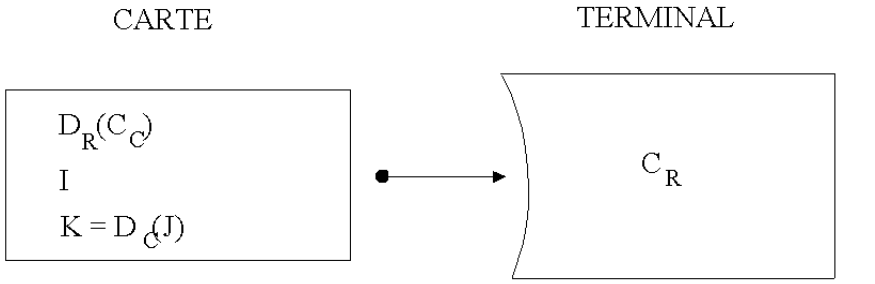
De plus le dialogue entre le terminal et la CB utilise un masque réservé à des sources agréées fourni par un DES de 56 bits. Ce masque contrôle les échanges, et doit être consolidé courant année 2000.

Décrivons maintenant la procédure d'authentification. Le but est de distinguer si la carte est bien homologuée, ou bien s'il s'agit d'une contre-façon. Pour cela, deux valeurs sont inscrite dans toute carte bancaire :

a) Une valeur d'authentification I, qui donne lieu au calcul d'une empreinte J (par exemple, J peut être obtenu par concaténation de I, c'est-à-dire que J=(I | I) représente le mot I répété deux fois).

b) Un mot K représentant le message J chiffré à l'aide d'une clé RSA privée.

Lors du paiement, le terminal lit l'empreinte J et le mot K, puis applique son propre algorithme RSA pour déchiffrer K. Si la valeur J et la valeur calculée à partir de K sont identiques, il estime que la carte est authentique. (voir plus loin et schéma de la page suivante)



Identification du porteur (Code à 4 chiffres)	
Lecture des données sur la carte	
Acquisition de la clé publique C_C de la carte	$C_R\,(\,D_R(C_C)) = C_C$
Déchiffrement de K	$C_C(K)$
Calcul de l'empreinte J de I	
Comparaison de $C_C(K)$ et de J	

Fig. 10

En fait, le terminal ne doit pas nécessairement détenir la clé publique correspondant à la clé secrète de la carte. Il peut seulement conserver une clé publique de référence, délivrée par une « autorité de certification », et c'est la carte elle-même qui lui communiquera sa clé publique signée par la clé secrète de référence.

Ce schéma à deux niveaux de RSA permet d'individualiser les clés par banque, par carte, etc., autorisant du même coup beaucoup plus de souplesse dans la gestion des cartes. Il limite aussi la quantité d'information détenue dans le terminal.

Pour bien comprendre ce processus, notons

C_C la clé publique de la Carte,
D_C la clé secrète de la Carte,
C_R la clé publique de Référence,
D_R la clé secrète de Référence.

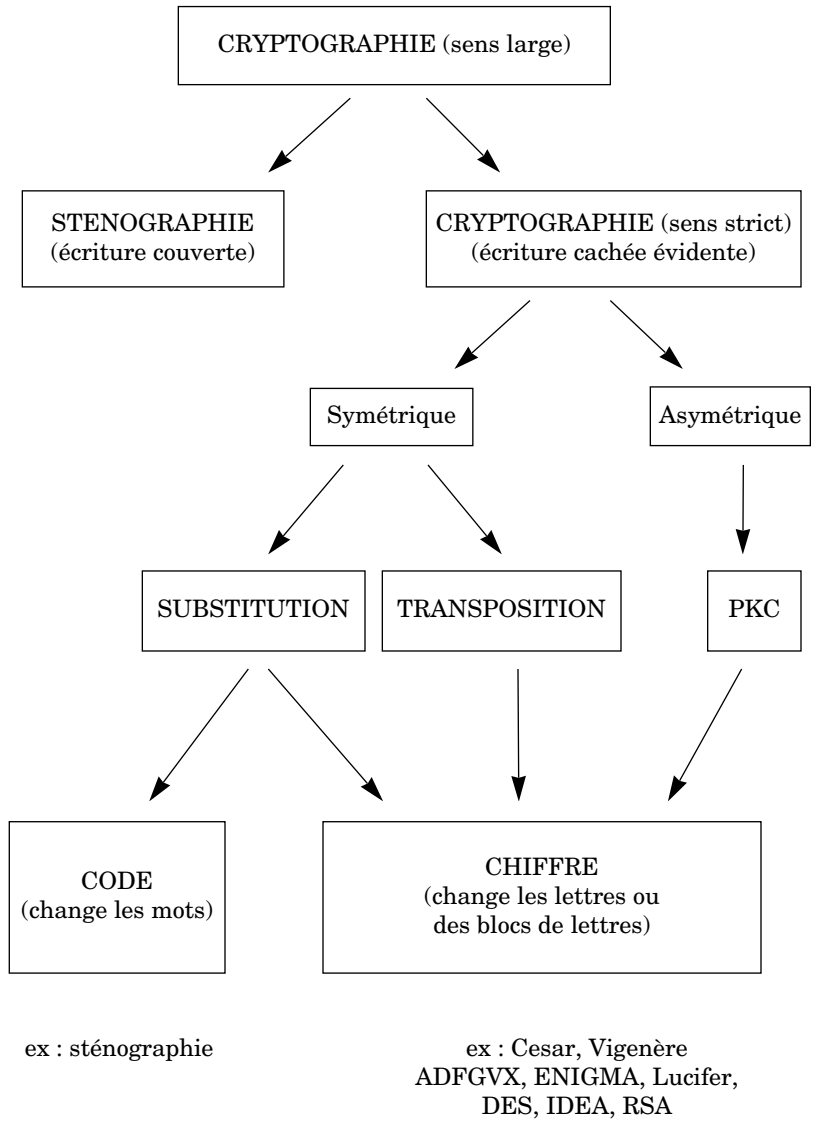
La CB contient les indications suivantes : sa clé publique de chiffrement $D_R(C_C)$ signée par l'autorité de certification, une valeur d'authentification I, et la valeur $K = D_C(J)$ correspondant au chiffrement de J par la clé secrète de la carte. Le terminal conserve seulement la clé C_R. La procédure d'authentification est alors celle qui est représentée sur la figure 10.

a) Identification du porteur : saisie du code à 4 chiffres.
b) Lecture des données sur la carte.
c) Acquisition de la clé publique C_C de la carte par le terminal : celui-ci calcule $C_R(D_R(C_C)) = C_C$.
d) Déchiffrement de K : le terminal calcule $C_C(K)$.
e) Calcul de l'empreinte J de I.
f) Comparaison de $C_C(K)$ et de J.

Le nombre $n = pq$ permettant l'utilisation de RSA au niveau de la clé de référence peut a priori être connu de tout le monde, mais n'avait jamais été distribué massivement par les banques. Récemment, ce nombre d'environ 97 chiffres a été divulgué sur le net (affaire Serge Humpich) et les facteurs premiers p et q ont pu être calculés (c'est la faiblesse actuelle du RSA 320), ce qui a poussé les banques à allonger au plus vite cette clé de sécurité.

ANNEXE 1

Méthodes de la cryptographie



ANNEXE 2

Les algorithmes utilisés dans RSA

Cette annexe contient quelques explications sur la viabilité du système RSA. Au niveau mathématique, le système RSA fonctionne — en particulier — parce qu'il existe des algorithmes rapides qui fournissent de très grands nombres premiers, parce qu'il est facile de résoudre une équation de Bezout, ou encore parce que le calcul exact de x^m modulo n n'est pas un problème « insurmontable » en temps de calcul. Nous poserons donc trois « grandes » question...

1 Comment obtenir un grand nombre premier ?

La recherche d'un nombre premier supérieur ou égal à un grand nombre entier naturel b fixé à l'avance s'effectue de manière très rapide sans qu'il soit besoin de calculer de décompositions en produits de facteurs premiers. Dans [1], les créateurs de RSA proposent d'utiliser le test probabiliste de primalité de Solovay et Strassen. Il s'agit d'abord de savoir si b (impair) est premier ou non avec une probabilité très faible de se tromper. Si b n'est pas premier, il suffit de recommencer le test avec b+1. En continuant de cette façon, on finit par débusquer un nombre $p \geq b$ qui «satisfait le test de Solovay et Strassen», et donc qui est premier avec une probabilité très grande.

Evidemment, avec beaucoup de malchance, il se pourrait que p ne soit pas premier et vérifie tout de même le test. Dans ce cas l'utilisation du système RSA va créer des messages chiffrés qui seront totalement indéchiffrables par le récepteur, et les utilisateurs sauront très vite qu'ils doivent chercher d'autres nombres premiers.

Pour comprendre le fonctionnement du test de Solovay et Strassen, il faut rappeler la définition des symboles de Legendre et de Jacobi. On dit qu'un entier n est un résidu quadratique modulo p s'il existe un entier a tel que $n \equiv a^2 \ [p]$. Si $n \in \mathbf{N}$ et si p désigne un nombre premier, on définit le symbole de Legendre de n modulo p par

$$\left(\frac{n}{p}\right) = \begin{cases} 1 & \text{si } n \text{ n'est pas divisible par } p \text{ et est un résidu quadratique modulo } p \\ -1 & \text{si } n \text{ n'est pas un résidu quadratique modulo } p \\ 0 & \text{si } n \text{ est divisible par } p \end{cases}$$

On démontre alors la formule d'Euler :

Théorème : {[5], Proposition 5.4 p. 109} Si p est un nombre premier différent de 2, alors $\left(\frac{n}{p}\right) \equiv n^{\frac{p-1}{2}} \ [p]$ pour tout entier relatif n.

Si m est un entier relatif et si n est un entier naturel impair, notons $n = p_1 \ldots p_k$ la décom-

position de n en produit de facteurs premiers (éventuellement répétés). Le symbole de Jacobi $\left(\frac{m}{n}\right)$ est défini par : $\left(\frac{m}{n}\right) = \left(\frac{m}{p_1}\right) \ldots \left(\frac{m}{p_k}\right)$. On vérifie alors que $\left(\frac{m}{n}\right)$ ne dépend que de la classe de m modulo n, et que :

$$\left(\frac{m}{np}\right) = \left(\frac{m}{n}\right)\left(\frac{m}{p}\right) \quad \text{et} \quad \left(\frac{mp}{n}\right) = \left(\frac{m}{n}\right)\left(\frac{p}{n}\right)$$

dès que les symboles ont un sens. On a aussi :

Théorème : *Loi de réciprocité* {[5], Proposition 5.14 p. 120} Si m et n sont deux entiers naturels impairs et premiers entre eux,

$$\left(\frac{m}{n}\right) = (-1)^{\frac{(m-1)(n-1)}{4}} \left(\frac{n}{m}\right).$$

Théorème : *Lois complémentaires* {[5], Proposition 5.15 p. 120} Si n est un entier impair positif,

$$\left(\frac{-1}{n}\right) = (-1)^{\frac{(n-1)}{2}} \quad \text{et} \quad \left(\frac{2}{n}\right) = (-1)^{\frac{(n+1)(n-1)}{8}}.$$

Tous ces résultats permettent de donner l'algorithme suivant de calcul de :

$$J(m, n) := \left(\frac{m}{n}\right)$$

lorsque m et n sont des entiers naturels premiers entre eux et lorsque n est impair :

$J(m, n)$ = Si $m = 1$ alors 1, sinon si m est pair alors $J(\frac{m}{2}, n) \times (-1)^{\frac{(n+1)(n-1)}{8}}$

sinon $J(n \bmod m, m) \times (-1)^{\frac{(m-1)(n-1)}{4}}$.

Avec Maple, on obtient :

```
jacobi := proc(m,n)
if m=1 then 1
else
    if type(m,even) = true then jacobi(m/2,n)*(–1)^((n^2–1)/8)
        else jacobi(n mod m,m)*(–1)^((m–1)*(n–1)/4);
        fi;
    fi;
end;
```

A partir de là, et en notant φ l'indicateur d'Euler, le résultat crucial est le suivant :

Théorème de Solovay et Strassen {[5], Proposition 5.18 et Corollaire 5.19} Soit n un entier naturel impair > 2.

1) Alors n est premier si et seulement si $\left(\frac{a}{n}\right) \equiv a^{\frac{n-1}{2}} \ [n]$ pour tout a appartenant à $\{1, \ldots, n-1\}$ et premier avec n.

2) Si n est composé, le cardinal de l'ensemble

$$E_n = \left\{ a \in \{1, \ldots, n-1\} \,/\, \text{pgcd}(a, n) = 1 \text{ et } \left(\frac{a}{n}\right) \equiv a^{\frac{n-1}{2}} \ [n] \right\}$$

est $\leq \frac{\varphi(n)}{2}$.

Pour déterminer si un entier n est premier, on procède de la façon suivante. On choisit un nombre a au hasard dans $\{1, \ldots, n-1\}$, puis on regarde s'il appartient à E_n. Si a n'appartient pas à E_n, on dit que le test a échoué, et l'on sait que n n'est pas premier. Si a appartient à E_n, on dit que le test a réussi, et on le recommence avec un autre entier a. Si n passe avec succès N tests successifs, on peut considérer que n est premier avec une probabilité de se tromper inférieure à $\left(\frac{\varphi(n)}{2} / (n-1)\right)^N$, donc inférieure à $\frac{1}{2^N}$.

Avec seulement $N = 100$ tests consécutifs on peut déterminer si un nombre est premier avec une probabilité d'erreur inférieure à $\frac{1}{2^{100}} \approx 10^{-31}$.

2. Comment choisir c et d ?

Les entiers $n = pq$ et $m = (p-1)(q-1)$ sont connus, et il s'agit de déterminer c, d et k tels que $cd = km + 1$. Le choix d'un entier d premier avec m et arbitrairement grand — pour ne pas faciliter la cryptanalyse — ne pose pas de problème : il suffit de choisir n'importe quel nombre premier d supérieur à $\text{Max}(p, q)$. Avec ce choix, en effet, supposer $\text{pgcd}(d, m) \neq 1$ implique que d divise m, donc divise $(p-1)$ ou $(q-1)$, et contredit l'hypothèse $d \geq \text{Max}(p, q)$. Une fois d choisi, la recherche de c et k se fait par le classique *algorithme d'Euclide étendu*.

3. Comment calculer rapidement une puissance modulo n ?

Le calcul de x^m modulo n est rapide si l'on utilise l'écriture

$$m = a_k 2^k + a_{k-1} 2^{k-1} + \ldots + a_1 \times 2 + a_0$$

de m en base 2 et si l'on décompose x^m de la façon suivante :

$$x^m = \left(\left((x^2 \times x^{a_{k-1}})^2 \times x^{a_{k-2}} \right)^2 \ldots \right)^2 \times x^{a_0}.$$

On obtient alors l'algorithme d'exponentiation par carrés et multiplications suivant :

Dans la variable c placer 1.
Pour i variant de k à 0 faire les deux pas suivants
c := $c^2 \mod n$.
Si a_i :=1 alors faire $c := c * x \mod n$.
La variable c contient x^m modulo n.

Par exemple, lorsqu'on fait tourner cet algorithme pour calculer $x^{2^4+2^3+2+1}$, la variable c prend successivement les valeurs suivantes :

$$1 \longrightarrow 1 \longrightarrow x \underset{i=3}{\longrightarrow} x^2 \longrightarrow x^2 \times x \underset{i=2}{\longrightarrow} (x^2 \times x)^2 \underset{i=1}{\longrightarrow} ((x^2 \times x)^2)^2 \longrightarrow ((x^2 \times x)^2)^2 \times x$$

$$\underset{i=0}{\longrightarrow} (((x^2 \times x)^2)^2 \times x)^2 \longrightarrow (((x^2 \times x)^2)^2 \times x)^2 \times x .$$

et l'on obtient bien $x^{2^4+2^3+2+1} = (((x^2 \times x)^2)^2 \times x)^2 \times x$.

On peut vérifier que cet algorithme, qui utilise au plus $2\log_2 m$ multiplications modulo n, est plus performant que l'algorithme naïf qui multiplie par x et réduit modulo n à chaque boucle.

Bibliographie

[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] F.L. Bauer, Decrypted Secrets, Methods and Maxims of Cryptology, 2nd ed., Springer-Verlag, 2000 (Un ouvrage précis et bien documenté, en anglais).

[3] J.-P. Delahaye, La cryptographie RSA vingt ans après, Pour la Science, n° 267, janvier 2000.

[4] S. De Lastour, La France gagne la guerre des codes secrets 1914-1918, Editions Tallandier, 1998.

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

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

[7] J. Donio, J. L. les Jardins, E. de Rocca et M. Verstrepen, La Carte à Puce, Coll. Que sais-je ? PUF, 2ème édition, 2000.

[8] Le Chiffre pendant la Grande Guerre, Dossier concours, revue Historia n° 623 de novembre 1998.

[9] Les secrets d'ENIGMA, revue L'Histoire n° 247 d'octobre 2000, p. 4.

[10] A. Menezes, P. van Oorschot, S. Vanstone, Handbook of Applied Cryptography, CRC Press, 1997 (Excellent ouvrage très complet sur la cryptographie et proposé gratuitement par l'éditeur en téléchargement sur le site http://www.cacr.math.uwaterloo.ca/hac/. Le DES est expliqué pp. 250-258).

[11] A. Nekrassov, Le langage secret de la Carte Bancaire, Le Monde Informatique du 16 juin 2000 (Renseignements pratiques concernant le double niveau de RSA de la CB).

[12] PGP : le site http://www.pgpi.org/ contient de la documentation sur le logiciel de chiffrement asymétrique PGP (Pretty Good Privacy) et permet le téléchargement de la dernière version gratuite.

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

[14] S. Singh, Histoire des Codes Secrets, De l'Egypte des Pharaons à l'ordinateur quantique, JC Lattès, 1999 (Une histoire de la cryptographie d'accès rendu facile, et qui se lit comme un roman).

[15] Références à la biclé RSA et au masque DES de la CB relevées en
http://www.bull.fr/securinews/courant/31-02_1.html,
et renvoyant sur l'hebdomadaire 01 Informatique n° 1580 du 17 mars 2000.

[16] Sécurité et Informatique, Revue du SCSSI (Service Central de la Sécurité des Systèmes d'Informations), CNRS (A télécharger sur le site :
http://www.cnrs.fr/Infosecu/Revue.html.
Le n° 24 d'avril 1999 contient des informations précises sur les aspects et le rôle de la cryptographie, sur les produits de cryptologie, ainsi que sur la législation en cours.