Cryptographie classique et cryptographie publique à clé révélée

Dany-Jack Mercier

IUFM des Antilles et de la Guyane
Université des Antilles et de la Guyane
Equipe Applications de l'Algèbre et de l'Arithmétique

Septembre 1996

Résumé

A l'ère des communications électroniques, le codage des informations destiné à en assurer la confidentialité est devenu une nécessité. Cet article précise les enjeux actuels de la cryptographie, évoque quelques systèmes classiques comme le D.E.S. puis décrit l'apport décisif de l'arithmétique dans deux systèmes récents.

1 Introduction

L'idée de coder un message dans le but de le rendre inintelligible à toute tierce personne ne date pas d'aujourd'hui. Les « messages secrets » ont joué un rôle important dans tous les conflits depuis que l'homme sait écrire, et sont habituellement associés aux guerres ou aux agents secrets...

Ce qui est nouveau, c'est le besoin quasi vital de coder toute sorte d'information dans notre vie de tous les jours. Ainsi, les cartes bancaires sont devenues nos proches compagnes et utilisent deux systèmes de cryptage : le système standard à clé secrète DES (Data Encryption Standard) et le système RSA dont nous parlerons plus loin.

Dans un autre domaine, les transactions qui s'effectuaient par lettres, par téléphone ou par contact direct entre deux personnes se font de plus en plus par l'intermédiaire de réseaux de communications électroniques. Les micro-ordinateurs sont connectés grâce aux modems et au réseau téléphonique, on s'envoit des fax et l'on utilise la poste électronique. Les informations circulent entre les machines sans qu'il y ait nécessairement intervention humaine, et il est de plus en plus facile d'émettre un message sous une fausse signature ou de lire un message sans que son destinataire ne le soupçonne. Ainsi la

[0] [ccry0001] v 1.02 APMEP **406**, 1996, pp. 568-581.

généralisation de l'emploi de moyens modernes de communication multiplie les possibilités d'indiscrétion et de falsification. Les conséquences de telles falsifications peuvent être graves. Ne pas protéger un message, c'est s'exposer :

- à voir un concurrent connaître les termes exacts d'un contrat signé entre deux partenaires,
- à recevoir des messages sans être parfaitement sûr de leur provenance puisque étant correctement signés,
- à voir son compte en banque débité à la réception d'un ordre de virement électronique émis sous une fausse signature,
- à ce que des documents confidentiels tels des renseignements médicaux, financiers ou commerciaux soient interceptés et utilisés à mauvaises fins,
- à ce qu'un satellite obéisse à un quelconque groupe terroriste et pointe ses missiles sur son pays...

Finalement, qu'est-ce qu'un bon système de cryptographie ? C'est un système qui permet de rendre un message indéchiffrable par toute personne à laquelle il n'est pas destiné tout en pouvant être facilement traduit par le destinataire officiel. Dans la mesure du possible, ce système doit permettre de joindre une signature inviolable au message pour permettre au destinataire de contrôler son authenticité.

2 Structure générale d'un cryptosystème

Les ingrédients d'un cryptosystème sont :

- une fonction de codage (ou « d'encryptage ») C_K,
- une fonction de décodage (ou « de décryptage ») D_K,
- une clé K.

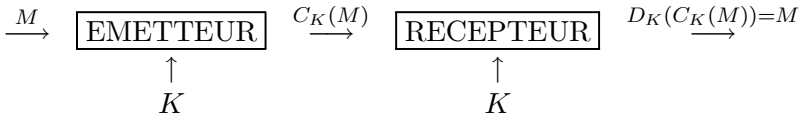
L'émetteur désire envoyer un message M. Il utilise la clé K pour produire le message codé $C_K(M)$. Le récepteur doit ensuite pouvoir à l'aide de D_K et de $C_K(M)$ reconstituer le message M.

En d'autres termes, on doit avoir :

$$D_K(C_K(M)) = M \text{ pour tout message } M.$$

Cette égalité entraîne l'injectivité de C_K. C_K doit associer des textes distincts à des messages distincts !

La connaissance de D_K et de C_K ne doit être possible que si l'on connaît la clé K. Cette clé doit être tenue secrète par l'émetteur et le récepteur. Un tel système exigeant une clé unique est appelé *cryptosystème conventionnel* ou *cryptosystème à clé simple* et peut se schématiser dans la diagramme ci-dessous :



Cryptosystème à clé simple

Nous allons maintenant voir trois exemples de cryptosystèmes conventionnels.

3 Code de César (101-44 avant JC)

Supposons que le message M soit formé avec l'alphabet usuel $\mathcal{A} = \{A, B, ..., Z\}$. Soit σ une permutation de $\mathcal{A}$. Le codage d'une lettre x du message M sera $\sigma(x)$ et son décodage se fera au moyen de σ^{-1}. Avec la permutation cyclique :

$$\sigma = (e, b, h, r, u, q, l, i, x, z, v, f, c, t, a, w, m, n, j, o, g, p, k, y, s, d)$$

on obtient :

Message :	B	O	N	J	O	U	R
Message crypté :	H	G	J	O	G	Q	U

Le cycle σ représente la clé permettant le codage et le décodage du message. Evidemment un tel codage est facile à percer, une méthode possible consistant à calculer les fréquences d'apparition de chaque lettre dans le message codé puis à les comparer à celles, connues, d'un texte écrit en clair.

Ce mode de cryptage est appelé *code de César*. Suétone rapporte en effet que les lettres de César à Cicéron étaient codées en remplaçant chaque lettre par celle située 3 places plus loin dans l'alphabet [2][4].

4 Code de Vigénère (1523-1596)

Dans les codes de César, on traduit chaque lettre du message en une autre lettre grâce à la permutation σ. On peut compliquer ce procédé en traduisant cette fois-ci globalement le message M, formé d'une succession de lettres, à l'aide d'une application injective de l'ensemble des mots *de longueurs quelconques* dans lui-même.

Prenons la permutation σ du paragraphe précédent. Elle est cyclique d'ordre 26 et l'on peut introduire les 26 permutations σ^j, $1 \leq j \leq 26$. Servons-nous des lettres de l'alphabet pour noter chacune de ces permutations : A pour σ^1, B pour σ^2, etc.

Prenons le mot $ACID$ pour clé. On a :

$$A \simeq \sigma^1 \quad C \simeq \sigma^3 \quad I \simeq \sigma^9 \quad D \simeq \sigma^4$$

et le codage :

Message :	B	O	N	J	O	U	R
Clé :	A	C	I	D	A	C	I
Message crypté :	H	K	E	K	G	I	C

Ici la clé est le couple $(\sigma, ACID)$ et les transformations sont du type "addition modulo 26 " dans $\mathbb{Z}/26\mathbb{Z}$. On vient de voir un exemple de *code de Vigénère*, ou *polyalphabétique* par opposition au *code monoalphabétique* de César.

Le code de Vigénère, largement utilisé durant des siècles, est facile à percer surtout si l'on connaît la longueur du mot-clé. Dans l'exemple précédent où le mot-clé $ACID$ comporte 4 lettres, il suffit de calculer les fréquences d'apparition des lettres du message crypté situées aux premières, cinquième, neuvième, ... places puis de les comparer à celles des lettres d'un message en clair [6].

5 Le système D.E.S.

Ce mode de cryptage a été retenu en janvier 1977 par le U.S. National Bureau of Standards pour toutes les organisations fédérales en ce qui concerne la protection des données informatiques. A l'heure actuelle, le Data Encryption Standard (D.E.S.) est présent dans chacune de nos cartes bancaires et permet leur authentification à chacune de leur utilisation.

Ici les mots M sont formés de 64 bits et la clé secrète en comporte 64 dont 56 choisis aléatoirement, les 8 restants servant de bits de contrôle. Cette clé permet de procéder à 16 séries d'opérations et le décryptage consiste à inverser l'ordre de ces opérations.

Il s'agit d'un algorithme de « codage par blocs ». On applique d'abord une permutation p sur le bloc M, puis on divise le bloc en deux parties : la moitié gauche G_0 et la moitié droite D_0. Une série d'opérations consiste à passer du bloc (G_{i-1}, D_{i-1}) au suivant (G_i, D_i) en faisant :

$$\begin{aligned} G_i &= D_{i-1} \\ D_i &= G_{i-1} \oplus f(D_{i-1}, K_i) \end{aligned}$$

pour $1 \leq i \leq 16$.

K_i est une sous-clé de 48 bits obtenue à partir de la clé K. $f(D_{i-1}, K_i)$ est obtenu ainsi : D_{i-1} contient 32 bits. On lui fait subir une expansion (ajout de bits à certaines positions) pour obtenir un mot D'_{i-1} de 48 bits. On forme $D'_{i-1}+K_i$ puis on réduit le mot de 48 bits obtenu via des tables pour obtenir un mot de 32 bits. On permute enfin une dernière fois pour arriver à $f(D_{i-1}, K_i)$.

A la fin des 16 séries d'opérations, on applique la permutation p^{-1} pour obtenir le message crypté. Le tableau en annexe permet de se faire une idée de l'algorithme de cryptage, et plus de renseignements sur le D.E.S. pourront être obtenus en [11], en [4] et dans l'article [10].

6 La cryptographie à clé révélée

Un cryptosystème conventionnel montre vite ses limites s'il s'adresse à un public très large. La difficulté réside déjà à déterminer suffisamment de clés et à les distribuer confidentiellement à tous les couples émetteurs-récepteurs. De plus une société désirant communiquer avec un millier de correspondants potentiels devra conserver secrète une liste de mille clés correspondant à chacun de ses interlocuteurs possible, mais sera dans l'obligation d'autoriser plusieurs personnes « sûres » à accéder à cette liste, multipliant dangereusement les risques d'indiscrétion.

Pour finir, les besoins en clés croissent trop vite : des communications cryptées entre n abonnés nécessiteront n^2 clés distinctes K_{ij} correspondant à chacun des couples (E_i, R_j) formés par un émetteur E_i et un récepteur R_j, avec i et j dans $\{1, ..., n\}$.

C'est pour pallier ces inconvénients que Diffie et Hellman ont introduit les *cryptosystèmes à clés révélées* en 1976 ([3], [5]). Le principe en est simple : puisqu'il s'agit de limiter le nombre de clés tenues secrètes, il suffira d'attribuer une clé secrète K au récepteur puis de construire, à l'aide d'une fonction judicieuse T, une clé $T(K)$ qui sera connue de tous les émetteurs.

Evidemment, la fonction T devra être suffisamment compliquée pour que l'on ne puisse pas retrouver la clé K de décodage à partir de $T(K)$, en un temps raisonnable et malgré l'utilisation des plus puissants ordinateurs.

Un tel système à clés révélées possède au moins deux avantages :

- le nombre de clés nécessaires aux communications entre n abonnés devient n au lieu des n^2 précédentes, et sont distribuées seulement aux récepteurs. Elles continueront à être gardées secrètes. Ce sont les clés de décodage.

- les clés de codage $T(K)$, par contre, pourront être connues du public. Le progrès est énorme : ces clés pourront être regroupées dans des annuaires semblables à nos bottins téléphoniques et accessibles à tous par minitel.

Supposons que nous ayons n récepteurs potentiels $R_1, R_2, ..., R_n$ auxquels ont été attribuées n clés $K_1, K_2, ..., K_n$ de décodage, secrètes. Un émetteur E désire envoyer un message M à R_j. Il consulte un annuaire public donnant toutes les clés de codage $T(K_1), T(K_2), ..., T(K_n)$ des abonnés et obtient la clé $T(K_j)$ de R_j. Il envoie le message codé $C_{T(K_j)}(M)$ et le destinataire le décodera en calculant $D_{K_j}C_{T(K_j)}(M)$. On devra évidemment avoir :

$$D_K C_{T(K)}(M) = M$$

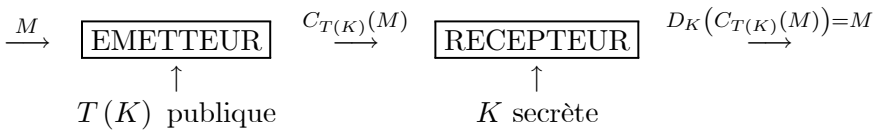
pour toute clé K et pour tout message M.

Mais il y a mieux. Nous avons vu que, deux précautions valant mieux qu'une, il est recommandé de signer son message de sorte que le destinataire R_j puisse avoir la preuve irréfutable que ce message provient bien de E_i. Cette précaution est inutile lorsqu'on utilise une clé simple conservée secrète à la fois par l'émetteur et le récepteur, mais devient importante à partir du moment où n'importe quelle personne peut accéder à la clé de codage. Supposons dorénavant que l'émetteur E soit un des abonnés R_i pour i convenable. $E = R_i$ possède sa propre clé de décodage K_i dont il va se servir en formant le message codé suivant :

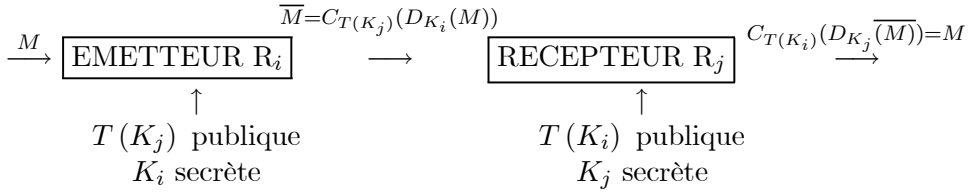
$$\overline{M} = C_{T(K_j)}(D_{K_i}(M))$$

qu'il envoie à R_j. Ce dernier connaît sa clé secrète de décodage K_j et peut lire la clé $T(K_i)$ de codage pour tout envoi de donnée vers R_i sur l'annuaire. Il forme alors $C_{T(K_i)}(D_{K_j}(\overline{M}))$ et obtient le message originel M et la certitude qu'il provient bien de R_i.

Résumons-nous par deux schémas :



Communication à clé révélée sans signature



Communication à clé révélée avec signature

Une telle fonction T qui à une clé K associe une clé $T(K)$, qui rend impossible le calcul de K à partir de $T(K)$ malgré l'utilisation des ordinateurs les plus performants (et pour un temps de calcul raisonnable), et qui vérifie :

$$D_K(C_{T(K)}(M)) = M \quad \text{et} \quad C_{T(K)}(D_K(M)) = M$$

pour tout M , sera appelée *fonction trappe* ou *fonction à sens unique*.

Les paragraphes suivants donnent deux exemples de fonction trappe arithmétique parmi les plus usitées.

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

7.1 La fonction indicatrice d'Euler

Ayant à écrire un message, la première question est de savoir quel alphabet employer. On peut évidemment utiliser l'alphabet usuel ou bien tout ensemble fini de symboles. Le problème réside alors dans la façon de définir les fonctions T, C et D. L'idée bien naturelle est de s'offrir un alphabet muni d'une structure algébrique, par exemple l'anneau quotient :

$$\mathbb{Z}/n\mathbb{Z} = \{\dot{0}, \dot{1}, ..., \overline{n-1}^{\cdot}\}$$

formé des restes des divisions euclidiennes des entiers par n.

La fonction arithmétique φ qui à tout entier naturel non nul n associe le nombre d'entiers naturels premiers avec n et strictement inférieurs à n, est appelée *fonction indicatrice d'Euler*. On sait que $\varphi(n)$ est aussi le cardinal du sous-groupe multiplicatif U des éléments inversibles de l'anneau $\mathbb{Z}/n\mathbb{Z}$.

D'autre part le théorème de Lagrange énonce que si G est un groupe multiplicatif fini de cardinal s et d'élément neutre e, alors tout élément x de G vérifie $x^s = e$. Appliqué au groupe U, ce théorème montre que :

$$\forall \dot{x} \in U \quad \dot{x}^{\varphi(n)} = \dot{1}$$

soit :

$$x^{\varphi(n)} \equiv 1 \ (n)$$

pour tout entier x premier avec n.

Nous aurons aussi besoin de savoir que si p et q sont premiers entre eux, alors :

$$\varphi(pq) = (p-1)(q-1).$$

On pourra se référer à [9] ou [6] pour ce qui concerne la fonction indicatrice d'Euler.

7.2 Principe du système RSA

Donnons-nous deux entiers premiers p et q, et formons $n = pq$. Fixons un entier m premier avec $\varphi(n) = (p-1)(q-1)$. Le Théorème de Bezout montre l'existence de 2 entiers u et v vérifiant $um + v\varphi(n) = 1$. Donnons-nous un message M formé d'une succession de lettres de l'alphabet U inclus dans :

$$\mathbb{Z}/n\mathbb{Z} = \{\dot{0}, \dot{1}, ..., \overline{\dot{n-1}}\}.$$

Notons x une telle lettre. On peut imaginer x comme un entier premier avec n, ie non multiple de p ou q.

De $x^{um+v\varphi(n)} \equiv x$ et $x^{\varphi(n)} \equiv 1 \ (n)$ on déduit $x^{um} \equiv x \ (n)$.

Il suffit de poser :

$$C(x) \equiv x^u \ (n)$$

$$D(y) \equiv y^m \ (n)$$

pour que, x étant quelconque dans $\{1, ..., n-1\}$ et premier avec n, l'on ait :

$$D(C(x)) \equiv D(x^u) \equiv x^{um} \equiv x \ (n)$$

Les fonctions D et C seront nos fonctions de décodage et de codage.

Ce procédé appelle plusieurs remarques :

1) Pour crypter, on a besoin de C, soit de u et de n, qui seront donc du domaine public et accessibles en consultant un annuaire.

2) Pour décrypter, il faut connaître m et n.

3) Avec les notations du §. 6,

$T(K) = (u, n) =$ clé publique de codage,
$K = (m, n) =$ clé secrète de décodage,
T est la fonction trappe.

Si $n = pq$ est connu de tout le monde, p et q seront conservés secrets car leur connaissance entraine celle de $\varphi(n) = (p-1)(q-1)$ et celle de m en résolvant Bezout : $um + v\varphi(n) = 1$, u étant donné dans l'annuaire. Pour rendre le calcul de p et q impossible en un temps raisonnable et en utilisant un ordinateur puissant, il faut que n soit très grand.

Pour donner un ordre de grandeur de n, disons qu'en 1982 un ordinateur aurait été dans l'obligation de fonctionner pendant un million de milliards d'années pour obtenir les deux seuls facteurs p et q d'un nombre n à 126 chiffres (cf [3]). Il y a de quoi décourager les plus tenaces !

Réfléchissons maintenant au dernier obstacle important : est-il facile d'obtenir en un temps raisonnable de tels produits de deux nombres premiers d'environ 63 chiffres chacun ? Là, nous avons beaucoup de chance et c'est ce qui rend la construction RSA si pertinente. Il est en effet beaucoup plus facile, pour un ordinateur, de déterminer si un nombre est premier que de donner sa décomposition en facteurs premiers. Ainsi, en 1983, le temps moyen nécessaire à un gros ordinateur pour tester la primalité d'un entier dans les cas les plus défavorables est donné dans le tableau ci-dessous dû à Pomerance ([8], voir aussi [3] ou [7]).

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

Résumons : on peut « facilement » obtenir deux nombres premiers p et q de 63 chiffres. On effectue leur produit, en une fraction de seconde sur ordinateur, pour obtenir notre nombre $n = pq$ dont on ne pourra plus retrouver la décomposition. L'algorithme d'Euclide permet alors le calcul "rapide" de u une fois m choisi. L'application $T : (m, n) \longrightarrow (u, n)$ mérite bien son nom de fonction trappe puisque la connaissance de la clé publique de codage (u, n) ne rend pas possible celle de la clé secrète de décodage (m, n). Le problème de cryptage à clé révélée est résolu avec l'aide de la fonction φ .

7.3 Exemple numérique

Prenons $p = 13$ et $q = 23$. On aura $n = pq = 299$ et $\varphi(n) = 12 \times 22 = 264$. Une lettre du message sera un élément x de $\{0, 1, ..., 298\}$ premier avec 299, et l'on peut choisir $m = 17$ qui est bien premier avec $\varphi(n)$.

On résout l'équation $17u + 264v = 1$ par la technique des divisions euclidiennes successives :

$$\begin{aligned} 264 &= 17 \times 15 + 9 \\ 17 &= 9 \times 1 + 8 \\ 9 &= 8 \times 1 + 1 \end{aligned}$$

qui permettent d'obtenir, en remontant les calculs :

$$1 = 9 - 8 = 9 - (17 - 9) = 2 \times 9 - 17$$
$$1 = 2(264 - 17 \times 15) - 17 = 17 \times (-31) + 264 \times 2.$$

$(u, v) = (-31, 2)$ est une solution de $17u + 264v = 1$. Les fonctions C et D seront ici :

$$C(x) = x^u = x^{-31} = x^{233} \text{ car } x^{264} = 1,$$
$$D(y) = y^m = y^{17}.$$

Le calcul de $C(x) = x^{233}$ est inquiétant, mais il ne faut pas oublier que l'on travaille modulo 299, ce qui permet une simplification à chaque pas de calcul. Prenons par exemple $x = 200$.
$x^2 = 40000 \equiv 233 \ (299)$
$x^3 = x^2.x \equiv 233 \times 200 \equiv 255 \ (299)$
$x^4 = x^3.x = 255 \times 200 \equiv 170 \ (299)$
etc

Un petit programme permet de calculer $C(x)$ et $D(y)$. On trouve par exemple ici $C(200) \equiv 187$, puis $D(187) \equiv 187^{17} \equiv 200$ pour le décodage.

Passons maintenant à une autre solution mettant en jeu une fonction trappe...

8 La méthode des empilements

8.1 Problème de la pile

Donnons-nous n boîtes de hauteurs respectives $a_1, ..., a_n$ distinctes deux à deux. Choisissons-en certaines que nous empilons et mesurons la hauteur h de la pile ainsi obtenue. « Résoudre le problème d'empilement pour ces n boîtes » signifie retrouver les boîtes que nous avons utilisées pour constituer la pile à partir des seules connaissances de la hauteur h et des hauteurs $a_1, ..., a_n$ des boîtes susceptibles d'être utilisées. En d'autres termes, il s'agit de résoudre l'équation :

$$a_1x_1 + ... + a_nx_n = h$$

où les inconnues $x_1, ..., x_n$ ne peuvent prendre que les valeurs 0 ou 1.

Un tel problème est difficile à résoudre dès que n devient suffisamment grand car l'on ne connaît pas de méthode plus rapide que celle consistant à effectuer les 2^n essais correspondant aux 2^n valeurs possibles de la n-liste $(x_1, ..., x_n)$.

Le problème de la pile posé avec $a_1, ..., a_n$ est un problème de taille n qui exige un temps de résolution proportionnel au nombre 2^n de tests à effectuer.

Comme 2^n croît plus vite que toute fonction polynomiale de n, on dira que le problème de la pile est un *problème de classe NP*, c'est-à-dire non résoluble en temps polynomial.

En d'autres termes, il n'existe pas de méthode connue permettant de résoudre ce problème de taille n en un temps qui soit une fonction polynomiale de n. La résolution du problème de la pile est difficile en un temps raisonnable et même si l'on dispose d'un ordinateur extrêmement puissant.

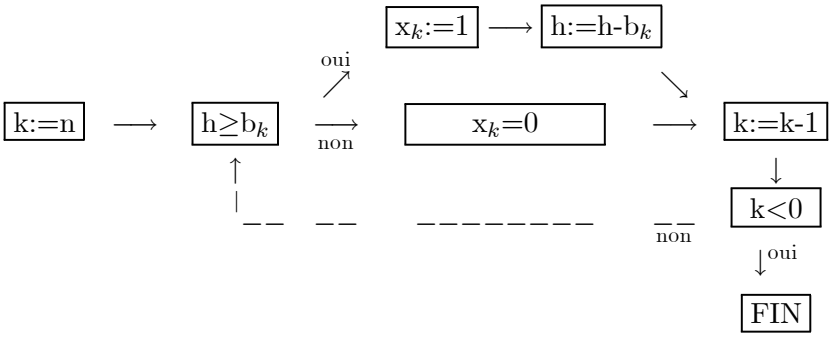
8.2 Un cas particulier salvateur

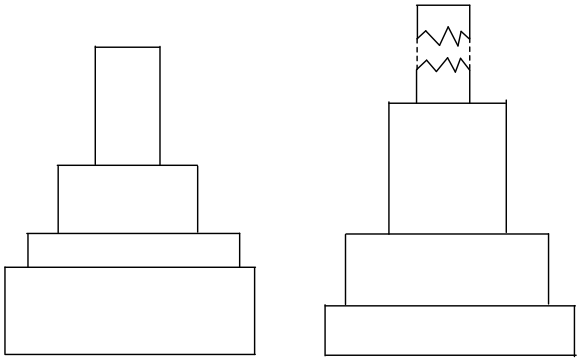
Supposons que les hauteurs des n boîtes soient $b_1, ..., b_n$ et vérifient cette fois-ci :

$$\forall k \in \{2, ..., n\} \quad b_1 + b_2 + ... + b_{k-1} < b_k \quad (*)$$

Si h désigne la hauteur d'une pile formée par certaines de ces boîtes, et si $h \geq b_n$, alors $x_n = 1$. En effet $x_n = 0$ entraînerait $h \leq b_1 + ... + b_{n-1} < b_n$, ce qui est absurde.

Par contre si $h < b_n$ alors $x_n = 0$. On recommence avec b_{n-1} et ainsi de suite. On obtient l'algorithme suivant :





Pile dans le cas général

Cas particulier

Cet algorithme est précieux car utilise seulement n tests successifs au lieu des 2^n essais de la méthode générale. Le problème de la pile muni de cette nouvelle hypothèse sur les b_i devient résoluble en temps polynomial.

8.3 Le cryptage

Par quel artifice peut-on passer du problème facile vu en 8.2 à celui non résoluble en temps polynomial évoqué en 8.1 ? Et bien c'est l'arithmétique des congruences qui fera le travail : partant de $b = (b_1, ..., b_n)$ satisfaisant la condition $(*)$ du 8.2, on fixe 2 entiers naturels m et w premiers entre eux et l'on calcule :

$$a_i \equiv b_i.w \ \ (m)$$

pour tout i. Cela rend la recherche des b_i à partir des seuls a_i impossible. Si $\dot{x}$ désigne la classe de $x \in \mathbb{Z}$ dans l'anneau $\mathbb{Z}/m\mathbb{Z}$, on sait que $\dot{x}$ est inversible ssi x est premier avec m. $\dot{w}$ sera donc inversible dans $\mathbb{Z}/m\mathbb{Z}$ et il existera un entier w^{-1} tel que $\dot{w}.\dot{w}^{-1} = \dot{1}$, i.e. $w.w^{-1} \equiv 1 \ \ (m)$, de sorte que l'on ait :

$$a_i.w^{-1} \equiv b_i \ \ (m)$$

pour tout i.

Supposons que l'émetteur veuille envoyer le message $x = (x_1, ..., x_n)$ où $x_i = 0$ ou 1. Il utilise sa clé de codage $a = (a_1, ..., a_n)$ pour former le nombre :

$$C(x) = a_1x_1 + ... + a_nx_n = h$$

qu'il transmet au récepteur. Ce dernier applique la clé de décryptage (w, m, b) et forme :

$$h.w^{-1} \equiv a_1w^{-1}x_1 + ... + a_nw^{-1}x_n \ \ (m)$$
$$h.w^{-1} \equiv b_1x_1 + ... + b_nx_n \ \ (m).$$

Quitte à prendre des représentants modulo m de $h.w^{-1}$ situés dans l'intervalle $[0, m[$ et à supposer que $0 \leq b_1 + ... + b_n < m$, on aboutit à :

$$h.w^{-1} = b_1x_1 + ... + b_nx_n$$

soit au problème de la pile résoluble selon la méthode du 8.2. On possède bien un système de cryptographie publique à clés révélées puisque se résumant à la donnée :

- d'une clé secrète de décodage $K = (w, m, b)$,
- d'une clé publique $T(K) = a$ pour le codage,
- d'une fonction trappe T fournissant la clé publique $T(K)$ à partir de la clé secrète K. Il s'agit de $T(w, m, b) = wb$.

8.4 Exemple numérique

Soit $b = (b_1, ..., b_{12}) = (1, 3, 5, 10, 23, 45, 88, 180, 357, 715, 1430, 2865)$. Choisissons un entier $m > b_1 + ... + b_{12} = 5722$, par exemple $m = 5849$ qui est premier. Prenons $w = 3903$. L'algorithme d'Euclide permet de trouver u et v tels que $3903u + 5849v = 1$ et d'en déduire l'inverse de w modulo 5849, on trouve $w^{-1} = 532$.

Calculons $a = bw$ que l'on réduit modulo m :

$$a = bw = (3903, 11, 1968, 3936, 2034, 165, 4222, 660, 1309, 672, 1344, 4656).$$

Supposons que 100101110100 soit le message à envoyer. On calcule le message crypté :

$$\begin{aligned} h &= 3903 + 3936 + 165 + 4222 + 660 + 672 \\ &= 13558 \equiv 1860 \ (5849) \end{aligned}$$

que l'on envoie sur un canal non protégé. Le décryptage consiste à résoudre l'équation :

$$hw^{-1} = b_1x_1 + ... + b_{12}x_{12}$$

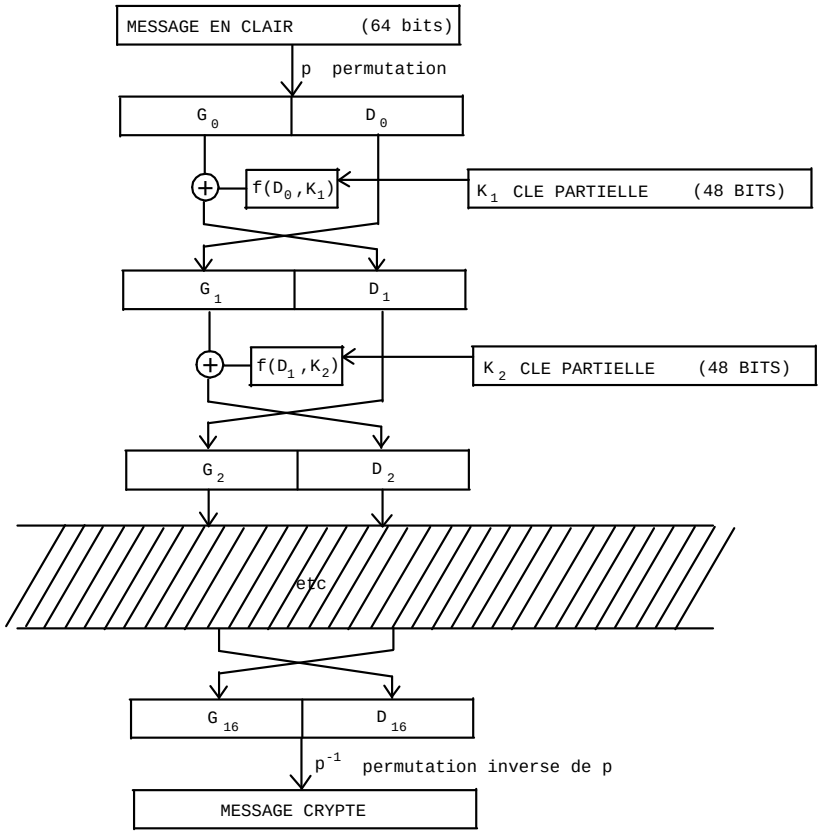
Calculons alors $hw^{-1} = 1860 \times 532 = 989520 \equiv 1039 \ (5849)$. On applique l'algorithme de la pile simple :

$1039 < 1430$ donc $x_{11} = x_{12} = 0$,
$1039 > 715$ donc $x_{10} = 1$ et $b_1x_1 + ... + b_9x_9 = 1039 - 715 = 324$,
$324 < 357$ donc $x_9 = 0$,
$324 > 180$ donc $x_8 = 1$ et $b_1x_1 + ... + b_7x_7 = 324 - 180 = 144$,
$144 > 88$ donc $x_7 = 1$ et $b_1x_1 + ... + b_6x_6 = 144 - 88 = 56$,
$56 > 45$ donc $x_6 = 1$ et $b_1x_1 + ... + b_5x_5 = 56 - 45 = 11$,
$11 < 23$ donc $x_5 = 0$,
$11 > 10$ donc $x_4 = 1$ et $b_1x_1 + ... + b_3x_3 = 11 - 1 = 10$,

et cela entraîne $x_1 = 1$ et $x_2 = x_3 = 0$.

ANNEXE

Les 16 opérations successives du D.E.S.



References

[1] Adleman L.M., Rivest R.L. and Shamir A., A method for obtaining digital signatures and public-key cryptosystems, Communications of the ACM, vol.21, 1978, pp. 120-126.

[2] Bauer Friedrich L. : Cryptology, methods and maxims, article dans [4].

[3] Bouvier Alain : Cryptographie publique, pp. 77-83 de *Didactique des Mathématiques, le dire et le faire*, Cedic-Nathan, 1986.

[4] Cryptography, Proceedings, Burg Feuerstein 1982, Lecture Notes in Computer Science 149, Springer-Verlag, 1983.

[5] Hellman Martin : Les mathématiques de la cryptographie à clé révélée, revue *Pour la Science* numéro 24 d'octobre1979, pp. 114-123.

[6] Koblitz Neal, A course in number theory and cryptography, Graduate Texts in Mathematics, 114, Springer-Verlag, seconde édition, 1988.

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

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

[9] Querré : Cours d'algèbre, Masson, 1976. L'indicateur d'Euler est traité à l'exemple 8 p61.

[10] Ryska, N. et Herda S. : Kryptographische Verfahren in der Datenverarbeitung, Infomatik-Fachberichte 24, Springer-Verlag 1980.

[11] Tarnowski Daniel : Cartes à puces : le secret du code inviolable, Science & Vie 917 de février 94, pp. 87-93.