INTRODUCTION À LA

CRYPTOGRAPHIE

Emeric Gioan

2003/2004

maîtrise d'informatique
(& maîtrise de mathématiques en option)

université de La Réunion

La cryptographie (art de crypter des messages), et son ennemie la cryptanalyse (art de déchiffrer des messages codés), sont des disciplines scientifiques omniprésentes aujourd'hui dans la société, aussi bien civile que militaire, commerciale ou politique.

Ce cours d'introduction présente les algorithmes de cryptographie couramment utilisés à la fois :
- d'un point de vue culturel, à travers l'histoire et l'actualité,
- d'un point de vue théorique, par des résultats mathématiques surtout en théorie des nombres, mais aussi en probabilités et en complexité,
- d'un point de vue pratique, par leur mise en place informatique, accompagnée des protocoles indispensables (identification, authentification...).

Le but est de donner ici un aperçu de la grande variété de problèmes et de pratiques relatifs à la cryptographie, quitte à ce que la plupart des notions ne soient pas approfondies.

Sommaire

0 Introduction

Le principe général de la *cryptographie* est résumé dans le schéma suivant([1]) : Aristide (A) veut envoyer un message secret à Barnabé (B). Aristide crypte alors son message de façon à ce que seul Barnabé puisse le décrypter. De son côté, Clotaire l'adversaire (C) intercepte le message codé, et tente la *cryptanalyse* de ce message, c'est à dire de retrouver le message d'origine.

$$\text{A émet } m \longrightarrow \text{chiffrement } c = E(m) \overset{\text{canal non sécurisé}}{\underset{\uparrow}{\longrightarrow}} \text{déchiffrement } m = D(c) \longrightarrow \text{B reçoit } m$$
$$\text{C intercepte}$$

De manière générale, et du point de vue mathématique, la cryptographie moderne utilise beaucoup la théorie des nombres, mais fait aussi appel à l'algèbre linéaire, la théorie des groupes, la théorie de la complexité, la théorie combinatoire et la théorie des graphes. La cryptanalyse se fonde aussi beaucoup sur la théorie des probabilités et des statistiques.

Ce cours d'introduction commence dans la partie 1 par une présentation progressives des algorithmes de chiffrement à clé secrète à travers l'histoire. La partie 2 présente les algorithmes modernes de ce type, jusqu'à l'algorithme Rijndael devenu en 2002 le standard des chiffrements à clé secrète, puis des techniques générales associées à la cryptographie. La partie 3 présente l'autre branche de la cryptographie moderne : les chiffrements à clé publique, qui utilisent des fonctions de théorie des nombres faciles à calculer mais impossible à inverser rapidement avec les techniques actuelles. La partie 4 présente la mise en pratique de la cryptographie à travers les signatures, les protocoles pour échanger publiquement des informations secrètes, et le procédé des certificats d'authentification. Enfin la partie 5 donne quelques compléments.

• **Notations.**
$[n]$ désigne "modulo n".
Si $a, b \in F_2 = \{0, 1\}$ sont deux *bits*, l'opérateur logique "ou exclusif" est noté :

$$a + b = a + b[2] = a \oplus b = a \ XOR \ b.$$

• Premières définitions.

cryptage (ou *chiffrement*) : moyen par lequel on transforme un message pour qu'il ne soit lisible que par son destinataire.

cryptologie : discipline scientifique étudiant le cryptage d'information, comprenant la cryptographie et la cryptanalyse.

cryptographie ("secret writing") : chiffrement ou cryptage des messages en clair et déchiffrement ou décryptage de messages codés, connaissant la clé.

([1]) Les noms Aristide, Barnabé et Clotaire, pour A, B et C, sont empruntés à la traduction française du livre "Logique sans peine" de Lewis Carroll.

cryptanalyse ("code breaking") : art de décrypter les messages codés sans connaître la clé. La complexité d'une attaque est mesurée par la place et le temps requis.

texte clair : message m à envoyer au destinataire

texte chiffré : message c obtenu à partir de m que seul le destinataire est censé pouvoir transformer en m.

cryptosystème : système (cryptographique) formé d'un algorithme de cryptage E (chiffrement, "encrypt") et d'un algorithme de décryptage D (déchiffrement, "decrypt") :

$$E(m) = c \quad D(c) = m$$

système de chiffrement à clé privée ou secrète, ou *algorithmes symétriques* : la clé servant à chiffrer les données peut-être facilement déterminée si l'on connaît la clé servant à déchiffrer, et réciproquement (voir parties 1 et 2).

système de chiffrement à clé publique, ou *algorithmes asymétriques* : la clé publique d'un destinataire, accessible à tous, permet de lui envoyer des messages cryptés, que lui seul pourra décrypter grâce à sa clé privée personnelle (voir partie 3).

• Objectifs de la cryptographie moderne

La prolifération des ordinateurs et des systèmes de communication dans les années 1960 à amené le secteur privé à protéger l'information sous forme numérique et à fournir des service sécurisés.

La sécurité des données chiffrées est entièrement dépendante de deux choses : la force de l'algorithme cryptographique, et le secret de la clé.

Le but fondamental de la cryptographie est de respecter adéquatement les quatre objectifs majeurs de la sécurité, en théorie et en pratique :

confidentialité : le message crypté doit rester secret, ne peut être décrypté par un tiers. C'est l'aspect principal de la cryptographie qui sera étudié dans ce cours.

authentification : assurance de l'authenticité, notamment de l'expéditeur ou de l'origine. Cet aspect indispensable dans les protocoles pratiques sera étudié à la fin de ce cours. Les moyens employés sont souvent dérivés des moyens cryptographiques habituels, et d'un autre côté de nombreuses attaques peuvent être portés aux cryptosystèmes si on ne s'assure pas de l'authenticité des interlocuteurs...

intégrité : assurance que le message n'a pas été modifié durant la transmission. Cet aspect ne sera pas étudié dans ce cours, il lié essentiellement aux problèmes techniques de transmission de données. Il est cependant à l'origine de la liaison entre la cryptographie et les *codes correcteurs* dont le principe est d'ajouter de l'information aux messages afin de pouvoir retrouver le message d'origine même si une erreur est survenue pendant la transmission.

non répudiation : l'expéditeur ne peut pas nier, ultérieurement, avoir envoyé le message. Cet aspect est ici sous-entendu dans l'authentification.

1 Evolution de la cryptographie

Les origines de la cryptographie remontent à l'antiquité. On verra dans cette partie, à travers des exemples historiques, comment l'évolution des problèmes de cryptographie symétrique a abouti aux techniques modernes.

1.1 Chiffrements par permutation

• *Le scytale* : en 487 av. J-C, pendant la guerre de Sparte qui opposait les Perses et les Grecs, ces derniers utilisaient pour coder leurs messages militaires un ruban et un bâton appelé *scytale*. Le ruban était enroulé autour du bâton, et le message écrit en colonne verticales le long du bâton. Ainsi il fallait avoir un bâton de même diamètre pour retrouver le message à partir du ruban déroulé. Dans ce cas, *la clé de chiffrement est le diamètre du scytale.*

• *Chiffrement par transposition.*

Cryptage : le message en clair m de taille $n = dk$ est écrit en colonnes de d lettres, et le message codé c est formé des lignes mises bout à bout. La clé de chiffrement est d.

Décryptage : connaissant la clé de déchiffrement $k = n/d$, on retrouve m à partir de c en écrivant le message en lignes de taille k et en lisant les colonnes.

Exemple.

V	I		S		R
O		M	A	S	E
I	L	E	G	E	T
C	E	S	E	C	

$m =$ "VOICI LE MESSAGE SECRET"

$c =$ "VI S RO MASEILEGETCESEC "

• Cryptanalyse : ce chiffrement est facile à casser puisqu'il suffit d'essayer toutes les clés possibles (entier d inférieur à la taille du message), et de vérifier à chaque fois si le message obtenu a un sens.

• *Chiffrement par permutation.*

Pour un message $m = m_1...m_n$ de taille n fixée, et une clé σ, permutation de $\{1, ..., n\}$:

Cryptage : $c_i = m_{\sigma(i)}$
Décryptage : $m_i = c_{\sigma^{-1}(i)}$

Le chiffrement par transposition est un cas particulier rudimentaire de chiffrement par permutation.
Remarque. Ce type de chiffrement permet d'augmenter la *diffusion* des lettres dans le message : quelqu'un qui verrait le message codé ne doit pas pouvoir facilement isoler les mots (même cryptés) du message, ou bien savoir quels mots font partie d'une même phrase. Ce principe est encore employé aujourd'hui, mais avec des méthodes plus complexes, au sein des algorithmes de chiffrement à clé secrète.

1.2 Chiffrements par substitution

• *Chiffre de César* : aux alentours de -50 av. J-C, l'empereur romain Jules César envoyait les messages à ses généraux en remplaçant toutes les lettres du message par la troisième suivante dans l'alphabet. Dans ce cas, la clé est le nombre de lettres utilisé pour le décalage. Ce code est très facile à casser si l'on sait quel est l'algorithme employé.

On considère les lettres comme des nombres([2]) de 1 à 26, $m = m_1 m_2 ... m_n$, $c = c_1 c_2 ... c_n$. Un cryptosystème par décalage de k lettres s'écrit :

Cryptage : $c_i := m_i + k[26]$	(remplacer la lettre i par la lettre $i + k[26]$)
Décryptage : $m_i := c_i - k[26]$	(remplacer la lettre i par la lettre $i - k[26]$)

Exemple. Avec $k = 3$,

$$m = \text{“BONJOUR”},$$

$$c = \text{“ERQMRXU”}.$$

Pour l'anecdote... Dans le film de Stanley Kubrick “2001 l'odyssée de l'espace”, l'ordinateur du futur s'appelle “HAL”, ce qui est un cryptage pour “IBM” avec un décalage alphabétique de 1.

• *Chiffrement par substitution monoalphabétique.*
On remplace chaque lettre du message par la lettre correspondante dans une bijection ϕ avec autre alphabet. Le décryptage se fait par la bijection réciproque.

Cryptage : $c_i := \phi(m_i)$
Décryptage : $m_i := \phi^{-1}(c_i)$

Exemple. A↔D, B↔K, C↔¡, D↔%, E↔3 ...

([2]) “modulo 26” pour les lettres signifie que lorsqu'on arrive au bout de l'alphabet par le décalage on recommence au début.

Le chiffre de César est un cas particulier où l'alphabet de substitution est l'alphabet de départ décalé. Ce type de systèmes a été très couramment utilisé à travers les âges.

Remarque. L'intérêt de ce type de chiffrement est d'augmenter la *confusion* du texte crypté. C'est à dire que les symboles utilisés n'ont plus leur sens habituel.

• Cryptanalyse.

La méthode employée pour casser ce type de code a été découverte au IXème siècle par le savant arabe Al-Kindi. Elle utilise les fréquences d'apparition des lettres dans les langages vivants. Par exemple en Français les lettres les plus fréquemment employées sont dans l'ordre([3]) : ESANTIRULO et en anglais : ETAONIRSHD.

On retrouve ainsi facilement certaines lettres du message, et le reste se fait par tâtonnements en essayant de deviner les mots selon le contexte et l'information déjà obtenue. Ceci n'est pas très scientifique mais la cryptanalyse est une discipline très empirique. A l'époque moderne aussi et peut-être encore plus étant donnés les enjeux, tous les moyens sont bons (voir plus loin).

Pour l'anecdote... Dans le livre "Le Scarabée d'Or" d'Edgar Poe, le personnage résout une énigme exactement de cette façon.

Pour éviter ce type d'attaque, le cryptographe peut complexifier ce système : employer plusieurs lettres pour une seule, introduire des lettres insignifiantes, utiliser des abréviations...

• *Chiffrement par substitution polyalphabétique.*

On remplace chaque lettre du message par une ou plusieurs lettres correspondante dans un autre alphabet. Le décryptage se fait par la fonction réciproque.

Exemple. A↔D ou E, B↔K ou é, C↔¡ ou ¿, D↔% ou 7, E↔3 ou 4 ou V...

Remarque. L'intérêt de ce chiffrement est qu'il augmente la *dispersion* des lettres, c'est à dire qu'il fausse leur probabilité d'apparition.

• Cryptanalyse.

Ce type de chiffrement est encore possible à casser avec des moyens statistiques plus élaborés. Par exemple, il existe aussi des statistiques sur les couples de lettres les plus fréquents (ES en français, TH en anglais...). Voir [Sa] pour plus d'informations. On peut utiliser aussi des résultats de linguistique statistique comme dans l'exemple suivant.

([3]) Ces ordres de lettres diffèrent légèrement selon les sources.

1.3 Un exemple historique raffiné

• *Le code de Vigenère* : inventé par Blaise de Vigenère en 1586, ce cryptosystème est en quelque sorte un raffinement du chiffre de César, car l'alphabet de substitution change avec la place de la lettre à crypter dans le message et dépend d'un *mot clé*.

La clé est un mot $t_0 t_1 ... t_{n-1}$. On note $m = m_1 m_2 ....$, et $c = c_1 c_2 ....$

Cryptage :

$$c_i = m_i + t_{i-1[n]} - 1[26]$$

Décryptage :

$$m_i = c_i - t_{i-1[n]} + 1[26]$$

Autrement dit : la clé étant un mot de taille n sur l'alphabet $1, ..., 26$, on remplace chaque lettre m_i du message par la $k-1$-ième suivante dans l'alphabet (modulo 26), où k est la lettre de la clé numéro i modulo n (numéro n lorsque $i = 0[n]$).

A	B	C_1	D	E	F	G	H	I	J	K	L	M	N	O	P_4	Q	R_2	S	T	U	V	W	X	Y_3	Z
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
C_4	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	$\mathbf{R}_4$	S	T	U	V	W	X	Y	Z	A	B
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
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
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
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
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
I_3	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	$\mathbf{G}_3$	H
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
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
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
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
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
O_2	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	$\mathbf{F}_2$	G	H	I	J	K	L	M	N
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
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
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
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
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
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
V_1	W	$\mathbf{X}_1$	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
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
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
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
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

Exemple. En pratique, on écrivait le mot clé (ici "CRYPTO"), de façon répétée, au dessus du message à coder, et on lisait les lettres codées dans le tableau précédent. On commence par repérer dans la ligne du haut du tableau la lettre de la clé à utiliser 'par exemple "C" pour la première lettre du message m suivant), puis dans la colonne de gauche du tableau la lettre du message à chiffrer (ici "V"), alors la lettre cryptée est à l'intersection de la colonne et de la ligne correspondantes (ici "X"). Les indices dans le tableau indiquent les lettres employées pour coder les 4 premières lettres du message m.

clé C R Y P T O C R Y P T O C R Y P T O C R Y P T

m =V O I C I U N M E S S A G E S E C R E T

c = X F G R ...

Ce procédé fausse a priori complètement les probabilités d'apparition des lettres puisque la même lettre dans le message crypté peut correspondre à des lettres différentes du message initial.

• Cryptanalyse.

Si on trouve la taille k de la clé, on sait que pour chaque ensemble de lettres prises une lettre toute les k lettres, la substitution est monoalphabétique (les lettres numéro i, $i+k$, $i+2k$..., sont codées avec le même alphabet déduit de la lettre numéro i de la clé). Et on sait casser les chiffrements par substituion monoalphabétique (partie 1.2).

Il reste donc à déterminer la longueur de la clé [JM].

- première méthode (test de Kasiski) : on remarque que *si on répète une même suite de lettres dans le message clair à une distance qui est un multiple de la longueur de la clé, le message crypté répètera aussi une même suite de lettres.* Ainsi, si on trouve dans le message crypté une suite de lettres qui se répète à une distance k, on peut espérer que la taille de la clé soit un diviseur de k. Cette méthode est efficace si le message est suffisamment long pour trouver de telles suites de lettres et donc suffisamment d'information sur la longueur de la clé.

- deuxième méthode (test de Friedman) : de façon fine et surprenante, on va utiliser justement le fait que le code disperse les lettres pour le casser plus facilement. On utilise pour cela un peu de *théorie des probabilités.* On suppose pour la suite que l'alphabet a 26 lettres.

On appelle l'*indice de coïncidence* I d'un texte la probabilité de trouver deux lettres identiques en tirant un couple de lettres au hasard dans ce texte. Si le texte est de longueur n et si n_i est le nombre d'apparitions de la lettre i dans le texte, on calcule facilement

$$I = \frac{\sum_{1 \leq i \leq 26} n_i(n_i - 1)}{n(n-1)}$$

que l'on peut approcher pour un texte assez long par

$$I \simeq \frac{\sum_{1 \leq i \leq 26} n_i^2}{n^2}$$

Si le texte est suffisamment long et écrit dans une langue naturelle, la proportion $\frac{n_i}{n}$ de lettres i dans le texte est environ p_i probabilité d'apparition de la lettre i lorsqu'on considère une lettre au hasard dans un texte de cette langue. Comme on l'a vu précédemment ces valeurs p_i sont approximativement connues.

Ainsi

$$I \simeq \sum_{1 \leq i \leq 26} p_i^2$$

est connu *pour un texte d'une langue donnée* : cette valeur est environ 0,074 en français, 0,065 en anglais.

Par contre *si le texte est une suite de symboles aléatoires* parmi 26 possibles, on devrait avoir $\frac{n_i}{n} \simeq \frac{1}{26}$ et donc

$$I \simeq \frac{1}{26} \simeq 0,038$$

qui est bien inférieur à la valeur trouvée dans un texte naturel. De manière générale, on a la propriété suivante.

Résultat : *plus la dispersion des lettres dans le texte augmente (autrement dit plus le texte est aléatoire), plus l'indice I diminue.*

Or d'autre part, l'indice I ne change pas par une substitution monoalphabétique (en effet on prend en compte tous les symboles), ni par une transposition des lettres (l'ordre d'apparition des lettres n'intervient pas).

Donc le calcul de I permet de savoir si un texte est crypté par une substitution monoalphabétique ou bien par une substitution polyalphabétique, même si une transposition a été employée en plus.

Dans le cas du code de Vigenère il suffit donc d'extraire du message chiffré une lettre sur k et calculer l'indice I de ce sous-message. Si I est inférieur à la valeur standard pour une langue naturelle, on déduit que la répartition des symboles dans le sous-message n'est pas naturelle, c'est à dire que le sous message a été obtenu par une substitution polyalphabétique, c'est à dire que k n'est pas multiple de la longueur de la clé. Par contre si I s'approche de la valeur d'une langue naturelle, on déduit que la substitution est monoalphabétique pour ce sous-message, c'est à dire que k est multiple de la longueur de la clé.

• Réponse du cryptographe : pour éviter cette cryptanalyse, il suffit de coder le début du message avec la clé, puis d'utiliser ensuite le message clair lui même comme clé de chiffrement. Ce procédé évite la cryptanalyse précédente, mais l'inconvénient est qu'une erreur de transmission se répercute tout au long du message. C'est ensuite au cryptanalyste de trouver une parade et ainsi de suite... Une autre réponse possible pour éviter la cryptanalyse précédente est de prendre un mot clé aussi long que le message.

Il faut remarquer que si le mot clé est aussi long que le message, le code de Vigenère est incassable (en effet la clé pouvant être quelconque, aucune information sur la clé ou sur le message clair ne peut-être obtenue à partir du message codé). Ce principe a été repris quelques siècles plus tard, avec le fameux cryptosystème suivant.

1.4 Le chiffre (incassable) à clé jetable

Inventé par Gilbert Vernam en 1918, ce cryptosystème est en quelque sorte un raffinement du code de Vigenère, car l'alphabet de substitution change avec la place de la lettre à crypter dans le message, et dépend d'un *mot clé*, mais qui est cette fois aléatoire et aussi long que le message.

Pour changer un peu, et pour se rapprocher du cadre informatique, on considère que l'alphabet est $\{0,1\}$.

$m = m_1...m_n$, $c = c_1...c_n$ et la clé $t_1t_2...t_n$ sont des mots de n bits.

Cryptage :

$$c_i = m_i + t_i[2]$$

Décryptage :

$$m_i = c_i - t_i[2]$$

Autrement dit : la clé étant un mot de taille n sur l'alphabet $\{0,1\}$, on remplace chaque bit m_i du message par le bit $m_i \ XOR \ t_i$.

Résultat : *ce cryptosystème est incassable car celui qui l'intercepte dispose d'une suite aléatoire de bits (puisque le mot t est aléatoire) et n'a donc aucune information sur le message ou sur la clé.*

Pour l'anecdote... Ce système a été longtemps employé par le *téléphone rouge* reliant Moscou et Washington, qui marchait plutôt comme un télégraphe.

Puisque tout repose sur le fait que la clé soit aléatoire, la clé ne doit servir qu'une seule fois et être ensuite jetée, d'où les noms *one time cipher, one time pad* ou *chiffre à clé jetable* donnés à ce code. Ce système a donc deux inconvénients majeurs :

- il faut transporter la clé d'une façon ou d'une autre pour que le récepteur et l'émetteur la connaissent (par exemple par une valise diplomatique), mais ce n'est guère pratique, et on n'est pas à l'abri d'une attaque au niveau humain (quelqu'un peut s'emparer de la clé).

- il est impossible de programmer de l'aléatoire... Les ordinateurs, fonctionnant de façon purement déterministe, ne peuvent générer que des suites pseudo-aléatoires (fonctions *random*), voir partie 2.6.

1.5 Vers la cryptographie moderne

Durant la seconde guerre mondiale, l'armée allemande utilisait une machine faite de rotors appelée *Enigma* pour crypter ses messages. Quelques scientifiques polonais ont réussi à obtenir des informations sur le système employé, puis plus d'une dizaine de milliers de chercheurs alliés ont tenté de casser le code allemand. Finalement l'équipe d'Alan Turing([4]) a terminé de résoudre le problème, et certains estiment que ce succès a permis de raccourcir la guerre de deux ans, économisant des millions de morts supplémentaires.

Ceci a fait prendre conscience à l'humanité de l'importance et des enjeux de la cryptographie dans l'époque moderne. Avec le développement des communications et de l'informatique à partir des années 1960, tout s'est accéléré et la cryptologie est devenue une branche extrêmement étudiée des mathématiques et de l'informatique. Aujourd'hui les organismes américains chargé d'étudier les cryptosystèmes sont les plus gros employeurs de mathématiciens au monde.

On a vu ici apparaître un problème majeur de la cryptographie moderne : comment générer des suites pseudo-aléatoires, et comment crypter les messages pour qu'ils donnent l'impression d'être aléatoires au cryptanalyste éventuel, et ce de façon la plus sécurisée possible et par des moyens pas trop encombrants. Les techniques modernes d'élaborations de chiffrement à clé secrète, tentent de répondre à cette demande, en combinant de façon très poussée des algorithmes de cryptage simples comme ceux que l'on vient de voir.

([4]) *Pour l'anecdote...* Alan Turing fût un visionnaire et l'un des pères de l'informatique, inventeur des *machines de Turing* (fondement du formalisme actuel de l'algorithmique), concepteur du premier ordinateur... On peut dire que l'humanité a été assez peu reconnaissante envers lui car quelques années à peine après avoir contribué à sauver des millions de vies, il a été condamné pour homosexualité, contraint à ingurgiter des hormones féminines, pour le stériliser, qui lui ont fait pousser les seins et l'ont poussé au suicide.

2 Cryptographie à clé secrète

Les systèmes de cryptage à clé privée, appelés aussi systèmes de cryptage symétriques, sont utilisés depuis déjà plusieurs siècles (cf. tous les exemples de la partie 2). C'est l'approche la plus authentique du chiffrement de données, et mathématiquement la moins problématique.

Dans ce type de système, la clé servant à chiffrer les données peut-être facilement déterminée (à l'aide d'ordinateurs) si l'on connaît la clé servant à déchiffrer, et réciproquement. Dans la plupart des systèmes symétrique, la clé de chiffrement est la même que la clé de déchiffrement.
Autres termes anglais utilisés : *single-key, one-key, private-key, conventional encryption.*

2.1 Schéma de Feistel

Soit $\mathcal{M} = \{0,1\}^{2n} = \mathcal{C}$ l'ensemble des messages possibles à $2n$ bits (clairs ou chiffrés). Pour $M \in \mathcal{M}$, on décompose M en deux blocs successifs de n bits (L pour "left" et R pour "right") chacun

$$M = (L, R).$$

La clé K est un ensemble de vecteurs binaires $K = \{K_1, ..., K_d\}$ chacun de longueur k. Soit F une application

$$F : \{0,1\}^{n+k} \to \{0,1\}^n.$$

Cryptage. Entrée : $M = (L_0, R_0)$
pour i variant de 1 à d faire

$$L_i := R_{i-1}$$

$$R_i := L_{i-1} + F(K_i, R_{i-1})$$

Sortie : $C = (L_d, R_d)$
Décryptage. Entrée : $C = (L_d, R_d)$
pour i variant de d à 1 faire

$$L_{i-1} := R_i + F(K_i, L_i)$$

$$R_{i-1} := L_i$$

Sortie : $M = (L_0, R_0)$

Ce schéma de chiffrement a été très employé durant les dernières décennies. Le premier standard **DES** *a été conçu sur ce principe (voir partie 2.2 ci-dessous), un autre algorithme usuel sur ce principe est* **Blowfish** *conçu en 1993 par Bruce Schneier. L'idée de faire interagir des parties distinctes du message pour disperser les informations se retrouve d'ailleurs dans les procédés de chaînage, et de hachage (voir parties 2.5 et 4.2). Cependant le nouveau standard de chiffrement l'AES 2002, le* **Rijndael**, *sort de ce cadre et complique encore la dispersion des informations (partie 2.3).*

2.2 L'algorithme DES : Data Encryption Standard

Le DES, ou "Standard de chiffrement de données", a été conçu en 1976 à partir de l'algorithme Lucifer développé par IBM au début des années 1970. Jusqu'à ce qu'il soit cassé en 1997, cet algorithme est resté la norme de chiffrement choisie par le gouvernement américain.

Cet algorithme fonctionne selon un schéma de Feistel de la façon suivante.

$2n = 64$, $d = 16$ (chiffrement à 16 tours), $k = 48$,
$(K_1, ..., K_n)$ est déduit d'une clé K_0 à 56 bits,

$$F(K_i, R) = \pi(S(E(R) \oplus K_i))$$

avec $E : F_2^{32} \to F_2^{48}$ linéaire,
$S : \{0,1\}^{48} \sim \{0,1\}^{6^8} \to \{0,1\}^{4^8} \sim \{0,1\}^{32}$ non linéaire
induite par une transformation $\{0,1\}^6 \to \{0,1\}^4$ non linéaire,
et π permutation des coordonnées.

On estimait à l'époque de sa création que cet algorithme serait sûr pendant de nombreuses décennies, c'était sans compter la rapidité de l'évolution des technologies de l'informatique. En 1997 le DES est cassé en 3 semaines par une fédération de petites machines sur Internet, par une simple recherche exhaustive de la bonne clé parmi les 2^{56} possibles. Le DES a été alors remplacé par le **Triple DES** (algorithme DES appliqué trois fois de suite avec une première clé, puis une deuxième, puis à nouveau la première, voir aussi l'exercice 2), largement suffisant à l'heure actuelle.

Cependant dès 1997, le NIST (National Institue of Standards and Technologies) lance un appel international pour trouver un standard de chiffrement successeur du DES. Le vainqueur est l'algorithme suivant.

2.3 L'algorithme Rijndael, AES : Advanced Encryption Standard

Conçu en Belgique en 1998 par Joan Daemen et Vincent Rijmen, cet algorithme a été testé en même temps que de nombreux concurrents pendant 4 ans, et a été élu en 2002 pour devenir le nouveau standard de chiffrement de données à clé secrète, l'**AES**.

La longueur de la clé peut varier entre 128 et 256 bits, par incréments de 32 bits, ce qui permet une adaptation à des supports variés. On considère ici le cas standard.

Une suite de 128 bits est considérée comme un tableau d'octets de 4 lignes et 4 colonnes.

entrée : clé K de 128 bits, message M de 128 bits.

0 - addition de la clé initiale

$$M_1 := M \oplus K$$

pour i variant de 1 à 10 faire

1 - transformation non linéaire (appliquée indépendamment à chaque octet)

$$M_i \to S(M_i)$$

2 - décalage de lignes : la ligne j, $1 \leq j \leq 4$, est décalée (circulairement) de $j-1$ cases dans le tableau

$$S(M_i) \to \pi(S(M_i))$$

3 - brouillage de colonnes : la colonne j, $1 \leq j \leq 4$, est multipliée par une matrice T dans le corps fini a 2^8 éléments

$$\pi(S(M_i)) \to T \otimes \pi(S(M_i))$$

4 - cadencement de la clé (transformation de la clé à partir de K)

$$K_{i-1} \to K_i$$

et addition de la clé de tour :

$$M_{i+1} := K_i \oplus \big(T \otimes \pi(S(M_i))\big)$$

sortie : message chiffré $C := M_{11}$

Remarque. Les coefficients de la matrice T à l'étape 3 de l'algorithme sont 1, 2 ou 3 et ont été choisis pour la facilité d'implémentation de l'algorithme. Dans le corps fini à 2^8 éléments, les additions reviennent à des additions de polynômes sur F_2, et les multiplications à des multiplications de polynômes modulo un polynôme irréductible.

2.4 Cryptanalyse

• Différents types d'attaques.

attaque à texte chiffré : l'analyste dispose seulement d'un ou plusieurs texte chiffré. C'est bien entendu le cas le plus difficile.

attaque à clair connu : l'analyste dispose de textes en clair, et des textes chiffrés correspondant. En effet en pratique, le cryptanalyste peut avoir eu l'occasion de se procurer de telles informations. Il faut bien avoir conscience du fait que la cryptanalyse touche à l'espionnage, qu'il soit industriel, militaire ou politique...

attaque à clair choisi : l'analyste peut connaître pour n'importe quel message le message chiffré correspondant. Actuellement, les algorithmes de cryptage sont censés résister à ce type d'attaque.

• Différentes méthodes employées.

méthode brutale : pour casser un cryptosystème on teste toutes les clés possibles. C'est un procédé parfaitement efficace, c'est ainsi qu'a été cassé le DES, mais les algorithmes actuels demanderaient un temps beaucoup trop long pour être cassés de cette manière (par exemple avec une seule opération par clé de 128 bits, il faut $2^{128} \simeq 10^{40}$ opérations : avec un million de stations travaillant à un milliard d'opérations par seconde, soit 10^{15} opérations par seconde, il faut 10^{25} secondes, soit un milliard de milliards d'années environ).

techniques non scientifiques : à l'autre extrême, les techniques de "cryptanalyse" reconnues comme étant les plus efficaces (par exemple par un ex chef des services secrets français) sont la corruption, le chantage, et la torture...

Entre les deux, se trouvent des technique scientifiques reposant essentiellement sur des *probabilités* et *statistiques*.

• Cryptanalyse fondée sur les probabilités.

La cryptanalyse utilise des analyses statistiques de l'ensemble des messages codés obtenus à partir d'un grand nombre de messages clairs (un exemple simple est donné dans le code de Vigenère précédent). En bref si les messages codés ne sont pas aléatoires, c'est qu'ils contiennent de l'information sur les messages clairs et sur la clé utilisée. Ceci se formalise mathématiquement en *théorie de l'information* développée par Shannon, en considérant les messages comme des variables aléatoires et en mesurant l'*entropie* (ou information moyenne) des messages clairs connaissant leur code.

Ces principes généraux montrent que l'on peut théoriquement casser n'importe quel système de chiffrement si celui-ci n'est pas parfait. Mais cela ne présage pas de la quantité de temps pour le faire.

Les cryptanalystes ont donc développé des techniques variées pour accélérer la découverte d'informations sur les messages clairs à partir des messages codés : *cryptanalyse linéaire, cryptanalyse différentielle...*

Il en ressort l'importance pour le cryptographe d'être capable de donner aux messages codés un aspect "le plus aléatoire possible" (l'idéal étant un message chiffré aléatoire du type Vernam, malheureusement impraticable). C'est pourquoi les algorithmes de chiffrements cherchent à brouiller le plus possible les données (confusion, dispersion, diffusion...).

D'autre part, les algorithmes précédents codent un message de taille fixée. Pour coder un message long, il faut le découper en message courts, et afin de limiter les attaques statistiques, il faut faire interagir les sous-messages entre eux par des procédé de chaînage.

Ce type de procédé est aussi utilisé pour générer des suites pseudo-aléatoires, ce dont on a besoin ne serait-ce que pour fabriquer des clés elles-aussi "les plus aléatoires possible".

2.5 Chaînage

Etant donné un cryptosystème quelconque (E, D) par blocs de taille n, le but est de crypter un message m long.

• Méthode générale :
On découpe m en blocs de taille n :

$$m = m_1 m_2 ... m_k$$

puis on *chaîne* les segments en les faisant interagir, en clair ou cryptés, pour augmenter la diffusion. Le cryptage se fait en combinant le cryptosystème (E, D) et des opérations XOR.

Il existe plusieurs modes de chaînage, principalement les quatre suivants.

• ECB ("electronic code book")
C'est l'absence de chaînage.

Cryptage :

$$c_i = E(m_i)$$

$$c = c_1 ... c_k$$

Décryptage :

$$m_i = D(c_i)$$

Ce chaînage expose à des attaques statistiques, car deux blocs égaux de la source donnent deux blocs chiffrés égaux.
Remarque : ce chaînage peut-être vu comme une substitution, l'alphabet étant formé des blocs de taille n.

• CBC ("cipher block chaining")

Bloc d'initialisation : c_0.
Cryptage :

$$c_i = E(m_i \oplus c_{i-1})$$

pour $i > 0$ et on envoie c_0 et les c_i.
Décryptage :

$$m_i = c_{i-1} \oplus D(c_i)$$

Ce mode de chaînage est le plus populaire. Comme il repose sur E et D, il s'utilise aussi bien à partir d'un algorithme à clé privée qu'à partir d'un algorithme à clé publique.

• CFB ("cipher feedback")

Ce mode de chaînage possède plusieurs variantes. Celui présenté ici n'utilise pas de fonction de déchiffrement, ce qui facilite l'implantation, mais suppose que le cryptosystème soit à clé privée. *La confidentialité des données repose donc uniquement sur la clé de chiffrement de la fonction E.* Il offre néanmoins une grande sécurité.

Bloc d'initialisation : c_0.
Cryptage :

$$c_i = m_i \oplus E(c_{i-1})$$

pour $i > 0$ et on envoie c_0 et les c_i.
Décryptage :

$$m_i = c_i \oplus E(c_{i-1})$$

• OFB ("output feedback")

Ce mode de chaînage possède plusieurs variantes. Celui présenté ici n'utilise pas de fonction de déchiffrement. *Sa confidentialité repose sur la clé de chiffrement de la fonction E.* La fonction de chiffrement E ne s'applique ici que de façon itérée au bloc d'initialisation, et se combine alors au message clair pour donner le message crypté.

Bloc d'initialisation : c_0.
Cryptage :

$$c_i = m_i \oplus E^i(c_0)$$

pour $i > 0$ et on envoie c_0 et les c_i.
Décryptage :

$$m_i = c_i \oplus E^i(c_0)$$

Ce mode de chaînage est souvent utilisé comme générateur de nombres pseudo-aléatoires.

2.6 Générateurs aléatoires et pseudo-aléatoires

• Générateurs aléatoires

Il est impossible de générer une suite de bits aléatoires à l'aide d'un ordinateur, puisque celui-ci fonctionne de façon exclusivement déterministe : on ne peut pas "programmer l'aléatoire". Par contre on peut utiliser des phénomènes physiques chaotiques, c'est à dire imprévisibles. Par exemple on peut utiliser les datations horaires des entrées-sorties sur un réseau, ou mieux déduire une suite de bits du "bruit" d'un récepteur radio, ou bien de phénomènes quantiques.

• Complexité de Kolmogorov

Etant donné que, par le hasard, n'importe quelle suite peut-être obtenue, il n'est pas du tout évident d'apprécier si une suite aléatoire est suffisamment complexe pour être difficile à retrouver. Par exemple la suite constituée de mille 1 à la suite a très bien pu être tirée à pile ou face sans tricher. Malheureusement cette suite n'est pas satisfaisante pour être utilisée en cryptographie car elle est trop simple à construire, elle est prévisible.

On a été ainsi amené à définir rigoureusement une évaluation du caractère "être plus aléatoire" d'une suite de bits. C'est la *complexité de Kolmogorov*, qui mesure en fait la complexité nécessaire à un algorithme déterministe pour générer la suite en question. Par exemple il est plus facile de réaliser un algorithme dont la sortie est une suite de mille 1, que tout autre suite quelconque de mille bits !

• Générateurs pseudo-aléatoires

Le problème est alors de trouver des moyens de générer des suites "le plus aléatoire possible", c'est à dire ayant une complexité de Kolmogorov la plus grande possible. Sans approfondir ici, on dira simplement que des processus comme les *chaînages avec rétroaction* (feedback) précédents fournissent de bons moyens d'obtenir de telles suites. Une propriété que l'on retrouve souvent dans l'étude mathématique des systèmes dynamiques est que l'itération d'une transformation simple peut avoir des comportements imprévisibles (par exemple l'itération E^i dans le mode OFB précédent). Ici répéter un processus d'un type précédent à partir d'une seule suite quelconque assez courte, peut donner finalement une suite longue ayant un caractère pseudo-aléatoire très fort. Voir [Z] pour plus d'informations.

3 Cryptographie à clé publique

Le principe de la cryptographie à clé publique est d'utiliser les limites de la technologie. En bref, la fonction de chiffrement est connue de tous, facile à programmer, mais l'inversion de cette fonction est impossible à calculer en temps raisonnable avec les techniques informatiques actuelles, sauf pour le destinataire qui a des informations privées supplémentaires sur cette fonction. Ce type de fonction sert aussi à établir des protocoles efficaces et surprenants (d'identification par exemple, voir partie 4.1).

Avantages d'un système à clé publique :

- il n'y a plus de danger que le code soit connu : il est public ! C'est pourquoi c'est le seul type de codage qui puisse fonctionner lorsqu'il y a des millions d'utilisateurs.
- il est possible de "signer" un message de telle sorte que l'on soit sûr de sa provenance.

Inconvénient : les algorithmes sont lents comparés aux algorithmes à clé secrète.

fonction à sens unique : il est facile de calculer $y = f(x)$ avec x mais 'impossible' (trop difficile en pratique) de trouver x avec y.

fonction à sens unique avec brèches, ou trappes, secrètes ("trapdoor") : idem, sauf que connaître certaines informations (trappes) permet de retrouver x facilement.

Cette partie repose beaucoup sur la *théorie des nombres*, les rappels mathématiques sont faits au fur et à mesure.

3.1 Une fonction à sens unique : l'exponentielle discrète

Théorème. *Soit p un nombre premier. L'ensemble $F_p \simeq \{0, ..., p-1\}$ des congruences modulo p muni de l'addition modulo p, et de la multiplication modulo p est un corps.*

Théorème. *Le groupe multiplicatif d'un corps fini est cyclique.*

Définition. Soit p est un nombre premier. Une *racine primitive modulo p* est un entier α générateur du groupe cyclique F_p^*.

Autrement dit si p est premier et α est une racine primitive modulo p, alors

$$F_p^* = F_p \setminus \{0\} = \{\alpha^k[p] \mid k = 1...p-1\}$$

Autrement dit si p est premier et α est une racine primitive modulo p, alors l'application f, appelée exponentielle discrète,

$$F_p^* \to F_p^*$$
$$x \mapsto \alpha^x$$

est bijective.

Pour p assez grand (plus d'une centaine de chiffres), cette application peut-être considérée comme une fonction à sens unique. En effet il est facile de calculer l'image de $x \in F_p^*$ (voir implémentation de RSA partie 3.5), mais il est très coûteux avec les moyens technologiques actuels de calculer le *logarithme discret* $f^{-1}(y)$ de $y \in F_p^*$.

Même si p n'est pas premier, l'exponentielle n'étant alors plus forcément bijective, un logarithme discret reste très difficile (impossible en pratique) à calculer.

Du point de vue de la complexité, on sait juste à l'heure actuelle que le problème de calculer un logarithme discret est équivalent au problème de décision $LOGD$ associé (décider si un logarithme discret est plus grand qu'une certaine valeur), et que celui-ci vérifie([5]) :

$$LOGD \in NP \cap co - NP$$

3.2 Protocole de Diffie-Hellman

En 1976, Diffie et Hellman ouvraient la voie à la cryptographie moderne, fondée sur les fonctions à sens unique, en résolvant un problème de partage secret considéré jusqu'alors comme insoluble.

Aristide et Barnabé veulent se mettre d'accord sur un nombre secret en utilisant un canal non sécurisé : n'importe qui peut écouter toute leur conversation, mais ne doit pas pouvoir déduire le nombre secret.

1 - A et B se mettent d'accord publiquement sur un grand nombre premier p, et une racine primitive modulo p notée α.
2 - A choisit secrètement et aléatoirement un nombre a qu'il gardera pour lui seul. Puis A transmet publiquement à B le nombre $\alpha^a[p]$.
3 - B choisit de même un nombre b et transmet publiquement à A le nombre $\alpha^b[p]$.
4 - Le nombre secret commun est $s = \alpha^{ab}[p]$.

A accède au nombre s en élevant le nombre α^b à la puissance a, et B accède au nombre s en élevant le nombre α^a à la puissance b. Mais Clotaire qui écoutait la conversation ne peut pas pratiquement déduire s car il lui faudrait déduire a ou b et donc résoudre un logarithme discret.

3.3 Une fonction à sens unique avec trappe

Les résultats très classiques d'arithmétique suivants s'assemblent pour former le théorème final, ingrédient de base de l'algorithme RSA qui est le système de chiffrement à clé publique le plus utilisé actuellement à travers le monde.

Définition. *Fonction d'Euler* ϕ : on note $\phi(n)$ le nombre d'entiers inférieurs à n premiers avec n.

Propriété. *Si p et q sont premiers, alors* $\phi(pq) = (p-1)(q-1)$

Théorème d'Euler. *Si m est premier avec n alors* $m^{\phi(n)} \equiv 1[n]$

Théorème de Bezout. *Soient a et b deux entiers naturels et d le plus grand diviseur commun de a et b, alors il existe x et y entiers relatifs tels que*

$$ax + by = d.$$

([5]) voir partie 3.7 pour quelques rappels sur la complexité algorithmique

Corollaire. *Si a et b sont premiers entre eux alors il existe x tel que $ax = 1[b]$.*

Preuve : le plus grand diviseur commun de a et b est 1, donc il existe x et y tels que $ax + by = 1$, donc il existe x tel que $ax = 1[b]$. □

Théorème chinois. *Si $p_1, ..., p_k$ sont premiers entre eux deux à deux, et $a_1, ..., a_k$ sont des entiers quelconques alors le système d'équations*

$$x = a_i[p_i]$$

pour $1 \leq i \leq k$ a une unique solution modulo $p_1...p_k$

Théorème pour RSA. *Soient $p \neq q$ deux nombres premiers, $n = pq$, et soit e un entier premier à $(p-1)(q-1)$ alors il existe d tel que $ed \equiv 1[(p-1)(q-1)]$ et $m^{ed} \equiv m[n]$ pour tout entier $m < n$.*

Preuve : L'existence de d est due au corollaire du théorème de Bezout. On a $\phi(n) = (p-1)(q-1)$ puisque p et q sont premiers. On note k un entier tel que $ed = k\phi(n) + 1$. Si m est premier avec n, alors $m^{ed} \equiv m[n]$ car $m^{ed} \equiv m^{k\phi(n)+1} \equiv m^{k\phi(n)}.m = (m^{\phi}(n))^k.m \equiv 1.m \equiv m[n]$. Si m n'est pas premier avec n, alors m n'est pas premier avec p, ou bien m n'est pas premier avec q. On suppose par exemple que $m = 0[p]$, on a alors m premier avec q. Puisque $\phi(q) = q - 1$, on a $m^{q-1} = 1[q]$ (théorème d'Euler). Donc $m^{\phi(n)} = 1[q]$ donc $m^{ed} = 1[q]$ comme ci dessus $m^{ed} \equiv m^{k\phi(n)+1} \equiv (m^{\phi}(n))^k.m \equiv m[q]$. D'autre part $m^{ed} = 0[p]$. Donc d'après le théorème chinois avec $p_1 = p$ et $p_2 = q$ on doit avoir $m^{ed} = m[pq]$. □

Conclusion. Comme il est pratiquement impossible de trouver d connaissant n et e, puisque cela revient à factoriser n, ni même en connaissant en plus m et m^e puisque cela revient à inverser une exponentielle discrète, ce théorème est un formidable ingrédient mathématique pour la cryptographie (voir partie 3.7). Finalement, si on ne connaît que m^e, n et e, on ne peut pas calculer facilement m : pour des nombres n et e assez grands, la fonction utilisée est pratiquement à sens unique. Mais si on connaît aussi d, il suffit de calculer m^{ed} pour retrouver m : le nombre d joue le rôle de *trappe*.

En pratique, e sera la clé publique servant à chiffrer m, et d sera la clé privée servant à déchiffrer m^e.

3.4 L'algorithme de chiffrement RSA

L'algorithme RSA a été inventé en 1978 par Rivest, Shamir, et Adleman. C'est l'algorithme de chiffrement asymétrique le plus employé aujourd'hui (Internet, cartes bleues...).

Comme vu ci-dessus, l'ingrédient de base est la théorie des nombres, plus particulièrement l'arithmétique. La méthode fonctionne parce que la théorie et la pratique sont très différentes en théorie des nombres :

- il est difficile pour un ordinateur de factoriser un grand nombre,
- il est facile de construire de grands nombres premiers,
- il est facile de décider si un grand nombre est premier.

Le cryptosystème RSA est défini comme suit.

Préparation des clés :
1 - Le destinataire Barnabé choisit deux grands nombres premiers p et q et calcule $n = pq$.
2 - Il choisit e un entier premier à $(p-1)(q-1)$.
3 - Il calcule d tel que $ed \equiv 1[(p-1)(q-1)]$.
(n, e) est la *clé publique*
d est la *clé secrète*

Cryptage. Aristide crypte un message $m < n$ pour Barnabé avec la clé publique de Barnabé :

$$c = E_e(m) \equiv m^e[n]$$

et envoie c à Barnabé.

Décryptage. Barnabé reçoit c et décrypte grâce à sa clé secrète

$$m = D_d(c) \equiv c^d[n]$$

Comme $c^d \equiv m^{ed} \equiv m$ il obtient le texte en clair.

Remarque. En pratique p et q ont plus de 100 chiffres. Les nombres p et q sont oubliés, ils constituent aussi des trappes puisqu'ils permettent de retrouver d.

3.5 Implémentation de RSA

• Génération de grands nombres premiers.

Pour préparer le cryptosystème, on a besoin d'abord de générer deux grands nombres premiers p et q. Ensuite pour choisir e premier à $(p-1)(q-1)$ il suffit de choisir e premier, inférieur à $(p-1)(q-1)$ et non diviseur de $(p-1)(q-1)$.

Il est facile de construire de grands nombres premiers. Cela vient du théorème suivant qui donne la distribution asymptotique des nombres premiers.

Théorème (Hadamard et de la Vallée Poussin 1896). *Soit* $\pi(N) = \#\{p \leq N \mid p \text{ premier}\}$. *Alors pour* N *tendant vers* ∞ *on a*

$$\pi(N) \sim \frac{N}{ln(N)}$$

Pour construire un nombre premier de 100 chiffres, on génère au hasard des nombres entiers impairs de 100 chiffres et on teste s'ils sont premiers. Le théorème précédent assure qu'après en moyenne 125 essais on devrait obtenir un nombre premier.

Remarque : tester des nombres successifs un par un jusqu'a en trouver un premier n'est pas une bonne méthode car il existe des suites arbitrairement longues d'entiers consécutifs qui ne sont pas premiers (exemple : $n! + 2, n! + 3, ..., n! + n$).

Il reste à savoir comment tester rapidement si un nombre est premier.

• Tests de primalité.

Un résultat très récent de 2002 affirme que

$$\text{PRIMALITÉ} \in P$$

en proposant un *algorithme polynomial pour tester la primalité d'un nombre.* On ne détaillera pas ici cet algorithme (voir [AKS]). Il est encore trop tôt pour savoir si cet algorithme va effectivement améliorer en pratique la vitesse du test de primalité par rapport aux algorithmes habituellement utilisés, non polynomiaux mais efficaces.

Partant d'une liste aléatoire de grands nombres, pour accélérer encore le choix d'un nombre premier parmi cette liste, on peut commencer par faire des *tests probabilistes de primalité.* Il en existe plusieurs, qui utilisent, comme les tests de primalité en général, des propriétés arithmétiques pointues des nombres premiers (propriétés de symboles de Jacobi par exemple), voir [Z], [JM], [M] ou [Sa]. On obtient alors une liste de nombres très probablement premiers. Pour établir un cryptosystème RSA, on a besoin d'être sûr que le nombre est premier, et on applique alors un algorithme déterministe de test.

• Théorème de Bezout

Cet algorithme sert à calculer d dans la préparation du cryptosystème.

Entrée : (a, b) entiers positifs
Sortie : (d, x, y) tels que $d = pgcd(a, b)$ et $ax + by = d$
1 - si $b = 0$ alors $(d, x, y) := (a, 1, 0)$
2 - $(x, y) := (1, 0)$, $(x', y') := (0, 1)$
3 - tant que $b > 0$ faire $q := \lfloor a/b \rfloor$; $r := a - bq$: $(x'', y'') := (x - qx', y - qy')$; $(a, b) := (b, r)$; $(x, y) := (x', y')$; $(x', y') := (x'', y'')$;
fin : $(d, x, y) := (a, x, y)$

Le temps de calcul est $O(max(\| a \|, \| b \|)^2)$ où $\| a \|$ est le nombre de bits de a.

• Exponentiation modulo n par algorithme dichotomique

Cet algorithme sert à calculer m^e au cryptage et c^d au décryptage.

Entrée : $a < n$ et k
Sortie : $a^k[n]$
1 - $b := 1$; $y := a$;
2 - tant que $k \neq 0$ faire si k impair alors $b \equiv y * b[n]$; $k := \lfloor k/2 \rfloor$; $y := y^2[n]$;
fin : $b = a^k[n]$

Le temps de calcul $O(\| n \|^3)$.

3.6 Une signature avec RSA

Un problème dérivé du cryptage de message est celui de l'*identification*. Il s'agit cette fois pour l'émetteur d'envoyer un message en prouvant son identité. On peut résoudre ce type de problème à l'aide des fonctions à sens unique.

L'exemple suivant est fondé sur le système RSA, simplement en inversant les rôles.

Le système RSA de B a pour clé privée d, et clé publique (n, e). 1 - B prépare un message m pour A et envoie à A le message m^d. 2 - A calcule $(m^d)^e = m[n]$, et vérifie ainsi que c'est B qui a envoyé le message, puisqu'il est le seul à avoir pu utiliser d pour crypter m.

Remarque. Il existe des procédés variés de signature. Une norme américaine très répandue depuis 1994 est le **DSS** (digital standard signature), qui est une amélioration de la signature d'El Gamal fondée sur l'exponentielle discrète et le théorème de Bezout [Z].

3.7 Cryptanalyse et attaques

Pour casser le système RSA de façon théorique, c'est à dire pour trouver d en connaisant n et e, il faudrait être capable de trouver d'abord p et q, c'est à dire de factoriser assez rapidement l'entier n. Comme on va le voir de suite, ce problème est d'une grande complexité algorithmique, et actuellement hors de portée en général.

Pour casser un système RSA avec une attaque à clair connu, c'est à dire connaissant un message clair m et son chiffrement m^e, il faudrait déduire la clé privée d sachant qu'elle vérifie $(m^e)^d = m$, c'est à dire qu'il faudrait calculer un logarithme discret, ce qui est aussi pratiquement impossible (voir partie 3.1).

Cependant des techniques variées sont développées permettant des cryptanalyses dans certains cas particuliers. D'autre part, comme on verra à la fin, une autre façon d'attaquer un cryptosystème est sur le plan pratique, en attaquant le protocole, c'est à dire en détournant la procédure de communication.

• Factorisation d'entiers et complexité algorithmique.

Un problème de décision associé à la factorisation des entiers est le suivant.

FACTORISATION : étant donné un entier N et un entier $M \leq N$, existe t'il un diviseur de N inférieur à M ?

On montre facilement que résoudre ce problème est équivalent, à un temps polynomial près, à factoriser N (par dichotomies successives). Du point de vue de la complexité([6]),

([6]) Les classes de complexité algorithmique sont définies rigoureusement à l'aide de *machines de Turing*, qui modélisent précisément ce que l'on appelle ici *algorithmes (déterministes)*. Grossièrement :

- le *temps* ou *coût* d'un algorithme (déterministe) est le nombre d'opérations élémentaires utilisées pour aller de l'entrée à la sortie.

- un *problème de décision* est une question dont la réponse est 'oui' ou 'non' selon les valeurs des données.

- un problème de décision est dans P si il existe un entier k et un algorithme (déterministe) donnant la réponse pour des données de taille n en temps inférieur à n^k.

- un problème de décision est dans NP si il existe un entier k et un algorithme (déterministe) prenant en compte

on sait que

$$FACTORISATION \in NP \cap co-NP$$

En effet, un certificat pour répondre 'oui' est un entier diviseur de N inférieur à M, et un certificat pour répondre 'non' est la factorisation de N en facteurs premiers avec pour chaque facteur un certificat de primalité, obtenu en temps polynomial.

En pratique on ne connaît pas d'algorithme polynomial pour résoudre ce problème, c'est à dire pas d'algorithme suffisamment rapide pour permettre une cryptanalyse de RSA. Cependant c'est une question ouverte, et donc assez problématique pour la cryptographie : rien ne garantit que l'on ne trouvera pas un jour un algorithme ingénieux et rapide pour casser RSA.

Pour dire à quel point les questions de ce type sont ouvertes, au delà de la cryptographie, la conjecture fondamentale de l'informatique théorique (un million de dollars a été promis à celui qui la résoudra) est

$$P = NP \ ?$$

• Attaques théoriques.

De nombreuses recherches sont faites pour trouver des techniques de résolution efficaces des problèmes comme la factorisation d'entiers ou le logarithme discret (qui sont liés pour la cryptographie, par exemple quelqu'un qui aurait réussi à intercepter m et m^e dans un système RSA pourrait vouloir trouver d par un logarithme discret). Les méthodes utilisent la théorie des nombres de façon très élaborées (voir [Z] par exemple, ou [JM][M]) : fractions continues, cribles quadratiques, courbes elliptiques...

Mentionnons qu'à l'heure actuelle, tous les algorithmes performants de factorisation suivent une démarche probabiliste, c'est à dire qu'ils cherchent à minimiser le temps moyen de calcul des facteurs d'un nombre n mais qu'ils peuvent éventuellement tourner beaucoup plus lentement si l'on est malchanceux.

D'autre part, si le problème général de la factorisation est très difficile, il y a néanmoins des résultats théoriques permettant des attaques efficaces dans des cas particuliers. Par exemple, en 1990, Wiener a montré que si la clé privée d du système RSA était inférieure à la racine quatrième de la clé publique e, alors on pouvait retrouver la clé privée d assez facilement par le développement en fractions continues de e/n [W].

---

les données d'entrées et des "données supplémentaires", appelées *certificats*, tels que lorsque la réponse est "oui" pour des données d'entrée de taille n, il existe un certificat pour ces données pour lequel l'algorithme donne la réponse en un temps inférieur à n^k.

- un problème de décision est dans $co-NP$ si le problème de décision complémentaire est dans NP (i.e. remplacer 'oui' par 'non' dans la définition ci-dessus).

Les algorithmes des problèmes NP ou $co-NP$ sont *non-déterministes* : ils ne calculent pas une solution, ils *vérifient* qu'une réponse est correcte grâce à une "donnée supplémentaire" qui ne fait pas partie des données d'entrées, que l'on a "deviné" et qui permet l'exécution en temps polynomial.

Un problème de décision dans $NP \cap co-NP$ est dit *bien caractérisé* car il possède un certificat polynomial en cas de réponse 'oui' **et** en cas de réponse 'non'.

Tout ce qui est polynomial est "faisable". Au contraire, pour évaluer le temps réellement nécessaire à la résolution d'un problème dans $NP \cap co-NP$, il faut prendre en compte aussi les "données supplémentaires", sachant qu'on ne peut pas en général deviner la valeur du certificat qui va convenir. Et de fait, l'algorithme employé nécessite en général un temps largement hors de portée dès que la taille des données est grande.

Pour l'anecdote... Une tentative précédente de chiffrement à clé publique était d'utiliser le problème du sac à dos dont on savait qu'il était $NP-$complet (étant donnés des entiers $n_1, ...n_k$ et N, quels entiers choisir parmi les n_i pour que leur somme fasse N). Mais il fallait choisir une clé pour laquelle le problème était suffisamment facile, et le problème devenait trop facile pour ces clés faciles, comme l'a montré Shamir avant d'inventer RSA avec ses collègues. Il faut donc se méfier des problèmes NP, car ils contiennent parfois des cas particuliers trop simples...

Les attaques théoriques étant très difficiles, il reste la possibilité d'attaquer directement le protocole.

• Une attaque sur le protocole : l'attaque par le milieu.

L'attaque très classique suivante porte sur la communication de clés. Elle s'adapte à de nombreux cryptosystèmes et est connue sous le nom d'*attaque par le milieu* (ou "man in the middle").

1 - Aristide demande la clé publique à Barnabé
2 - Barnabé envoie e à Aristide
3 - Clotaire intercepte le message et envoie sa clé e'
4 - Aristide crypte avec e'
5 - Clotaire intercepte le message c' d'Aristide et décrypte avec sa clé secrète
6 - Clotaire change le message d'Aristide et le crypte avec e

Moralité : Aristide doit être sûr que e est bien la clé de Barnabé.

Cette attaque fait ressortir la nécessité pour les interlocuteurs de pouvoir s'identifier. Pour ce faire, on peut imaginer que les interlocuteurs emploient un procédé de signature comme celui mentionné plus haut (partie 3.6) dérivé de RSA. Mais cela suppose encore qu'il n'y ait pas d'usurpation de la clé publique, sans quoi une attaque par le milieu est toujours possible. C'est pourquoi d'une part les clés publiques doivent être largement diffusées, pour que l'on puisse vérifier leur authenticité. D'autre part il faut prévoir des possibilités de vérifier de façon fiable l'authenticité d'une information. Et aussi il faudrait être capable d'effectuer des échanges d'informations pour lesquels des intrusions naïves comme dans l'attaque précédente sont impossibles. Tout ceci est possible grâce aux fonctions à sens unique, et nous amène à l'étude des protocoles et des certificats, thèmes très importants dans la mise en pratique de la cryptographie. En particulier, un moyen de contrecarrer l'attaque précédente est donné par le protocole de "preuve sans transfert de connaissance" de la partie 4.1 suivante.

4 La cryptographie en pratique

En pratique, la cryptographie est utilisée pour permettre des échanges sécurisés d'informations et des vérifications d'authenticité de messages. Comme on vient de le voir dans l'exemple d'attaque précédent, cela suppose, en plus de la conception d'algorithmes de chiffrements théoriquement efficaces, la mise en place de protocoles précis, et de techniques de signatures élaborés afin d'empêcher toute tentative de fraude.

On va présenter ici quelques protocoles cryptographiques permettant différents types d'échange d'information, et la notion de certificat couramment répandue aujourd'hui permettant de s'assurer de l'authenticité d'un message par l'intermédiaire d'un tiers de confiance.

4.1 Protocoles cryptographiques

Un *protocole* est un ensemble de règles permettant à deux machines (ou plus) qui communiquent de réaliser quelque chose. On peut le formaliser à l'aide de deux machines de Turing, ce qui donne naissance à une nouvelle classe de complexité **IP** (classe des problèmes de décision pour lesquels il existe un *protocole interactif*), que l'on n'étudiera pas ici.

En pratique, les protocoles utilisent essentiellement des fonctions à sens unique pour permettre à deux interlocuteurs d'échanger de l'information secrètement en utilisant un canal public. Il y a de très nombreuses variantes suivant le type d'échange qui doit être fait, il peut s'agir de passer un accord, de donner une information parmi plusieurs sans savoir laquelle, etc. On a déjà vu dans la partie précédente le protocole précurseur de Diffie-Hellman permettant de se mettre d'accord sur un nombre secret. On va voir ici 3 protocoles permettant d'autres types d'échanges (voir [Z] pour plus d'informations).

• Jouer à pile ou face par téléphone

Le protocole suivant est présenté de façon ludique, mais est significatif de la puissance des fonctions à sens unique. Aristide et Barnabé veulent jouer à pile ou face par téléphone, mais veulent être sûrs qu'il n'y aura pas de tricherie : celui qui "lance la pièce" ne doit pas pouvoir mentir sur le résultat obtenu.

On suppose pour cela que l'on dispose d'une fonction à sens unique bijective f de E dans F, et d'une partition $E = E_0 \uplus E_1$.

1 - A choisit un $x \in E$ aléatoire, calcule $y = f(x)$ et communique y à B.
2 - B choisit son bit aléatoire $b \in \{0, 1\}$ et l'annonce à A.
3 - A déclare qui a gagné suivant que $x \in E_b$ ou non. A prouve sa bonne foi en révélant x.
4 - B vérifie la bonne foi de A en vérifiant que $y = f(x)$.

Ce protocole met en évidence la notion d'*engagement* : Aristide s'engage sur x en publiant $f(x)$. Cela ne révèle pas d'information sur x mais force A à ne pas modifier son choix. La plupart des protocoles utilisent cette notion de manière plus ou moins explicite.

• Preuve sans transfert de connaissance

Pour éviter l'attaque par le milieu (cf. partie 3.7), ou bien pour prouver son identité sans avoir à révéler son mot de passe, il faudrait idéalement être capable de prouver son identité, sans rien révéler à un éventuel intercepteur. Il est possible de prouver que l'on possède un nombre secret, sans révéler d'information utile grâce au protocole suivant (voir aussi l'exercice 4).

On suppose que les nombres p premier et α primitif modulo p sont publics. Aristide détient un nombre secret s, et rend public le nombre $I = \alpha^s[p]$. On dit qu'il est *identifié* par I. Aristide veut prouver son identité à Barnabé, c'est à dire qu'il détient le nombre secret s, mais comme à son habitude, Clotaire écoute la conversation.

1- A choisit un $r[p-1]$ aléatoire, calcule $t = \alpha^r[p]$ et le communique à B.
2 - B choisit un bit aléatoire $b \in \{0, 1\}$ et la communique à A.
3 - A donne x à B où :
$x = r[p-1]$ si $b = 0$
$x = r + s[p-1]$ si $b = 1$
4 - B vérifie que $\alpha^x = t[p]$ si $b = 0$, ou que $\alpha^x = It[p]$ si $b = 1$.

Dans ce protocole, A s'est engagé sur r par $t = \alpha^r$. Si Clotaire essaye de se faire passer pour A auprès de B, il peut soit espérer que le tirage aléatoire de b donnera $b = 0$, et alors transmettre $x = r$, mais il ne saura pas quoi transmettre si $b = 1$, soit choisir un r aléatoire et transmettre $t' = \alpha^r / I$ à B en espérant que le tirage soit $b = 1$, mais alors il ne saura pas quoi transmettre si $b = 0$. De fait, C ne peut pas se faire passer pour A de façon certaine car il faudrait pour cela qu'il connaisse un logarithme de α^r et de $I\alpha^r$, c'est à dire qu'il connaisse s.

Dans tous les cas C a une chance sur deux de se faire passer pour A.
En répétant k fois ce protocole, C a une chance sur 2^k de se faire passer pour A.

Pour k assez grand, ce protocole est largement dissuasif, et permet donc une authentification de A sans interception possible. Il faut noter que la liste des k nombres transmis lors du protocole complet sont des nombres aléatoires, et donc *rien n'est révélé sur le nombre secret s*.

• Transfert inconscient

Un protocole proposé par Brassard, Crépeau et Robert en 1986 permet de résoudre le problème suivant : Aristide dispose d'un ensemble de secrets $\{s_1, s_2..., s_m\}$, et est prêt à en donner un à Barnabé. Celui-ci veut en obtenir un mais ne veut pas que A sache quel secret l'intéresse. Un exemple d'utilisation proposé par les auteurs est le suivant : A et B sont des agents secrets, et B ne veut pas que A ajoute comme secret à sa liste "je sais quel secret intéressait l'agent B".

Ceci est possible grâce à un protocole assez compliqué, qui repose sur des notions de théorie des nombres plus élaborées que les précédentes. Il est présenté ici brièvement et de façon simplifiée, afin surtout de donner une idée de la puissance des protocoles reposant sur les fonctions à sens unique.

On suppose que chaque secret est un bit $s_i \in \{0,1\}$, $i \in \{1,...,m\}$. B veut obtenir s_i, ne doit pas pouvoir obtenir un s_j, $j \neq i$, et A ne doit pas savoir combien vaut i.

1 - A choisit deux grands nombres premiers p et q et calcule $n = pq$, puis choisit un entier a, non résidu quadratique modulo n et de symbole de Jacobi égal à 1. Pour chaque $i \in \{1,...,m\}$, A choisit aléatoirement $x_i \neq 0[n]$ et calcule

$$y_i = x_i^2 a^{s_i}[n]$$

A donne à B les entiers n, a et la liste $(y_i)_{i=1...m}$.
2 - B choisit aléatoirement $r \neq 0[n]$ et un bit aléatoire $b \in \{0,1\}$. B souhaite connaître le secret s_i et transmet à A l'entier

$$q = y_i r^2 a^b[n]$$

qui est sa *question*
3 - A dit à B si q est un carré modulo n ou non. Le bit s_i est égal à b si et seulement si q est un carré modulo n

En effet, si $s_i = b$ alors q est évidemment un carré modulo n par définition, et on montre mathématiquement facilement que la réciproque est vraie grâce aux propriétés choisies pour a.

Cependant, le protocole précédent manque de fiabilité (par exemple A peut tricher en utilisant un a qui soit résidu quadratique modulo n, ou bien B peut envoyer $q = y_i y_j r^2 a^b[n]$ et déduire $s_i + s_j$), et en fait il doit et peut être raffiné (voir [Z]).

4.2 Hachage

Le principe du hachage est d'associer un petit message à un message quelconque permettant de vérifier l'authenticité de ce dernier. C'est encore une signature mais cette fois-ci associée au message et non à l'expéditeur, on parle d'*empreinte* associée au message. Ceci revient sensiblement au même dans le principe, mais en pratique on veut un procédé qui brouille au maximum le message d'origine tout en le compactant. On utilise pour cela différentes techniques, qui rappellent le schéma de Feistel ou bien les procédés de chaînages, afin de faire interagir entre eux des parcelles du messages de départ.

fonction de hachage H : m étant un message de taille arbitraire,

$$h = H(m)$$

est de longueur fixe (par exemple 128 bits), tel que
- h est facile à calculer à partir de m,
- il est difficile de trouver un antécédent de h,
- connaissant m, il est difficile de trouver m' tel que $H(m) = H(m')$.

Cette dernière propriété est appelée *résistance à la collision.*

Un premier exemple simple est donné ci-dessous. On donne ensuite la construction plus générale d'une fonction de hachage pour une longueur quelconque, à partir d'une fonction de compression pour une longueur fixe (voir [JM][M]).

• message authentification code **MAC**

On utilise le chaînage CBC et une fonction de chiffrement E pour des messages de k bits.

le message $m = m_1...m_t$ décomposé en messages de longueur k.

$$H_1 = E(m_1)$$

pour i variant de 2 à t faire

$$H_i = E(H_{i-1} \oplus m_i)$$

• construction d'une fonction de compression

A partir d'une fonction de chiffrement d'un message de n bits à clé secrète de n bits

$$e : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$$

on peut construire une fonction de compression

$$g : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$$

soit directement si elle est résistante aux collisions, soit en la "perturbant" un peu, par exemple de l'une des façons suivantes

$$g(k,x) = e(k,x) \oplus x$$

$$g(k,x) = e(k,x) \oplus x \oplus k$$

$$g(k,x) = e(k, x \oplus k) \oplus x$$

$$g(k,x) = e(k, x \oplus k) \oplus x \oplus k$$

• construction d'une fonction de hachage

A partir d'une fonction de compression

$$g : \{0,1\}^m \to \{0,1\}^n$$

on peut construire une fonction de hachage

$$H : \{0,1\}^\star \to \{0,1\}^n$$

où $\{0,1\}^\star$ est l'ensemble des mots finis de longueur quelconque sur $\{0,1\}$, de la façon suivante (proposée par Merkle).

Soit $x \in \{0,1\}^\star$ et l sa longueur en binaire.
1 - on complète x avec des 0 en tête pour obtenir $u = 0^i x$ avec $u = 0[m-n]$
2 - on complète l avec des 0 en tête pour obtenir $y = 0^j l$ avec $u = 0[m-n-1]$
3 - on découpe y en blocs de $m-n-1$ bits et on ajoute 1 au début de chaque bloc pour former le mot v
4 - on construit le mot $w = u0^{m-n}v$ composé de t blocs w_i, $1 \leq i \leq t$, de longueur $m-n$
5 - on définit $H(x)$ par induction

$$H_0 = 0^n$$

$$H_i = g(H_{i-1}w_i)$$

pour $1 \leq i \leq t$

$$H(x) = H_t$$

Propriété. *Si g est résistante aux collisions, alors H l'est aussi.*

Les fonctions de hachage les plus répandues sont conçues selon ce principe, par exemple **SHA** (secure hash algorithm) donne un hachage à 160 bits.

4.3 Certificats

Les certificats sont utilisés pour vérifier l'authenticité d'un fichier d'informations (par exemple concernant un logiciel téléchargeable, ou tout autre document), en le soumettant à un organisme extérieur.

autorité de certification ou *tiers de confiance* : organisme qui délivre des certificats.

certificat : petit fichier divisé en deux parties,
- la partie contenant les informations,
- la partie contenant la signature de l'autorité de certification.

Les informations contenues dans le certificat sont selon une norme en vigueur :
- le nom de l'autorité de certification
- le nom du propriétaire du certificat
- la date de validité du certificat
- l'algorithme de chiffrement utilisé
- la clé publique du propriétaire

Délivrance du certificat :
1 - le fichier d'informations est *haché* par l'autorité de certification par un algorithme public, ce qui donne une *empreinte* des informations.
2 - l'autorité de certification *crypte* l'empreinte avec sa clé publique, ce qui donne la *signature* de l'autorité de certification.
3 - le certificat est constitué du fichier d'informations, et de la signature précédente

Pour s'assurer de l'identité d'un interlocuteur, ou de l'authenticité d'un fichier d'informations, si ceux-ci ont été certifiés, on doit soumettre le certificat à l'organisme qui l'a délivré.

Vérification du certificat
1 - le fichier d'informations est *haché* par l'autorité de certification, qui obtient l'*empreinte* du fichier
2 - la *signature* est *décryptée* par l'autorité de certification avec sa clé privée.
3 - l'autorité vérifie qu'elle obtient bien ainsi l'empreinte du fichier d'informations, et répond que le certificat est en règles.

La notion de certificat est l'aboutissement de ce cours d'introduction, car elle repose sur l'ensemble des notions précédentes : chiffrements, protocoles, et cætera. Cependant on est très loin d'avoir fait le tour de la cryptographie, même avec les quelques compléments de la partie suivante...

5 Compléments

5.1 Stéganographie

La *stéganographie* consiste à dissimuler un message (secret) dans un message apparemment anodin. Par exemple les initiales de chaque phrase d'un texte ou de chaque vers d'un poème forment la phrase cachée, ou bien une lettre sur k... C'est un procédé vieux comme le monde, employé souvent en littérature([7]). Ce procédé est aujourd'hui encore employé, et rien ne peut pratiquement empêcher de s'en servir, puisque seul le destinataire est censé savoir qu'un message lui est destiné (ce qui permet de contourner les lois sur la cryptographie, voir partie 5.4).

Par exemple pour estampiller une image (pour la signer, ou bien introduire un copyright, ou un message secret...), on peut modifier le dernier bit de couleur de chaque pixel (un bit sur 24 donne une nuance à la couleur indécelable à l'œil nu) pour qu'il dissimule un message. Plus finement, on peut se limiter aux pixels pour lesquels le contraste sur la photo est fort.

L'inconvénient de ce procédé simple est que le message caché se dégrade lorsque le fichier image est compressé, mais des moyens plus sophistiqués existent.

5.2 Frontières avant-gardistes de la recherche

Parmi les recherches scientifiques d'avant-garde, on peut retenir par exemple un grand espoir pour révolutionner les temps de calculs des algorithmes, qui est *pour l'instant encore à la limite de la science fiction*, mais remettrait en cause toute la cryptographie moderne fondée sur les limitations des capacités technologiques de calcul.

([7]) *Pour l'anecdote...* La sulfureuse correspondance entre la femme écrivain George Sand et Alfred de Musset est un fameux exemple de stéganographie littéraire (trouver le message caché de chaque lettre).

Lettre de George Sand à Alfred de Musset :
Je suis très émue de vous dire que j'ai
bien compris, l'autre jour, que vous aviez
toujours une envie folle de me faire
danser. Je garde un souvenir de votre
baiser et je voudrais que ce soit
là une preuve que je puisse être aimée
par vous. Je suis prête a vous montrer mon
affection toute désintéressée et sans cal-
cul et si vous voulez me voir ainsi
dévoiler, sans aucun artifice, mon âme
toute nue, daignez donc me faire une visite.
Et nous causerons en amis et en chemin
je prouverai que je suis la femme
sincère, capable de vous offrir l'affection
la plus profonde et la plus étroite
amitié, en un mot, la meilleur amie
que vous puissiez rêver. Puisque votre
âme est libre, alors que l'abandon où je
vis est bien long, bien dur, et bien souvent
pénible, ami très cher, j'ai le cœur
gros, accourrez vite et venez me le
faire oublier. A l'amour je veux me sou-
mettre.

Réponse d'Alfred de Musset :
Quand je mets à vos pieds un éternel hommage,
Voulez-vous qu'un instant je change de visage ?
Vous avez capturé les sentiments d'un cœur
Que pour vous adorer forma le créateur.
Je vous chéris, amour, et ma plume en délire
Couche sur le papier ce que je n'ose dire.
Avec soin de mes vers lisez les premiers mots,
Vous saurez quel remède apporter à mes maux.

Et enfin la géniale réponse de George Sand :
Cette insigne faveur que votre cœur réclame
Nuit à ma renommée et répugne à mon âme.

Il s'agit de l'utilisation de la physique quantique : grossièrement, une interprétation des propriétés quantiques de la matière est que lorsqu'un système est livré à lui-même, sans observateur, il évolue dans des univers parallèles, et ne se fige dans la réalité que lorsqu'il est observé à nouveau. Ce principe suggère que la matière est capable d'effectuer des calculs massivement parallèles au niveau quantique, ce qui est très motivant pour les informaticiens. Ces principes ont donné naissance au concept d'ordinateur quantique, qui est cependant plus fictif que théorique pour l'instant. Le problème majeur est que l'on n'est pas encore capable de manipuler suffisamment la matière à ce niveau pour concevoir l'ingénierie nécessaire.

D'un autre côté, les principes de la physique quantique, si on pouvait les domestiquer, permettraient de réaliser un cryptosystème parfaitement incassable. Grossièrement, le principe d'incertitude, qui affirme que l'observation modifie les données, peut-être utilisé de la façon suivante : un message est envoyé de l'émetteur au destinataire, et l'observateur extérieur ne peut percevoir qu'une information faussée. Ceci se traduit plus concrètement en termes de probabilités et d'orientation des spins des particules, constituant le nouvel alphabet pour le message.

Pour plus d'informations, voir [Si].

5.3 Espionnage et tricheries

Etant donnée l'importance des enjeux économiques et politiques autour de la cryptographie, il n'est pas étonnant que de nombreuses techniques d'espionnage et de tricheries quant aux cryptosystèmes se développe. Par exemple il existe aujourd'hui des appareils permettant de détecter la frappe des touches sur un clavier d'ordinateur à distance, permettant de saisir un message avant même qu'il ne soit codé ! Bien entendu l'utilisation de ces outils est interdite aux particuliers, mais il est probable que des organismes plus secrets ne s'en privent pas.

Des scandales variés sont apparus ces dernières années, par exemple une entreprise suisse avait conçu un système de chiffrement vendu à différentes sociétés, mais ce système contenait une trappe cachée vendue discrètement à un état, lui permettant aussi de déchiffrer les messages.

Les fabricants de codes peuvent aussi mentir sur la complexité de la clé. Un exemple naïf est le suivant : plutôt que de choisir une clé aléatoire dans un espace de dimension n, la clé est choisie de façon aléatoire dans un sous espace de dimension $n' < n$. La tricherie peut passer inaperçu si l'on fait des tests statistiques simples sur l'ensemble des clés aléatoires.

A titre d'exemple de ce type d'escroquerie, voici un article du Monde qui en dit long sur *comment on espionne (vicieusement) des messages privés, en trompant sur la force de la clef de cryptage.*

Le Monde du 20/04/98 :

> Outre-Atlantique, une publicité pour le téléphone cellulaire montrait récemment un mouton et un "mobile", assurant qu'on ne pouvait cloner le second. Deux étudiants de l'université de Berkeley ont relevé le défi : ils affirment être parvenus à reproduire la carte à puce SIM - réputée inviolable - que les abonnés glissent dans leur téléphone cellulaire de type GSM. Cette manipulation, rendue publique lundi 13 avril, permettrait de téléphoner aux frais du propriétaire de la carte.
>
> "Craqueurs" de codes dans l'âme, Ian Goldberg et David Wagner ont déjà à leur tableau de chasse le système de, cryptage du logiciel de navigation sur Internet de Netscape. Pour y ajouter la carte SIM, il ont d'abord bénéficié d'une fuite, dont ils taisent la source. Cela leur a permis de reconstituer l'algorithme d'authentification (COMP 128) et d'y trou-ver la faille permettrait de décrypter le code de la carte et de mimer celle-ci.
>
> "C'est un défaut partiel, précise cependant Ian Goldberg. Il faut pouvoir disposer de la carte SIM pour pouvoir la cloner."
>
> (...)Le GSM, qui utilise la technologie numérique assortie d'un système de cryptage des conversations, était considéré comme beaucoup plus sûr. Mais Goldberg et Wagner, en démontant le mécanisme de codage, se sont également aperçus que **la clé de cryptage de 64 bits** (unités élémentaires d'information) **se termine en fait par dix zéros**. "Il s'agit d'un affaiblissement délibéré du système de cryptage, destiné à favoriser les écoutes téléphoniques", accusent-ils, soupçonnant la puissante National Security Agency d'être à l'origine de cette "interférence". Aussi prêchent-ils pour une évaluation publique des systèmes de cryptage, seule garantie, à leur sens, de leur fiabilité. (...)

Enfin, on peut noter que des soupçons sont régulièrement formulés quant à la fiabilité et l'honnêteté des chiffrements utilisés par les logiciels de navigation sur Internet comme Netscape ou Internet Explorer de Microsoft. Le problème est que les codes de ces programmes sont privés, et, vu de la France, qu'ils appartiennent à des pays étrangers. Ainsi s'ajoute toute une série de problèmes juridiques.

5.4 Aspects juridiques de la cryptologie

La cryptographie pose de nombreux problèmes juridiques, d'une part car elle a évolué beaucoup plus vite que les lois, et d'autre part car elle se situe à l'échelle internationale, alors que les lois touchent d'abord des échelles nationales. Elle pose des problèmes aussi variés que la propriété privée des algorithmes de chiffrement, la validité juridique des preuves électroniques (voir http://www.internet-juridique.net/), etc. Pour les réglementations officielles sur la cryptographie on peut consulter http://www.internet.gouv.fr/.

Aujourd'hui, il semble que la France se soit alignée sur d'autres pays européens et les Etats-Unis, en autorisant l'usage de clé de longueur maximale 128 bits à l'usage privé, avec obligation de mettre à disposition les clés à la justice sur sa demande.

Voici pour terminer quelques extraits d'un article un peu ancien mais révélateur sur le vide juridique qui entoure la cryptographie, et les problèmes que cela pose pour les individus et leur liberté.

Par Joanne Baagoe (http://www.censure.org/cryptographie/loi_crypto.htm) :

La réglementation de la cryptologie

Les médias le répètent à satiété : l'Internet n'est pas sûr. Les messages peuvent être interceptés ou même falsifiés. Pourtant, il y a un remède technique imparable : la cryptologie. Mais on se heurte au problème de sa réglementation.

La situation actuelle

[...] Les progrès de la cryptologie permettent désormais à chacun de prendre des mesures qui rendent toute tentative de décryptage parfaitement futile. Sur le plan technique, la partie est définitivement jouée : le bouclier a gagné face au glaive. Il est dorénavant possible de protéger ses secrets de manière inviolable, et tout nouveau progrès de l'informatique ne fera qu'accroître l'avantage du chiffreur sur le décrypteur.

Cela implique que la Mafia, par exemple, peut conserver sa comptabilité et ses fichiers clients et fournisseurs sous une forme que personne, pas même les meilleurs spécialistes des meilleurs services d'Etat, ne peut lire sans la clé. Les trafiquants de drogue peuvent prendre des commandes sans risque. Les terroristes peuvent conspirer par téléphone ou par télématique selon des moyens qui font de toute interception une perte de temps. Et cela, évidemment, n'est pas réjouissant.

Seulement, que faire ? A moins d'interdire l'informatique, on ne peut pas priver les criminels des moyens techniques nécessaires. En fait, n'importe quel boy-scout avec un PC même d'occasion dispose du matériel suffisant. L'Union soviétique avait essayé de réglementer l'accès à l'ordinateur (ainsi qu'à tout moyen de communication, jusqu'aux photocopieuses) ; ce fut un élément décisif de sa perte à cause des conséquences évidentes de marginalisation technologique. Personne ne peut proposer d'aller dans ce sens.

Reste donc à proposer des nouvelles mesures, plus raisonnables, de réglementation de la cryptologie elle-même. C'est le but du projet de loi (http://www.telecom.gouv.fr/francais/activ/telecom/lrt121.htm) modifiant l'article 28 de la loi du 29 décembre 1990 sur les télécommunications, adopté par le Conseil des ministres du 3 avril 1996, modifié et adopté par l'Assemblée nationale, le 10 mai 1996.

Le projet de loi

Le projet rend enfin libre l'usage de la cryptologie dans un certain nombre de cas. (Notons que c'est la première fois que le mot "libre" apparaît dans un texte officiel français sur la cryptologie. On imagine la douloureuse révolution culturelle que cela a dû impliquer dans les services concernés.) Cependant, il ne faut pas se faire d'illusions sur la portée pratique de ce changement de vocabulaire : si jadis, la cryptologie était interdite, sauf quand elle était autorisée, maintenant, elle est libre, sauf quand elle est soumise à autorisation. Il n'en reste pas moins qu'il y a comme une amorce de changement d'état d'esprit, ainsi que quelques réels progrès.

La cryptologie sans confidentialité

Liberté, d'abord, pour l'usage de la cryptologie à des fins d'authentification et de contrôle d'intégrité, mais non de confidentialité : le régime passablement irréaliste qui exigeait théoriquement une déclaration préalable à chaque application d'une fonction de hachage telle que SHA ou MD5 (voire d'un banal checksum !) est supprimé. Désormais, on voit mal ce qui s'opposerait à une vérification par PGP de la signature d'un correspondant étranger - la détention de PGP, comme de tout autre moyen de cryptologie, n'est pas interdite, même si son usage à des fins de confidentialité le reste. (Attention, toutefois, à prendre soin de le télécharger à partir d'un serveur situé dans l'Union européenne si le projet de loi est adopté - sinon, vous risquez la prison pour importation !)

Les tiers de confiance

Ensuite, l'usage de la cryptologie pour assurer la confidentialité sera libre si "le moyen ou la prestation assure des fonctions de confidentialité et n'utilise que des conventions secrètes gérées selon les procédures et par un organisme agréé". C'est cette disposition, les fameux "tiers de confiance" qui constitue la principale innovation du projet de loi. [...]Notons que la France est le premier pays au monde à mettre en place ce genre de mécanisme, qui a fait l'objet de très vifs débats, et dont le caractère obligatoire a jusqu'à présent été écarté partout ailleurs. Il s'agit d'organismes (privés ou publics - en pratique, on semble s'orienter vers, par exemple, le GIE CB, groupement d'entreprises bancaires de droit privé qui gère les cartes de crédit) agréés par l'Etat, et chargés de gérer les clés secrètes des usagers. Ils sont tenus de conserver les clés, et de les remettre en cas de besoin à la police ou à la justice. En dehors de ce cas, ils sont tenus au secret

professionnel. Si leur principe peut paraître intéressant, il soulève tout de même des interrogations. D'abord, dans quelles modalités exactes ces clés pourront-elles être communiquées, et au juste à qui? [...]

Une réglementation peut-elle être efficace ?

Personne ne peut souhaiter que la cryptologie favorise les activités contraires à la loi.

Cependant, on peut s'interroger sur l'efficacité d'une réglementation, quelle qu'elle soit : il est loin d'être évident que les criminels les plus intelligents et les plus redoutables auront le bon goût de s'y soumettre... On voit mal les espions et terroristes manipulés depuis l'étranger utiliser sagement les mécanismes officiels français. Quant aux trafiquants de drogue et autres grands criminels organisés, ils ne vont pas adopter des procédés autorisés mais peu sûrs de leur point de vue, même s'ils encourent trois ans de prison de plus en en adoptant d'autres - ils risquent déjà la perpétuité, et leur priorité, c'est de ne pas se faire prendre.

En outre, la détection de messages chiffrés clandestins se heurte à des difficultés considérables, à la fois légales et techniques. Certes, on peut supposer que les services secrets français, comme les autres, ne s'embarrassent pas toujours des dispositions de la loi pour espionner ce qui passe sur le réseau. Cependant, il sera problématique d'utiliser de tels renseignements dans un prétoire. Mais surtout, il est techniquement difficile de repérer ce qui, dans le flot des messages, pourrait constituer un message chiffré : sa principale caractéristique (une distribution régulière, en apparence aléatoire) se retrouve également dans les fichiers tout simplement comprimés. De plus, rien n'empêche, dans l'état actuel de la législation, d'envoyer sur les réseaux des fichiers réellement aléatoires, et une interdiction éventuelle se heurterait à de grandes difficultés de définition. Enfin, et sans entrer dans les détails, il est parfaitement possible de dissimuler des messages chiffrés dans des messages d'apparence anodine.

Si donc ce sont les trafiquants de drogue et les terroristes qui servent d'épouvantail pour justifier la réglementation, ce ne sont probablement pas eux qui sont, en fait, visés, mais plutôt les délinquants "ordinaires". [...] En revanche, il ne faut pas s'attendre à ce que la répression de la fourniture de moyens non autorisés diminue le moins du monde leur disponibilité. Il ne faut pas même l'effort nécessaire à obtenir de fausses pièces d'identité ou des armes clandestines pour trouver des moyens de cryptologie sûrs : il suffit de télécharger PGP.

Est-ce que, compte tenu des limites des résultats qu'on peut en attendre, et des inconvénients, des lourdeurs et des complications internationales que cela implique, une réglementation du genre proposé se justifie ? Le mécanisme des tiers de confiance ne sera sûrement pas gratuit - que ce soit l'utilisateur ou le contribuable, quelqu'un devra le payer. Il en résultera forcément un certain obstacle à la diffusion de la cryptologie, ce qui est peut-être le but recherché, mais qui favorise l'espionnage, et pas seulement au profit de l'administration française.

Car il y a des alternatives moins contestables pour lutter contre les abus de la cryptologie, et le projet de loi en amorce une : réprimer, non pas la cryptologie en soi, mais son utilisation à des fins criminelles. [...] On aurait pu dire, tout simplement, que l'usage de la cryptologie en vue de dissimuler un crime ou un délit constitue, de toutes façons, une circonstance fortement aggravante. Et l'on peut réprimer le refus, par l'intéressé lui-même, de fournir ses clés à la demande du juge d'instruction : c'est une entrave à la justice par soustraction de documents, punie de trois ans de prison et de 300 000 F d'amende par l'article 434-4 du nouveau Code pénal - donc, plus sévèrement que ce qui est prévu le manquement à leurs devoirs de la part des tiers de confiance.

Toute nouvelle technique peut être détournée à des fins criminelles. Quand la bande à Bonnot utilisa pour la première fois un véhicule automobile pour prendre la fuite après un vol à main armée, on aurait pu être tenté de limiter l'usage de l'automobile... Dans le cas de la cryptologie, le problème est compliqué par le fait que les spécialistes qui ont l'oreille du gouvernement sont soit des professionnels du renseignement qui ne veulent pas être privés de leurs sources, soit des inventeurs de procédés de chiffrement à destination militaire ou diplomatique qui voient d'un mauvais oeil leur spécialité traditionnellement très discrète s'étaler désormais sur la place publique. Il serait peut-être temps que les pouvoirs publics se rendent compte des dangers que font naître les obstacles à la cryptologie, de la délinquance ordinaire (chantage, détournements de fonds, intrusions sur les ordinateurs, etc.) jusqu'à l'espionnage industriel international. En un mot, qu'ils se mettent un peu moins dans la peau de l'espion pour se mettre un peu plus dans celle de l'espionné.

5.5 Conclusion

Avec les moyens techniques actuels, la cryptographie a "gagné" sur le plan théorique face à la cryptanalyse : on sait concevoir des cryptages impossibles à déchiffrer en un temps raisonnable. Cependant la science avance rapidement et surtout, comme le montrent par exemple les extraits de presse ci-dessus, d'autres paramètres sont à prendre en compte. Pour celui qui veut déchiffrer un code secret, tous les coups sont permis : sans

même parler des techniques violentes ou militaires reconnues comme étant les méthodes de "cryptanalyse" les plus efficaces, il existe aussi des méthodes d'espionnage qui contournent le cryptage, des algorithmes de chiffrement où des trappes sont dissimulées permettant de contourner les clés secrètes et vendues au plus offrant (états, entreprises...), ou bien certaines clés sont "simplifiées" pour permettre à celui qui sait qu'elles le sont de les trouver plus facilement. A cela s'ajoute un cadre juridique encore très mal défini. Par exemple en France il était interdit officiellement jusqu'à récemment de coder ne serait-ce que son propre journal intime, alors que n'importe qui pouvait télécharger le petit programme **PGP** (pretty good privacy) permettant facilement de crypter tous ses messages électroniques... Sur tous les plans, les choses évoluent très vite, et l'importance de la cryptologie ne fait que s'accroître à notre époque de plus en plus moderne !

Références

Les références suivantes sont de bons moyens d'approfondir tout ou partie de ce cours d'introduction. D'ailleurs elles en ont directement inspiré certains passages, et leurs auteurs en sont vivement remerciés. Merci aussi à Serge Burckel pour sa relecture attentive.

Sur Internet :

Eléments de cryptographie par Markus Jaton. *(bon cours d'introduction)*
http://docpacks.tcom.ch/data/Cryptographie/

Tutoriel de cryptographie par Simon Guillem-Lessard. *(nombreux liens Internet)*
http://www.uqtr.ca/ delisle/Crypto/introduction/

Résumés de certaines notions de cryptographie par Jean Berstel. *(cours succins)*
http://www-igm.univ-mlv.fr/%7Eberstel/Cours/Crypto.html

Cours de sécurité informatique du groupe TiOS *(cours technique et approfondi)*
http://cui.unige.ch/tios/cours/securite/2003/cours/

FAQ de crypto en anglais, site officiel de RSA Security *(clair et complet)*
http://www.rsasecurity.com/rsalabs/node.asp?id=2152

[AKS] PRIMES is in P. Agrawal, Kayal, Saxena (2002).

[JM] Cryptographie & Cryptanalyse. Sandrine Julia, Bruno Martin. Cours de DEA de l'Université de Nice (2003).

[M] Codage, Cryptologie et Applications. Bruno Martin. Presses polytechniques et universitaires romandes (2004).

[PLS] La cryptographie - L'art du secret. Dossier Pour La Science 36 (hors série juillet/octobre 2002).

[Sa] Public-Key Cryptography. Arto Salomaa. Texts in theoretical computer science, Springer (1996).

[Si] Histoire des codes secrets. Simon Singh. J.C. Lattès ed.

[W] Cryptanalysis of short RSA secret exponents. Michal Wiener. IEEE Transaction on information theory, 36 (3), (1990).

[Z] Cours de Cryptographie. Gilles Zémor. Editions Cassini (2000).

Exercices

Exercice 1
Les lettres de l'alphabet français sont représentées par les nombres de 1 à 26, dans l'ordre croissant. Pour deux mots de taille n, les opérations $+$ et $-$ désignent respectivement l'addition et la soustraction lettre à lettre modulo 26.
On considère le cryptosystème suivant, pour lequel le message clair m en français est découpé en blocs m_i, $1 \leq i \leq k$, de taille n :

$$m = m_1 ... m_k,$$

le message chiffré c est découpé aussi en blocs c_i, $1 \leq i \leq k$, de taille n :

$$c = c_1 ... c_k,$$

et la *clé de chiffrement* est un mot t de taille n.
La taille n des blocs est supposée connue de tous.

Cryptage. $c_1 = m_1 + t$
et pour $1 < i \leq k$, $c_i = m_i + m_{i-1}$
Déryptage. $m_1 = c_1 - t$
et pour $1 < i \leq k$, $m_i = c_i - m_{i-1}$

Trouver une méthode de cryptanalyse pour ce système, pour un message suffisamment long (i.e. trouver m à partir de c et n sans connaître t).

Exercice 2 (d'après [Z])
Soit $\mathcal{K}$ l'ensemble des mots de n bits. Pour améliorer un cryptosystème f_K dont les clés $K \in \mathcal{K}$ sont trop courtes et n'interdisent pas la recherche exhaustive (on peut penser au DES), on peut penser à appliquer le chiffrement deux fois de suite pour deux clés différentes :

$$g_{K_1,K_2}(m) = f_{K_1}(f_{K_2}(m))$$

pour $K_1, K_2 \in \mathcal{K}$. On double ainsi la longueur de la clé.
a) Montrer qu'ordonner une liste de 2^n mots binaires dans l'ordre lexicographique (i.e. ordre des entiers naturels écrits en base 2) peut se faire en $n2^n$ comparaisons de mots.
b) On sait programmer f_K et f_K^{-1} pour tout $K \in \mathcal{K}$, on connaît un message clair m et son cryptage $c = g_{K_1,K_2}(m)$, et on cherche un ensemble très réduit de couples (K_1', K_2') susceptibles d'être la clé (K_1, K_2).
Montrer que cette *attaque à clair connu* peut-être faite en $O(n2^n)$ opérations élémentaires.
Indication : établir deux listes bien choisies de taille 2^n, les ordonner, et les comparer.
Remarque : lorsque, dans une attaque à clair connu, on connaît un message m et son cryptage c, on peut déduire un ensemble très réduit de mots binaires de n bits susceptibles d'être la clé en calculant toutes les images $f_{K'}(m)$, $K' \in \mathcal{K}$ et en retenant les K' pour lesquels on trouve c. Ceci se fait en $O(2^n)$ opérations élémentaires. En doublant la longueur

de la clé, on espère que le nombre d'opérations élémentaires devienne au mieux $O(2^{2n})$. Avec cet exercice on voit que ce n'est pas le cas pour la construction ci-dessus.
C'est pour cette raison que l'on n'utilise pas de Double-DES, mais un Triple-DES, comme amélioration du DES.

Exercice 3
La clé publique de Barnabé pour son système RSA est $(55, 27)$. Aristide lui envoie un message m (nombre entier), et vous interceptez le message chiffré (nombre entier) correspondant $c = 4$.
Trouvez m.

Exercice 4 ([Z])
Le but du protocole suivant est de réduire le nombre de communications à une seule dans le *protocole sans transfert de connaissance* vu en cours, avec la même efficacité.
A cherche à prouver à B qu'il détient s tel que $\alpha^s = I[p]$, où les nombres I identifiant de A, p premier et α primitif modulo p sont publics.

1 - A choisit k entiers aléatoires $r_1, ..., r_k$, calcule et envoie à B les entiers $t_1 = \alpha^{r_1}[p]$, ..., $t_k = \alpha^{r_k}[p]$.
2 - B choisit aléatoirement et envoie à A un vecteur binaire $\epsilon = (\epsilon_1, ..., \epsilon_k) \in \{0,1\}^k$.
3 - A révèle à B les entiers $x_1, ..., x_k$ où
$x_i = r_i$ pour tout indice i tel que $\epsilon_i = 0$,
$x_j = s + r_j$ si j est le plus petit indice tel que $\epsilon_j = 1$,
$x_i = r_i + r_j$ pour tout indice i tel que $i > j$ et $\epsilon_i = 1$.
4 - B vérifie que $\alpha^{x_j} = It_j$, que $\alpha^{x_i} = t_i$ pour tout i tel que $\epsilon_i = 0$, et que $\alpha^{x_i} = t_i t_j$ pour tout $i > j$ tel que $\epsilon_i = 1$.

a) Vérifier que ce protocole est bien sans transfert de connaissance.
b) Montrer que la probabilité qu'un imposteur réussisse à se faire passer pour A est au plus $1/2^k$.

Corrections

Exercice 1
On a les relations :

$$c_1 = m_1 + t$$

$$c_2 - c_1 = m_2 - t$$

$$c_3 - (c_2 - c_1) = m_3 + t$$

...

De cette façon, à partir de c, on peut déduire par récurence la suite $(m_{2i} - t)_{i\geq 1}$, et la suite $(m_{2i+1} + t)_{i\geq 0}$:

$$m_1 + t = c_1$$

et pour $i \geq 1$

$$m_{2i} - t = c_{2i} - (m_{2i-1} + t)$$

$$m_{2i+1} + t = c_{2i+1} - (m_{2i} - t)$$

Pour $1 \leq j \leq n$, on extrait alors la j-ème lettre de chaque mot de la suite $(m_{2i+1} + t)_{i\geq 0}$. Cet ensemble de lettres a été obtenu en ajoutant la j-ème lettre de t à des lettres de m, qui est un message écrit en français. Il suffit donc de chercher quelle est la lettre la plus fréquente dans cette liste, si le message est suffisamment long, cette lettre doit correspondre à la lettre française e, de loin la plus fréquente. On connait ainsi la j-ème lettre de t. En faisant la même chose pour tous les $1 \leq j \leq n$, on en déduit la clé t.

Exercice 2
a) On fait une preuve par récurence. Si $n = 1$, une seule comparaison suffit. On suppose qu'ordonner 2^{n-1} mots se fait en $(n-1)2^{n-1}$ comparaisons, et on a une liste de 2^n mots. On découpe la liste en deux parties de 2^{n-1} mots, on ordonne chacune des deux, et on obtient les listes ordonnées $l_1 < ... < l_{2^{n-1}}$ et $l'_1 < ... < l'_{2^{n-1}}$. On déduit la liste ordonnée voulue à $2 \times 2^{n-1}$ éléments en parcourant la première liste dans l'ordre, et en insérant les éléments de la seconde liste selon les comparaisons. Plus précisément l'algorithme employé est le suivant qui calcule la liste L.
$j := 1$, $i := 1$, $(L) := \emptyset$
tant que $i < 2^{n-1}$ et $j < 2^{n-1}$ faire
si $l_i < l'_j$ alors $(L) := (L, l_i)$, $i := i + 1$,
sinon $(L) := (L, l'_j)$, $j := j + 1$.
La boucle est effectuée $2 \times 2^{n-1}$ fois, avec une comparaison de mots à chaque fois. On a donc besoin au total pour ordonner la liste de $2\times(n-1)2^{n-1}+2\times 2^{n-1} = n2^n$ comparaisons de mots.

b) On calcule la liste L_1 des $f^{-1}_{K'_1}(c)$ pour $K'_1 \in \mathcal{K}$, et la liste L_2 des $f_{K'_2}(m)$ pour $K'_2 \in \mathcal{K}$. Le mot $f_{K_2}(m)$ est commun aux deux listes par définition de m et c. Il faut donc cherher les élements communs aux deux listes, les couples (K'_1, K'_2) correspondants sont susceptibles

d'être le couple (K_1, K_2). Il y en a au moins un (la clé elle-même), mais il peut y en avoir plusieurs, même si c'est très peu probable.
Le calcul des deux listes se fait en $O(2^n)$ opérations élémentaires (le nombre d'opérations élementaires pour calculer l'image ou l'antécédent d'un mot donné par $f_{K'}$ pour un $K' \in \mathcal{K}$ est connu et ne dépend que de l'algorithme de chiffrement).
Pour trouver l'intersection des deux listes, il sufit d'ordonner la réunion des deux, ce qui se fait d'après la question a) en $O(n2^n)$ opérations élémentaires.
On a donc besoin finalement de $O(n2^n)$ opérations élémentaires.

Exercice 3
Pour le système RSA de Barnabé, $n = pq = 55$ donc $p = 5$ et $q = 11$. La clé publique est $e = 27$, la clé privée d vérifie par définition $de = 1[(p-1)(q-1)]$, c'est à dire $27d = 1[40]$. Sans calculatrice : $2 \times 40 + 1 = 81 = 3 \times 27$. Donc $d = 3$.
On décrypte le message c avec $m = c^d[n]$, soit $m = 4^3[55] = 9$.

Exercice 4
a) Quelqu'un qui suit la communication dispose de la suite $t_1, ..., t_k$ de laquelle il ne peut pas déduire la suite $r_1, ..., r_k$ (logarithme discret), et de la suite $x_1, ..., x_k$ déduite de la suite aléatoire $r_1, ..., r_k$ inconnue. Pour celui qui observe, la suite $x_1, ..., x_k$ est une suite aléatoire car : pour $\epsilon_i = 0$, $x_i = r_i$ est aléatoire, $x_j = s + r_j$ est aléatoire puisque r_j l'est, et pour $\epsilon_i = 1$ avec $i > j$, $x_i = r_i + r_j$ est aléatoire puisque r_i l'est. Rien ne permet de distinguer la suite $x_1, ..., x_k$ d'une suite d'entiers choisie au hasard, donc aucune information exploitable pour trouver s n'a été transmise : le protocole est sans transfert de connaissance.
b) On suppose que C prend la place de A dans la communication et choisit lui-même la suite $t_1, ..., t_k$ transmise à B. C ne connaît pas le logarithme discret s de I. Il suffit de montrer que pour tout i, $1 \leq i \leq k$, C a au plus une chance sur deux de passer le test de vérification correspondant à l'indice i (autrement dit, quel que soit le choix de la suite $t_1, ..., t_k$ que C envoie, il y a au plus un vecteur ϵ qui authentifiera C comme étant A).
A l'étape i, si $\epsilon_i = 0$, C doit donner un logarithme discret de t_i. Si $\epsilon_i = 1$ et si i est le plus petit tel que $\epsilon_i = 1$, C doit donner un logarithme discret de It_i. Si $\epsilon_i = 1$ et si $j \neq i$ est le plus petit tel que $\epsilon_j = 1$, C doit donner un logarithme discret de $t_i t_j$. Pour pouvoir passer le test quelle que soit la valeur de ϵ_i, C doit soit connaître un logarithme de t_i et de It_i, donc de I, ce qui est impossible, soit connaître un logarithme de t_i et de $t_i t_j$, donc de t_j. Mais dans ce cas $\epsilon_j = 1$ et C devait aussi connaître un logarithme de It_j à l'étape $j < i$, il devait donc encore connaître un logarithme de I, ce qui est impossible. Dans tous les cas, C a au plus une chance sur deux de passer le test à l'étape i. Donc C a au plus une chance sur 2^k de passer les k tests.