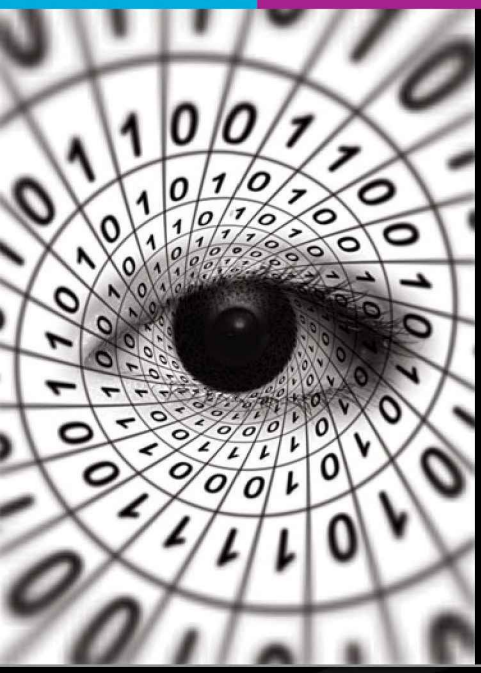


UNE INTRODUCTION À

INTRODUCTION

LA CRYPTOLOGIE

L'art des codes secrets

Philippe Guillot

edp sciences

Collection « Une Introduction à »
dirigée par Michèle Leduc et Michel Le Bellac

La cryptologie

L'art des codes secrets

Philippe Guillot

edp sciences

17, avenue du Hoggar
Parc d'activités de Courtabœuf, BP 112
91944 Les Ulis Cedex A, France

Dans la même collection

Les atomes froids
Erwan Jahier, préface de M. Leduc

Le laser
Fabien Bretenaker et Nicolas Treps, préface de C. H. Townes

Le monde quantique
Michel Le Bellac, préface d'A. Aspect

Les planètes
Thérèse Encrenaz, préface de J. Lequeux

Naissance, évolution et mort des étoiles
James Lequeux

La fusion thermonucléaire contrôlée
Jean-Louis Bobin

Le nucléaire expliqué par des physiciens
Bernard Bonin, préface d'É. Klein

Mathématiques des marchés financiers
Mathieu Le Bellac et Arnaud Viricel, préface de J.-Ph. Bouchaud

Physique et biologie
Jean-François Allemand et Pierre Desbiolles

Retrouvez tous nos ouvrages et nos collections sur
http://www.edition-sciences.com

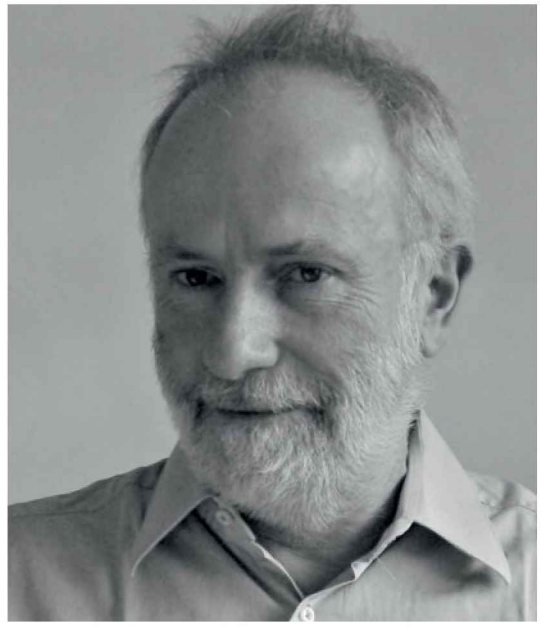
Illustration de couverture : Clément Doranlo

Imprimé en France.

© **2013, EDP Sciences**, 17, avenue du Hoggar, BP 112, Parc d'activités de Courtabœuf,
91944 Les Ulis Cedex A

Tous droits de traduction, d'adaptation et de reproduction par tous procédés réservés pour tous pays. Toute reproduction ou représentation intégrale ou partielle, par quelque procédé que ce soit, des pages publiées dans le présent ouvrage, faite sans l'autorisation de l'éditeur est illicite et constitue une contrefaçon. Seules sont autorisées, d'une part, les reproductions strictement réservées à l'usage privé du copiste et non destinées à une utilisation collective, et d'autre part, les courtes citations justifiées par le caractère scientifique ou d'information de l'œuvre dans laquelle elles sont incorporées (art. L. 122-4, L. 122-5 et L. 335-2 du Code de la propriété intellectuelle). Des photocopies payantes peuvent être réalisées avec l'accord de l'éditeur. S'adresser au : Centre français d'exploitation du droit de copie, 3, rue Hautefeuille, 75006 Paris. Tél. : 01 43 26 95 35.

ISBN 978-2-7598-0811-3



Né en 1956, Philippe Guillot a été ingénieur de recherche en cryptologie à Thomson-CFS à partir de 1990. Il est devenu chef du laboratoire de cryptologie de Thales jusqu'en 2001, puis responsable du pôle sécurité de Canal-Plus Technologies de 2001 à 2003. Depuis 2003, il est maitre de conférences à l'Université Paris 8 en charge des cours de cryptologie, d'histoire de la cryptologie et d'algorithmique algébrique dans le master « mathématiques fondamentales et protection de l'information ».

Table des matières

Préface

La numérisation massive des données et le développement des réseaux de communication ont rendu possibles certains traitements automatisés auparavant inimaginables. En une fraction de seconde, le médecin peut désormais accéder à l'ensemble du passé médical et des résultats d'analyses de son patient ; le policier peut retrouver le nom d'un suspect à partir de son ADN ; un client peut consulter la liste des opérations effectuées sur son compte en banque et donner un ordre de virement depuis son téléphone mobile. Mais un individu mal intentionné peut aussi usurper l'identité d'un client et effectuer un virement en son nom ; il peut tenter d'obtenir des informations sur l'état de santé d'une personnalité connue, établir et rendre publiques des listes de patients atteints de maladie grave ou manipuler des dizaines de milliers de dossiers médicaux pour dissimuler les effets secondaires d'un traitement.

Si les technologies impliquées sont nouvelles, cette situation ne l'est évidemment pas : l'apparition de nouvelles techniques et l'ouverture d'un champ de possibilités inédit vont généralement de pair avec de nouveaux risques. En même temps que les avions sont apparus les accidents aériens et les pirates de l'air... Fort heureusement, le législateur, les scientifiques et les ingénieurs ont établi de nombreux garde-fous : les comportements répréhensibles sont punis par la loi, certaines opérations sont rendues techniquement impossibles. Mais ces solutions ne sont pas toutes parfaites et tous les risques ne peuvent pas être évités. Il ne faut pas pour autant tomber dans un rejet en bloc de ces technologies par peur du danger, réaction tout aussi absurde qu'une utilisation systématique sans évaluation critique des conséquences éventuelles. Comme dans tout autre domaine, le citoyen a besoin d'être conscient des garanties offertes par un système, et des risques encourus. Peut-être encore plus que dans tout autre domaine, un système informatique n'est socialement acceptable que s'il recueille la confiance des citoyens, et la confiance ne peut être établie que par la connaissance. Cette connaissance ne doit en aucun cas être réservée à quelques informaticiens ou férus de mathématiques au prétexte qu'ils seraient les seuls capables d'appréhender la complexité du domaine. Il ne s'agit pas que chacun maîtrise tous les détails

du fonctionnement de sa carte bancaire, mais qu'il comprenne ses fonctionnalités, les garanties qu'elle offre et celles qu'elle n'offre pas. Un patient qui doit subir une opération chirurgicale n'a nul besoin de savoir manier un bistouri, mais il est souhaitable qu'il sache à quoi sert une anesthésie. De la même façon, il est important que tout citoyen apprenne que les techniques permettant de protéger nos données et communications n'ont pas toutes les mêmes fonctionnalités, qu'elles n'offrent pas toutes la même sécurité et qu'elles ont bien sûr des limites.

C'est à cette œuvre salutaire que s'est attelé Philippe Guillot. Il a su brosser un panorama détaillé des techniques cryptologiques, de leurs applications et de leurs limites, qui ne soit pas un cours de troisième cycle universitaire, mais un ouvrage accessible à un large public. Pour autant, cet ouvrage ne se cantonne pas dans la présentation des procédés les plus simples, comme le RSA. Il évoque les techniques utilisées en pratique qui sont pourtant négligées par bien des auteurs, au prétexte qu'elles seraient moins élégantes, plus complexes ou trop récentes. Philippe Guillot parvient ainsi à donner une vue d'ensemble de la cryptologie qui est à la fois réaliste puisqu'elle englobe les méthodes employées dans la plupart des applications, et accessible. Par ses qualités pédagogiques hors du commun, il réussit même à familiariser le lecteur novice avec la sécurité physique des cartes à puce ou avec des concepts mathématiques aussi avancés que l'appariement de Weil. La lecture de cet ouvrage permettra ainsi à chacun de comprendre pourquoi aucun des procédés cryptographiques utilisés en pratique n'est parfaitement sûr, pourquoi la taille d'une clef cryptographique dépend fortement du système employé, ou comment les paramètres d'un système doivent évoluer au fil du temps pour prendre en compte l'augmentation de la puissance des ordinateurs. Elle apportera un éclairage indispensable à qui veut appréhender avec un esprit critique la mise en place de nouvelles téléprocédures, le déploiement de vastes systèmes d'information de santé ou de machines à voter.

Anne CANTEAUT,
directrice de recherche,
Inria Paris-Rocquencourt

Avant-propos

Alice aime son travail de paysagiste dans l'entreprise Thagem où elle doit aménager l'environnement de travail des mille cinq cents employés du site de Palombes-sur-Seine. L'essentiel de son activité est en plein air. C'est le printemps, les bouleaux lâchent leur pollen, et tout irait pour le mieux sans ce maudit rhume des foins qu'elle traîne depuis son adolescence. Ce soir en quittant le travail, il faudra qu'elle passe voir son médecin pour se faire prescrire un traitement anti-allergique.

En descendant les escaliers de son appartement parisien, elle allume son téléphone mobile :

– Allô, docteur Maison ? Puis-je passer vous voir cette après-midi vers 17 h 30 ?

Le rendez-vous est rapidement pris. La journée commence bien. Elle croise sans le remarquer le facteur venu déposer le courrier dans le hall de son immeuble et s'engouffre dans le métro, passe machinalement son sac à main le long du tourniquet et pense déjà aux aventures du commissaire Evenberg, héros du roman qu'elle a commencé avant-hier et qui lui fera passer plus vite son trajet.

Après avoir présenté son badge aux tourniquets d'accès de Thagem, son esprit commute déjà sur ses tâches de la journée. Elle démarre la fourgonnette de service pour aller prendre livraison des nouveaux rosiers destinés à agrémenter les abords du lac artificiel, fierté du directeur, et qui a obtenu un prix du meilleur environnement d'entreprise de la région.

À midi, elle vérifie le solde de la carte Moneix qui lui permet de payer le repas sans avoir à se préoccuper de faire l'appoint aux caisses. 1€ 23. Elle doit la recharger.

La journée passe vite. Elle repasse le tourniquet vers la sortie. C'est l'heure de son rendez-vous chez le médecin. Il fait beau. Elle décide de prendre un vélo en libre service avec son passe Circulo.

Elle avait oublié le changement d'adresse du docteur Maison ! Sans se démonter, elle télécharge l'application de navigation sur son téléphone qui lui indiquera la nouvelle adresse et l'itinéraire pour arriver à l'heure.

– Puis-je avoir votre carte Vitalix ?

Alice se laisse ausculter, et se réjouit d'avance à l'idée de soulager son nez bouché, ses démangeaisons et l'irritation insupportable de ses yeux.

– Vous n'avez qu'une sévère allergie au pollen, je n'ai rien remarqué d'autre, vous prendrez du Rhumactine en cas de production nasale abondante.

Alice sourit intérieurement en pensant au vocabulaire médical.

– Cela fera vingt-trois euros.

– Acceptez-vous la carte bancaire ?

– Oui, je préfère même ! Avoir moins d'espèces dans mon cabinet me rassure. Je me suis déjà fait braquer.

De retour dans son appartement, elle branche son ordinateur en se souvenant soudain qu'aujourd'hui est la date limite pour valider la déclaration de revenus du foyer.

« Une mise à jour est disponible pour votre ordinateur, télécharger ? »

– Encore !

Elle accepte la mise à jour, l'ordinateur redémarre. Enfin elle valide la déclaration des revenus.

Elle en profite pour commander sur Mississipi.fr la suite des aventures du commissaire Evenberg qui viennent de paraître.

C'est fini pour les préoccupations de la journée. Il est temps de se détendre avec Bob en allumant le téléviseur. Il y a au programme un bon film du cinéma italien des années soixante-dix sur la chaîne thématique à laquelle ils sont abonnés.

Cette tranche de vie fait intervenir pas moins de quinze situations au cours desquelles ont été menées une ou plusieurs opérations cryptologiques. Ceci illustre à quel point ce domaine a, en quelques années, envahi notre quotidien, sans que nous en ayons toujours pleinement conscience. Le lecteur est invité à identifier ces situations avant de consulter la solution page 173.

La cryptologie, née du besoin de transmettre des messages au seul destinataire autorisé, et dont le sens reste caché au messager et à quiconque pourrait l'intercepter, rassemble aujourd'hui un ensemble de méthodes destinées à protéger toute information contre une observation ou une intrusion malveillante.

En raison de la sensibilité des informations échangées, les milieux militaires et gouvernementaux sont naturellement intéressés à l'utilisation de la cryptologie. Avec l'essor des réseaux de télécommunication et la banalisation des données enregistrées, ces problèmes de sécurité concernent un ensemble de plus en plus large de la population.

Le développement de l'internet n'a été rendu possible qu'avec la confiance apportée par les moyens de protection des informations qu'il véhicule. Alors que jusqu'en 1998, l'utilisation des moyens cryptologiques était un monopole d'État, ces moyens ayant le statut d'arme de guerre, au même titre que les munitions et les explosifs, aujourd'hui, « l'usage des moyens de cryptologie est libre », comme il est stipulé dans l'article 30 de la loi pour la confiance dans l'économie numérique du 21 juin 2004.

Une connaissance des procédés mis en œuvre devient nécessaire pour en comprendre et en maîtriser l'usage. Un objectif de cette introduction à la cryptologie est de contribuer à la diffusion de ce savoir au plus grand nombre.

Quel service la cryptologie rend-elle ?

La cryptologie rend principalement deux services : la confidentialité et l'authentification. La problématique de la confidentialité est celle de la discrétion et du secret. L'information ne doit être accessible qu'à celui ou celle à qui l'information est destinée.

- Le programme de TV à péage ne doit être visible que par les abonnés.
- L'ordre des généraux, même s'il est intercepté, ne doit pas être connu de l'ennemi.
- Les parents de Jonathan doivent rester dans l'ignorance du lieu et de l'heure de son rendez-vous avec Olive.

Dans un système de confidentialité, l'adversaire est une oreille indiscrète.

La problématique de l'authentification est de s'assurer que l'information provient bien de l'émetteur annoncé, et qu'elle n'a été ni altérée ni intentionnellement modifiée au cours de son transfert ou de son stockage.

La banque vient de recevoir de Madame Betty Court un ordre de virement de dix mille ducats au bénéfice de son homme de confiance :

- Est-ce vraiment Betty Court qui a émis cet ordre de virement ?
- Le bénéficiaire désigné par elle est-il bien son homme de confiance ?
- Est-ce bien ce montant qu'elle a décidé de virer ?

L'adversaire d'un système d'authentification est un faussaire.

Les deux faces de la cryptologie

La cryptologie est la réunion de deux disciplines qui s'alimentent l'une l'autre : la cryptographie et la cryptanalyse. Le cryptographe conçoit des codes et des procédés résistants qui rendent le service de confidentialité ou d'authentification. Le cryptanalyste les attaque, brise leur résistance, cherche une faille, pour retrouver le sens caché du message, ou pour faire passer un faux pour un vrai. Ce sont ces attaques qui fournissent au cryptographe les critères de conception qui rendront ses procédés plus sûrs encore. Et ils seront à nouveau passés au crible du cryptanalyste.

Les utilisateurs d'un système cryptographique disposent d'un paramètre secret, d'une clé, détenue d'eux seuls, et sur lequel repose toute la sécurité. Si cette clé venait à être connue, tout le système s'effondrerait. Le cryptanalyste, lui, ne dispose pas de cette clé, et cherche quand même à pénétrer les messages.

L'activité cryptologique est encore présentée comme une course sans fin entre les codeurs et les briseurs de code. Une approche récente revendique une démarche scientifique et cherche à éviter cette boucle infinie en proposant dès la conception des preuves de sécurité.

Le premier chapitre présente les procédés de chiffrement traditionnels, balayant les moyens mis en œuvre depuis l'Antiquité grecque jusqu'à la mécanisation du calcul dans le courant du vingtième siècle. Leur présentation, suivant un point de vue historique, est l'occasion de voir comment les principaux concepts, encore en vigueur aujourd'hui, ont été introduits.

Le second chapitre est consacré aux procédés symétriques actuels. Ils sont appelés ainsi parce que la clé, c'est-à-dire le secret qui va permettre d'une part de chiffrer et d'autre part de déchiffrer, est partagée de manière symétrique entre l'émetteur et le destinataire du message.

Une avancée majeure est survenue dans le milieu des années 1970, avec l'invention de la cryptologie à clé publique, objet du troisième chapitre. Dans cette nouvelle cryptologie, les clés de l'émetteur et du destinataire ne sont plus identiques. Ces systèmes sont asymétriques. Et même, plus surprenant, la clé de l'émetteur peut sans inconvénient être dévoilée. Elle est publique. Le destinataire pourra déchiffrer à l'aide d'une clé différente qui, bien sûr, devra rester secrète. Ces mécanismes font appel à des mathématiques de plus en plus élaborées et ont permis de concevoir une multitude de nouveaux services, comme la signature, l'authentification, le vote, la monnaie électronique, etc.

Le quatrième chapitre est consacré à la cryptanalyse qui est le chiffre d'attaque, celui de l'adversaire qui cherche à pénétrer les messages sans disposer de la clé secrète qui lui permettrait de mettre à nu toutes les informations. Les principales attaques contre les systèmes symétriques sont présentées, ainsi que les algorithmes mathématiques qui permettent de contrer les systèmes à clé publique. La fin de ce chapitre présente un nouveau type d'attaque, non plus contre le procédé de camouflage, mais contre le dispositif matériel qui le réalise. Ces attaques reposent sur des mesures physiques, comme la consommation électrique ou le temps de calcul.

Le cinquième chapitre passe en revue la cryptographie de notre quotidien, en décrivant comment elle intervient dans nos objets familiers : carte de paiement, téléphone mobile, télévision à péage, ainsi que les protocoles qui assurent la sécurité de l'internet et qui ont permis de développer la confiance dans le commerce électronique.

L'émergence d'une science cryptologique autonome, dont l'objectif est de donner des assurances sur le niveau de sécurité atteint, est présentée au sixième chapitre. L'évaluation de la résistance des systèmes symétriques est fondée sur la théorie de l'information, qui est née de travaux effectués pendant la seconde

guerre mondiale pour établir des liaisons hautement sécurisées. Les preuves de sécurité des systèmes asymétriques sont plus récentes et dépendent des capacités de calcul de l'adversaire. On ne parle plus de sécurité inconditionnelle mais de sécurité calculatoire. Ce chapitre présente l'édifice cryptographique, qui consiste en une construction de méthodes de protection à clés secrètes ou à clés publiques, s'appuyant sur des fonctions élémentaires aux propriétés admises, dans une démarche de type axiomatique.

Le septième et dernier chapitre montre comment les sciences physiques font aujourd'hui leur entrée dans la cryptologie. Une contribution concerne l'attaque. Si un calculateur quantique voyait le jour, il rendrait inopérant la plupart des systèmes asymétriques actuels. L'autre contribution concerne la défense. La physique quantique apporte une solution originale au problème de l'échange de clé, en proposant un mécanisme, aujourd'hui fonctionnel, dont la sécurité repose, non plus sur certains problèmes mathématiques que l'on suppose difficiles à résoudre, mais sur les lois de la physique.

Trouverez-vous la clé ?

1

Les procédés traditionnels

Ce chapitre traite d'une période qui s'étend de l'Antiquité grecque jusqu'au début du vingtième siècle, quand le développement des calculateurs mécaniques, électromécaniques, puis électroniques a marqué le début d'une nouvelle ère. Cette cryptologie traditionnelle est un traitement de la langue écrite, avant d'être un calcul. Ce parcours historique montre comment les principaux concepts, toujours en vigueur aujourd'hui, ont été introduits.

1 Les substitutions simples

La première idée qui vient à l'esprit pour brouiller un texte écrit dans une langue à alphabet consiste à remplacer chaque lettre par une autre selon une règle convenue. Ce procédé s'appelle une *substitution simple*. Le chiffre de César en est un exemple. Il est réalisé en décalant l'alphabet. Il est mentionné par les historiens Suetone (vers 69, vers 130) et Aulu Gelle (vers 130, vers 180).

> *On possède enfin de César des lettres à Cicéron, et sa correspondance avec ses amis sur ses affaires domestiques. Il écrivait, pour les choses tout à fait secrètes, à travers des marques, c'est-à-dire un ordre arrangé de lettres de sorte qu'aucun mot ne pût être reconnu. Si on veut chercher et s'acharner jusqu'au bout, on change la quatrième lettre, c'est-à-dire un D à la place d'un A et pareillement pour toutes les autres.*
>
> Suetone, *La vie des douze Césars*

Le procédé de César tel que décrit ci-dessus consiste à appliquer un décalage de trois rangs dans l'alphabet. Voici la correspondance des vingt-trois lettres de l'alphabet du latin classique.

```
A B C D E F G H I K L M N O P Q R S T V X Y Z
↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓
D E F G H I K L M N O P Q R S T V X Y Z A B C
```

Pour retrouver le sens du message en clair, il suffit d'opérer le même décalage dans l'autre sens. Ce procédé subsiste encore aujourd'hui sous le nom de ROT13 qui opère un décalage de treize positions dans l'alphabet moderne. Il est utilisé pour brouiller un texte dans le réseau usenet qui sert encore à l'échange d'informations au sein d'une communauté. Ce décalage permet d'utiliser la même opération pour brouiller et pour clarifier le texte. Il n'y a aucun secret dans ce système qui ressemble plutôt à un *argot d'internet*.

La citation de Suetone montre que César utilisait ce procédé pour ses correspondances privées, et non pas pour des secrets militaires. Lors de ses campagnes, il utilisait un autre procédé qu'il cite lui-même dans *La guerre des Gaules*.

> *Il persuade alors un cavalier gaulois, en lui promettant de grandes récompenses, de porter une lettre à Cicéron. Il envoie celle-ci écrite en* ***lettres grecques****, afin que, si elle est interceptée, nos desseins ne soient pas pénétrés par les ennemis.*
>
> César, *La guerre des Gaules, V, XLVIII, 3–4*

Voici un message qu'aurait pu envoyer Jules César à son intendant, lorsque, retenu en Gaule dans ses quartiers d'hiver avec son armée, il en profite pour mettre à contribution les populations soumises par un tribut de guerre tout à fait habituel à l'époque (solution page 174) :

```
MMX KDOOMX TZM SVRAMPH LRXYHX IZHVDQY PDKQDX SHFZQMDX MPSHVDZM
SRSZOR VRPDQR, TZDX MQYVD GHFHP GMHX DFFMSMHX. P. YZOOMR DHX
DOMHQZP XROZHVH SRYHVMX HPHVHTZH D S. ZDOHVMR ZMOODP HMZX
FHQYZP PMOMEZX QZPPZP, MG HXY SVHYMZP TZRG FRQZHQHVDY.
```

Ces deux procédés de César ont en commun qu'ils opèrent le remplacement d'une lettre de l'alphabet par un signe différent. De nombreux autres graphismes sont utilisables pour camoufler du sens du message. Les Templiers utilisaient un alphabet spécial reposant sur la *croix de huit béatitudes* qui étaient l'emblème de leur ordre (Fig. 1.1). Les écoliers de toutes les époques reconnaîtront *le parc à cochon* (Fig. 1.2), utilisé aussi par les francs-maçons.

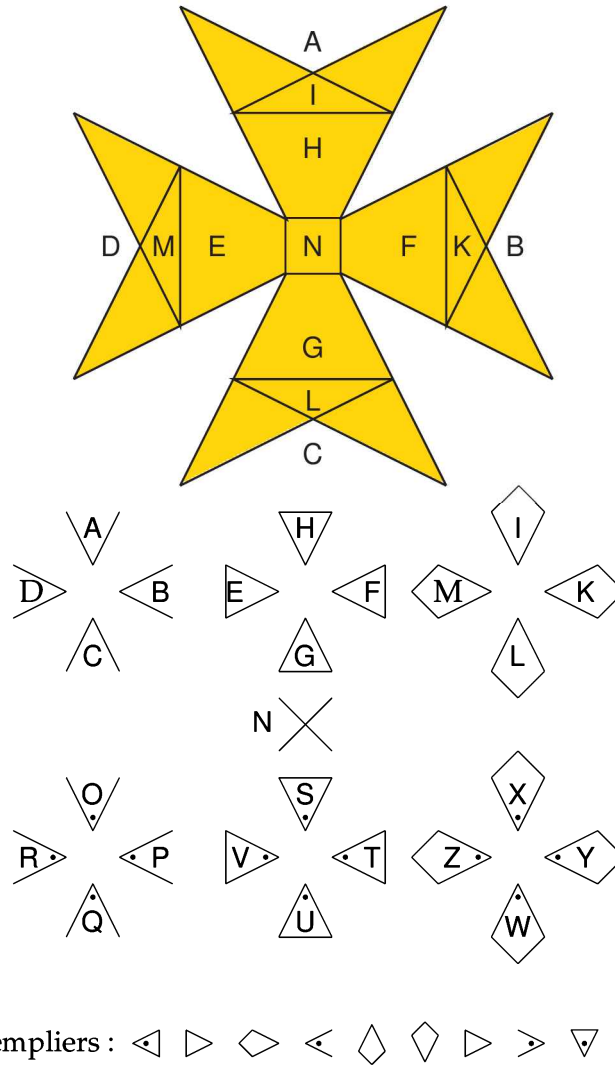


FIGURE 1.1. La croix des huit béatitudes, emblème de l'ordre des Templiers, est le support de l'alphabet dont se servaient les Templiers pour transmettre leurs lettres de crédit. Chaque lettre est représentée par un graphisme qui correspond à sa position sur la croix des huit béatitudes.

Dans la nouvelle *Les hommes dansants*, d'Arthur Conan Doyle, parue en 1903, Sherlock Holmes réussit à décrypter de mystérieux messages dessinés au crayon sur du papier, ou à la craie sur des murs, et où figurent les silhouettes de personnages qui ont l'air de danser (Fig. 1.3).

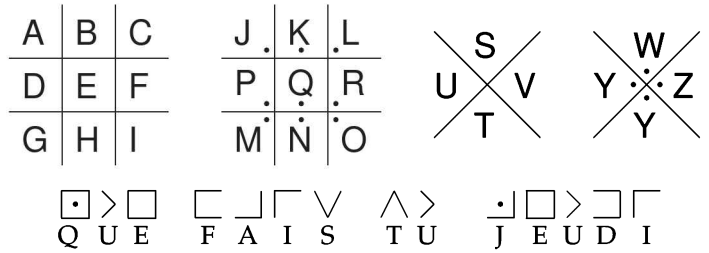


FIGURE 1.2. Le parc à cochon, procédé très ancien cité par Vigenère dans son *Traité des chiffres et des secrètes manières d'écrire*, Paris, 1586.

FIGURE 1.1. Les hommes dansants. Chaque figurine représente une lettre. Le talent de Sherlok Holmes et l'analyse des fréquences sont aisément venus à bout de ces mystérieux messages.

Si on se limite au décalage de César sur l'alphabet moderne, il n'existe que que vingt-cinq façons d'opérer et donc autant de décalages à tester pour retrouver le sens caché. Cela est réalisable manuellement, sans même l'aide d'outil de calcul. Il est plus sûr d'imaginer une substitution plus générale, qui remplace une lettre par une autre sans relation particulière. Il est seulement nécessaire que les lettres du texte en clair et celles du cryptogramme se correspondent une à une. Ce qu'on obtient ainsi est un *alphabet désordonné* comme par exemple celui-ci :

M R Z T C K E X F B U V Q G H Y D J P I O W N L A S

Le mode d'emploi de cet alphabet désordonné est des plus simples. Le A est remplacé par le M, le B par le R, le C par le Z, etc.

Le nombre d'alphabets désordonnés possibles est considérable. Il y a 26 choix possibles pour la première lettre, 25 pour la seconde, 24 pour la troisième, etc. Le nombre total d'alphabets désordonnés est le produit de tous les entiers de 26 jusqu'à 1, appelé la *factorielle* de 26 et noté 26! :

$$26 \times 25 \times 24 \times \cdots \times 2 \times 1 = 403\,291\,461\,126\,605\,635\,584\,000\,000.$$

Il n'est plus question d'essayer systématiquement toutes les possibilités, même en utilisant une machine très performante. Un super calculateur capable d'effectuer un milliard d'essais par seconde mettrait plus d'un milliard d'années pour explorer tous les alphabets possibles. Le cryptanalyste doit faire preuve d'habileté pour retrouver le sens caché d'un cryptogramme.

Les substitutions simples sont sensibles à l'analyse des fréquences, qui consiste à compter les occurrences des caractères et à comparer le résultat avec la distribution des lettres dans la langue du texte en clair. Le paragraphe 3 page 85 décrit les méthodes de résolution de ce type de chiffre. Pour le rendre plus résistant, les cryptographes ont imaginé de coder les lettres les plus fréquentes, comme le *e* ou le *a* de plusieurs façons différentes en alternant le choix du codage. En contrepartie, les lettres qui peuvent être remplacées par une autre sans inconvénient pour la compréhension sont supprimées. En français, on peut par exemple remplacer les *i* par des *j*, les *v* par des *u*. Ce type de substitution à représentations multiples s'appelle un *chiffre homophonique*. Il a été très utilisé pour les échanges diplomatiques à partir de la Renaissance.

2 Transpositions

Dans un procédé de transposition, les lettres du texte ne sont pas altérées. Seul l'ordre des lettres est changé de façon à aboutir à un mélange sans cohérence.

> *En un mot, les méthodes de transposition sont une salade des lettres du texte clair.*
>
> Étienne Bazeries (1846-1931), cryptanalyste militaire français.

Le chiffrement *Rail fence*, utilisé pendant la guerre de Sécession, consiste en une transposition obtenue en écrivant un texte dans un tableau par colonnes, éventuellement en descendant et en montant successivement. Le cryptogramme est constitué en écrivant le texte en suivant les lignes.

2.1 La grille tournante

La grille tournante, ou grille de Fleissner, du nom du colonel autrichien Édouard Fleissner von Wostrovitz (1825-1888), qui l'a présentée en 1881 dans son ouvrage *Handbuch der Kryptrographie*, est un chiffrement par transposition. Ce procédé cryptographique est décrit dans les articles du cryptographe polytechnicien français Gaëtan Viaris de Lesegno (1847-1901), ainsi que dans le roman de Jules Verne *Mathias Sandorf* en 1885. Le cryptogramme est disposé dans un carré, et en plaçant sur celui-ci une grille ajourée comprenant des trous convenablement placés, les premières lettres du message en clair apparaissent. La suite du cryptogramme est lue de manière similaire en tournant successivement la grille d'un quart de tour (Fig. 1.4).

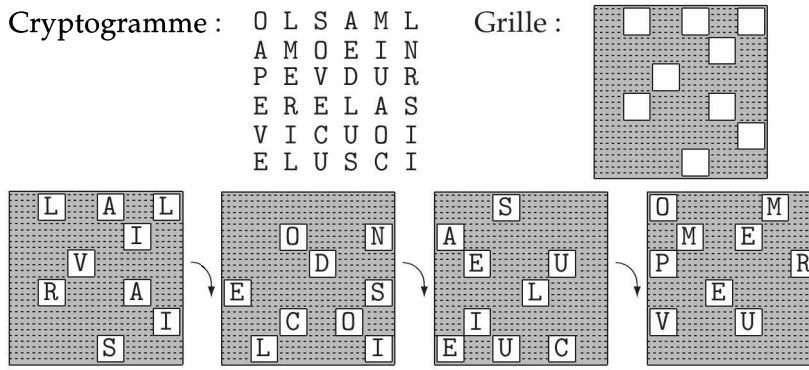


FIGURE 1.2. La grille tournante : la grille est placée sur le cryptogramme, puis est tournée quatre fois d'un quart de tour dans le sens des aiguilles d'une montre. Le message en clair apparaît dans les cases ajourées de la grille.

2.2 Le radiogramme de la victoire

Pendant la première guerre mondiale, l'armée allemande utilisait un système, appelé ADFGX, ainsi nommé car les cryptogrammes ne contenaient que ces cinq lettres. Les lettres ADFGX servent d'indice dans un tableau à double entrée pour définir l'une des 25 lettres de l'alphabet latin, le *v* et le *w* étant codés de la même manière. Ces lettres avaient été choisies en raison de leur code morse très différent, afin de limiter les erreurs de transmission des opérateurs télégraphistes de TSF. Cela permettait sans doute également de former plus rapidement ces opérateurs qui n'avaient que cinq lettres à apprendre. Ce code a ensuite été étendu à partir du 1er avril 1918 en un système ADFGVX, adjoignant la lettre V, ce qui permettait de coder les vingt-six lettres de l'alphabet et les dix chiffres.

Ce qui a été appelé par la suite le *radiogramme de la victoire*, est un message adressé par le haut commandement allemand vers une unité située au nord de Compiègne. Les services d'écoute français ont intercepté le 1er juin 1918 le message suivant :

```
FGAXA XAXFF FAFFA AVDFA GAXFX FAAAG DXGGX AGXFD XGAGX GAXGX
AGXVF VXXAG XFDAX GDAAF DGGAF FXGGX XDFAX GXAXV AGXGG DFAGD
GXVAX VFXGV FFGGA XDGAX ADVGG A
```

Ce message fut immédiatement transmis à la section du chiffre, puis décrypté dans la nuit du 2 au 3 juin 1918 par le capitaine George-Jean Painvin après un travail acharné. Il réussit à déterminer la transposition et la substitution qui

6	16	7	5	17	2	14	10	15	9	13	1	21	12	4	8	19	3	11	20	18
D	A	G	X	F	A	G	F	X	G	G	F	A	D	F	A	G	F	X	A	V
X	G	X	F	A	X	X	V	G	X	A	G	D	A	A	G	V	F	F	X	A
G	X	F	A	G	F	X	X	X	A	F	A	V	A	G	X	F	A	D	D	X
G	G	D	A	D	F	D	X	A	G	F	X	G	F	A	G	F	A	A	G	V
X	G	X	A	G	F	F	A	X	X	X	A	G	D	X	A	G	V	X	A	F
A	D	G	G	X	A	A	G	V	V	G	X	A	G	F	X	G	D	G	X	X

FIGURE 1.2. Radiogramme de la victoire : application de la transposition. La clé de transposition est une numérotation des colonnes. Le radiogramme intercepté est écrit verticalement dans les colonnes numérotées 1, 2... jusqu'à 21. La lecture horizontale dans le tableau donnera le clair après application de la substitution.

	A	**D**	**F**	**G**	**V**	**X**
A	c	o	8	x	f	4
D	m	k	3	a	z	9
F	n	w	l	0	j	d
G	5	s	i	y	h	u
V	p	1	v	b	6	r
X	e	q	7	t	2	g

FIGURE 1.6. Radiogramme de la victoire : application de la substitution. La clé de substitution est la façon de remplir le tableau avec les lettres et les chiffres. Les lettres ordonnées du radiogramme sont groupées par deux. La première est l'indice de ligne, la seconde est l'indice de colonne du tableau. DA $= m$, GX $= u$, FA $= n$, GF $= i$, XG $= t$, etc.

avaient servi au chiffrement pour reconstituer le message en clair (Fig. 1.5 et Fig. 1.6) :

> *Munitioneirung beschleunigen punkt soweit nicht eingesehen auch bei tag* (hâter l'approvisionnement en munitions, le faire même de jour tant qu'on n'est pas vu)

Ce texte, transmis au général Pétain, puis au général Foch, chef d'état-major interallié, a confirmé que l'offensive allemande allait se concentrer à cet endroit. Elle se produisit le 10 juin, mais l'information avait permis de prendre toute disposition pour la parer. Elle fut stoppée, ce qui fut décisif.

2 Les substitutions polygrammiques

Une substitution polygrammique est une substitution simple sur des groupements de plusieurs lettres. Par exemple, dans une substitution bigrammique, deux lettres du texte clair sont transformées en deux lettres du cryptogramme.

Cela rend plus difficile l'analyse des fréquences puisqu'il faut disposer cette fois de statistiques sur les groupements de deux lettres. L'étude est quand même possible. Il existe dans la langue française, près de 350 bigrammes, soit près de la moitié des 676 bigrammes possibles. L'autre moitié ne se présente jamais dans un texte. D'autre part, le bigramme le plus fréquent qui est *es* reste avec une fréquence d'apparition relativement faible, d'environ 3,05 % en moyenne.

3.1 Le chiffre de Playfair

Un chiffrement bigrammique général nécessite une table carrée de 26 lignes et 26 colonnes dans laquelle figurent les substitutions convenues. Il faut également établir la table réciproque qui n'est pas la même pour le déchiffrement. En raison de leur difficulté d'emploi et du risque d'erreur, on lui préfère la substitution par diagonale appelée *chiffre de Playfair*.

Il s'agit d'une invention de Charles Wheastone, datée du 26 mars 1854, mais c'est Lord Lyon Playfair (1818-1898), chimiste et politicien libéral écossais, promoteur de l'enseignement technique qui l'a popularisée. Ce chiffre utilise un carré de vingt-cinq cases rempli avec les lettres de l'alphabet. En français, on peut omettre le *w*, ou bien rendre équivalents le *i* et le *j*. Un procédé pour remplir ce carré est par exemple de convenir d'une phrase, d'écrire les lettres de cette phrase dans le carré sans les répéter, puis de compléter le carré avec les lettres manquantes dans l'ordre alphabétique. Ainsi, la phrase convenue, « *promenade cryptologique* », conduira au carré suivant :

P	R	O	M	E
N	A	D	C	Y
T	L	G	I	Q
U	B	F	H	J
K	S	V	X	Z

Deux lettres du texte clair sont transformées en deux lettres du cryptogramme selon leur disposition dans ce carré :

- les lettres doubles du texte clair sont éliminées en insérant entre elles une lettre rare comme par exemple le K ;
- si deux lettres sont sur une même ligne ou une même colonne, les lettres du cryptogramme sont les suivantes sur la ligne ou sur la colonne en convenant d'un sens, par exemple de haut en bas et de gauche à droite ;
- si les deux lettres forment les diagonales d'un rectangle, les lettres du cryptogramme sont les extrémités de l'autre diagonale en convenant d'un sens de rotation, par exemple dans le sens des aiguilles d'une montre.

Voici illustré le chiffrement du message « *l'attente est toujours longue* » :

```
clair :          LA TK TE NT EK ES TK TO UJ OU RS LO NG UE
cryptogramme :   BL UP PQ TU ZP ZR UP PG BU FP AR RG DT PJ
```

3.2 Le chiffre de Hill

Lester Hill a publié en 1929, alors qu'il était professeur assistant au Hunter College de New-York, un article intitulé *Cryptographie sur un alphabet algébrique* qui marque la première utilisation notable des mathématiques en cryptographie.

Tout d'abord, chacune des 26 lettres de l'alphabet est représentée par un nombre entier compris entre 0 et 25 selon un codage désordonné et tenu secret. Par exemple :

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

Ce codage des lettres revêt une grande importance. Maintenant que les lettres sont codées par des nombres, des opérations leur sont applicables. La cryptographie quitte le domaine du traitement de la langue écrite pour entrer dans celui du calcul. Sur ces nombres, les opérations d'addition et de multiplication sont réalisées modulo 26, ce qui signifie que, lorsque le résultat d'une opération dépasse 26, on le réduit en effectuant sa division par 26 et en ne conservant que le reste.

Encadré 1.1. Les opérations modulo n.

Il est onze heures. Le téléphone sonne.

– Allo ! Es-tu libre aujourd'hui ?

– Dans deux heures seulement.

– Alors on se voit à une heure ?

Cette conversation ne choquera personne. Pourtant à y regarder de près, nos correspondants se sont entendus sur l'addition $11 + 2 = 1$. C'est qu'ils ont effectué une addition modulo 12, très courante quand cela concerne des horaires.

Réaliser une réduction modulo n consiste à ramener la valeur dans l'intervalle $0, \ldots, n-1$ en soustrayant ou en ajoutant n le nombre convenable de fois. Par exemple, pour réduire 95 modulo 26, on soustrait successivement 26 jusqu'à atteindre un résultat entre 0 et 25 : $95 \rightarrow 69 \rightarrow 43 \rightarrow 17$. Si le nombre à réduire est négatif, on ajoute successivement 26 : $-32 \rightarrow -6 \rightarrow 21$. Il revient au même de diviser le nombre à réduire par n et de ne conserver que le reste.

Deux nombres qui ont la même réduction sont dits *congrus* modulo n. L'addition et la multiplication modulo n obéissent aux règles attendues pour ces opérations. Elles définissent une structure algébrique appelée *anneau des entiers modulo n*.

Comme les calculs sont effectués modulo 26, même s'ils sont très complexes, le résultat restera compris entre 0 et 25 et il sera possible de le transcrire en lettres avec le codage inverse de celui qui a transformé les lettres en nombres.

Le chiffrement de Hill appartient à la famille des chiffrements polygrammiques, car les lettres sont traitées par groupes. On présente ci-après un exemple où les lettres sont regroupées deux par deux. Leur codage représente des couples de nombres. Le chiffrement consiste à appliquer sur ces nombres un calcul linéaire. Les correspondants conviennent de quatre paramètres a, b, c, et d. Le couple de nombres (x,y) est transformé en un couple de nombres (u,v) en appliquant les formules suivantes :

$$\begin{cases} u = ax + by \\ v = cx + dy \end{cases} \tag{1.1}$$

Supposons que deux correspondants conviennent d'utiliser les paramètres $a = 9$, $b = 4$, $c = 5$ et $d = 7$, et que le message à transmettre soit `ALLO`. Le groupement des lettres deux par deux et l'utilisation de la table de correspondance ci-dessus conduit aux couples de nombres suivants :

$$\texttt{AL} : (5{,}16) \qquad \texttt{LO} : (16{,}3)$$

Ensuite, il faut appliquer la formule 1.1 avec les valeurs convenues des paramètres a, b, c et d :

$$\begin{cases} 9 \times 5 + 4 \times 16 = 109 \\ 5 \times 5 + 7 \times 16 = 137 \end{cases} \qquad \begin{cases} 9 \times 16 + 4 \times 3 = 156 \\ 5 \times 16 + 7 \times 3 = 101 \end{cases}$$

Le résultat est constitué de deux couples de nombres qu'il faut réduire modulo 26. On obtient alors des valeurs que la table de correspondance permet de convertir en lettres. Ces lettres constituent le cryptogramme :

$$(5{,}7) : \texttt{AN} \qquad (0{,}23) : \texttt{KB}$$

Le cryptogramme est donc `ANKB`. Pour le déchiffrer, le destinataire va effectuer les opérations inverses. Ces lettres sont groupées par deux puis traduites en vecteurs selon le même codage. Le récepteur utilise les formules qui expriment u et v en fonction de x et y. Les règles de résolution des systèmes d'équations linéaires conduisent à :

$$\begin{cases} u = a'x + b'y \\ v = c'x + d'y \end{cases}$$

où $a' = d/D$, $b' = -b/D$, $c' = -c/D$ et $d' = a/D$, avec $D = ad - bc$. Dans ces formules, la division par D s'entend modulo 26, ce qui revient à multiplier par l'inverse de D modulo 26.

Dans notre exemple, on a $D = 9 \times 7 - 4 \times 5 = 43$ qui est congru à 17 modulo 26. Comme indiqué dans l'encadré 1.2, son inverse modulo 26 vaut 23. Les coefficients de la formule inverse sont $a' = 5$, $b' = 12$, $c' = 15$ et $d' = 25$. Le destinataire code en nombre les lettres du cryptogramme qu'il a reçu, puis calcule :

$$\begin{cases} 5 \times 5 + 12 \times 7 = 109 \\ 15 \times 5 + 25 \times 7 = 250 \end{cases} \qquad \begin{cases} 5 \times 0 + 12 \times 23 = 276 \\ 15 \times 0 + 25 \times 23 = 575 \end{cases}$$

La réduction modulo 26 permet de retrouver les vecteurs initiaux, puis finalement le message en clair après décodage :

$$(5,16) : \texttt{AL} \qquad (16,3) : \texttt{LO}$$

Encadré 1.2. Inverse modulo n.

Tout comme l'inverse de 2 est 0,5 pour la raison que le produit $2 \times 0{,}5$ vaut 1, un nombre x sera dit *inversible* modulo n s'il en existe un autre y tel que le produit $x \times y$, réduit modulo n, est égal à 1.

Par exemple, $17 \times 23 = 391$ et $391 = 15 \times 26 + 1$. L'inverse de 17 modulo 26 est 23.

Si x et n n'ont pas de diviseur commun, le théorème de Bézout affirme qu'il existe deux entiers u et v, que l'on peut déterminer avec l'algorithme d'Euclide étendu, tels que $xu + nv = 1$. L'entier u est donc l'inverse de x modulo n.

Si x et n ont un diviseur commun, alors de tels entiers u et v n'existent pas, et l'entier x n'est pas inversible modulo n.

Dans l'exemple donné ci-dessus, les calculs sont effectués dans l'anneau des entiers modulo 26. Il n'y a que 26 éléments dans cette structure, et cela autorise un nombre illimité d'opérations sans avoir à craindre que les résultats obtenus ne deviennent gigantesques et impossibles à manipuler à force de calculs. C'est un atout pour le cryptographe qui peut imaginer des combinaisons d'opérations aussi complexes qu'il le souhaite pour contrarier son adversaire le cryptanalyste.

Toutes les structures algébriques finies sont utilisables à cette fin, comme les corps finis. Cette structure est plus riche car n'importe quel élément qui n'est pas zéro a un inverse, contrairement à l'anneau des entiers modulo 26, où les multiples de 2 et de 13 ne sont pas inversibles. Mais le nombre d'éléments d'un corps fini ne peut être que, soit un nombre premier, soit une puissance d'un nombre premier. Lester Hill a proposé d'utiliser ces structures pour enrichir les possibilités du cryptographe. Il écrit par exemple :

> *Si notre alphabet doit être converti en un corps fini, le mieux qui puisse être fait est d'omettre une lettre, disons j, pour obtenir un corps de 25 éléments, ou d'adjoindre un symbole additionnel pour obtenir un corps de 27 éléments.*

Le réel obstacle à l'utilisation pratique du chiffre de Hill est sa grande lourdeur d'emploi qui le rend très difficile à utiliser en l'absence de moyens de calcul mécanique destinés à éviter les erreurs. Hill a breveté un appareil, constitué de roues dentées reliées entre elles par une chaîne, qui peut chiffrer jusqu'à des hexagrammes, qui sont des blocs de six lettres. Selon David Khan, ce chiffre aurait servi au gouvernement des États-Unis pour chiffrer les trigrammes des indicatifs radio. Lester Hill, en y introduisant les mathématiques et les structures algébriques a eu un impact déterminant sur la cryptologie. Il a déclenché l'intérêt des mathématiciens pour ce domaine et marqué le début d'une nouvelle ère.

4 La genèse du polyalphabétisme

Un inconvénient majeur des substitutions simples est leur grande sensibilité à l'analyse des fréquences qui rend aisé leur décryptement. Les chiffrements homophoniques et les substitutions polygrammiques sont des tentatives pour contrer cette faiblesse. Mais le progrès essentiel a été le polyalphabétisme. Comme son nom l'indique, il met en œuvre plusieurs alphabets. Une même lettre du message sera codée par des lettres différentes dans le cryptogramme.

Le chiffre polyalphabétique fut inventé par l'italien de la Renaissance Léon Battista Alberti (1404, 1472), représentant, comme Léonard de Vinci, l'idéal de l'homme universel qui prévalait à cette époque. Il est plus connu comme architecte, mais il fut également peintre, compositeur, poète, organiste. Il est l'auteur d'une approche scientifique de la perspective et d'un traité sur la mouche domestique. Il a été introduit à la cryptologie par Leonardo Dato, secrétaire pontifical, qui a comparé le décryptement des secrets de la nature avec celui des lettres interceptées par les espions du pape. Dans le premier essai de cryptanalyse d'Occident, *De Componedis Cyphris*, il expose en 1466 une méthode de décryptement de textes écrits en latin, reposant sur le comptage des lettres, mais aussi sur la recherche des voyelles et des consonnes : « *sans voyelle, il n'y a pas de syllabe* ». Après avoir expliqué comment ces chiffrements peuvent être résolus, il a proposé un moyen d'y résister qu'il a qualifié d'incassable : son cadran chiffrant, représenté sur la figure 1.7.

Les deux correspondants doivent chacun disposer de cadrans identiques. Ils se mettent d'accord sur un index repéré par une lettre sur le disque mobile, par exemple la lettre *k*. Dans le cryptogramme, la première lettre écrite en majuscule, par exemple *B*, indiquera qu'il faut placer le *k* en face de cette lettre. Ensuite, chaque lettre du cryptogramme représentera la lettre fixée au-dessus d'elle. Après avoir écrit trois ou quatre lettres, la position de l'indice peut être changée de façon à ce que le *k* soit par exemple sous le *G*. Pour signifier ce changement, on

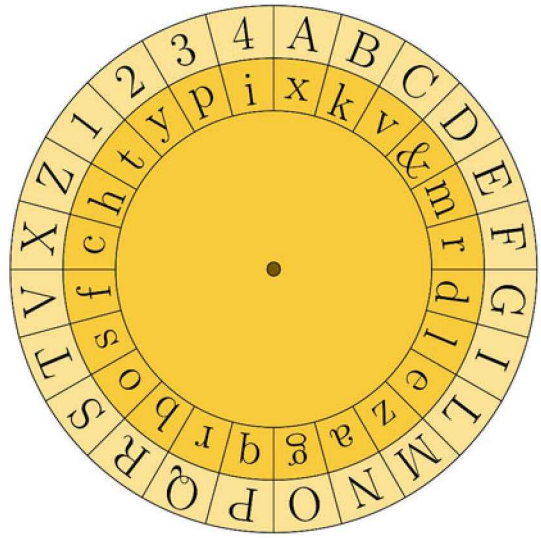


FIGURE 1.2. Le cadran d'Alberti est constitué d'un cadran fixe et d'un cadran mobile. Les lettres du cadran fixe sont écrites en majuscule et représentent les lettres du texte clair. Les lettres du cadran mobile sont écrites en minuscule et représentent les lettres du cryptogramme.

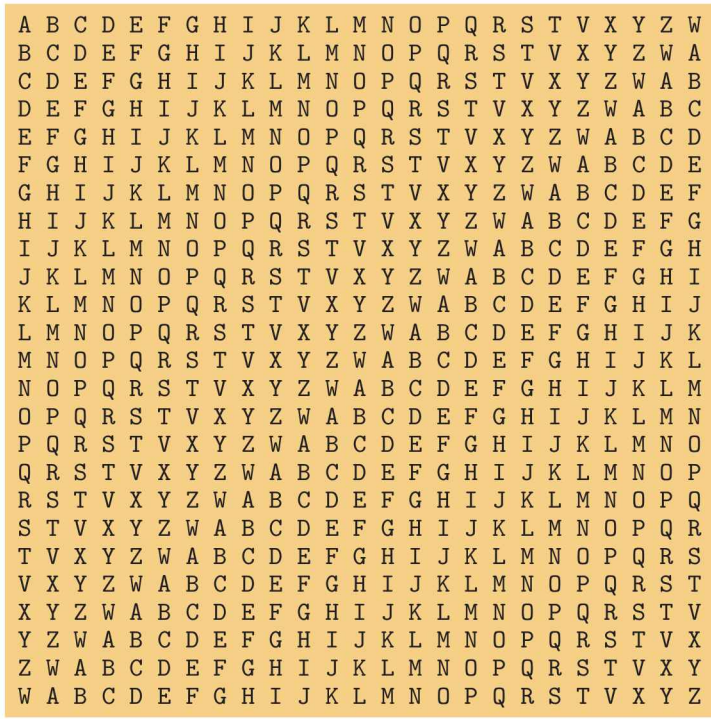
écrira un *G* majuscule dans le cryptogramme, et à partir là, *k* ne signifiera plus *B*, mais *G* et toutes les lettres du disque fixe auront de nouveaux équivalents. Voici un exemple de cryptogramme obtenu avec le cadran de la figure 1.7 (solution page 174) :

`BqxboGqvgiMteRkomcoyvXilya.`

Les chiffres du cadrant sont destinés à définir des entrées dans un répertoire pour signifier des mots courants. Ainsi, il peut être convenu que la suite 341 signifie *pape*, qui sera chiffrée d'une manière à un endroit du cryptogramme et d'une autre manière à un autre endroit.

Toute nouvelle position du disque conduit à un nouvel alphabet chiffrant, où la relation entre les lettres du message en clair et celles du cryptogramme a changé. Il y a autant d'alphabets que de positions possibles du disque. Le premier chiffre polyalphabétique était né. Le chiffre d'Alberti est encore imparfait. Sur de petites portions du cryptogramme, la substitution reste monoalphabétique et les lettres doubles du cryptogramme peuvent indiquer des lettres doubles du message en clair, comme dans *pappa* (pape).

La deuxième étape dans le développement du polyalphabétisme est due à l'abbé allemand Johannes Heidenbert, né en 1462 à Trittenheim, dit Jean Trithème. Il est connu en cryptographie pour être l'auteur en 1528 du premier grand



A B C D E F G H I J K L M N O P Q R S T V X Y Z W
B C D E F G H I J K L M N O P Q R S T V X Y Z W A
C D E F G H I J K L M N O P Q R S T V X Y Z W A B
D E F G H I J K L M N O P Q R S T V X Y Z W A B C
E F G H I J K L M N O P Q R S T V X Y Z W A B C D
F G H I J K L M N O P Q R S T V X Y Z W A B C D E
G H I J K L M N O P Q R S T V X Y Z W A B C D E F
H I J K L M N O P Q R S T V X Y Z W A B C D E F G
I J K L M N O P Q R S T V X Y Z W A B C D E F G H
J K L M N O P Q R S T V X Y Z W A B C D E F G H I
K L M N O P Q R S T V X Y Z W A B C D E F G H I J
L M N O P Q R S T V X Y Z W A B C D E F G H I J K
M N O P Q R S T V X Y Z W A B C D E F G H I J K L
N O P Q R S T V X Y Z W A B C D E F G H I J K L M
O P Q R S T V X Y Z W A B C D E F G H I J K L M N
P Q R S T V X Y Z W A B C D E F G H I J K L M N O
Q R S T V X Y Z W A B C D E F G H I J K L M N O P
R S T V X Y Z W A B C D E F G H I J K L M N O P Q
S T V X Y Z W A B C D E F G H I J K L M N O P Q R
T V X Y Z W A B C D E F G H I J K L M N O P Q R S
V X Y Z W A B C D E F G H I J K L M N O P Q R S T
X Y Z W A B C D E F G H I J K L M N O P Q R S T V
Y Z W A B C D E F G H I J K L M N O P Q R S T V X
Z W A B C D E F G H I J K L M N O P Q R S T V X Y
W A B C D E F G H I J K L M N O P Q R S T V X Y Z

FIGURE 1.6. La *tabula recta* de Trithème. Cette table contient toutes les manières de décaler l'alphabet. Le premier alphabet sert à coder la première lettre d'un texte, le deuxième alphabet code la deuxième lettre, etc.

ouvrage occidental de cryptologie : *Polygraphiae Libri Sex*, polygraphie en six livres, dont le livre cinq contient sa contribution majeure : la *tabula recta*, constituée sur chaque ligne des décalages de l'alphabet de l'époque (Fig. 1.8).

Le mode d'emploi de ce tableau selon Trithème est des plus simples : la première lettre est codée avec le premier alphabet, la seconde lettre avec le second alphabet, etc. Ainsi, le message `HVNC CAVEO VIRVM` (méfie toi de cet homme) sera-t-il codé en `HXPF GFBX DSCGW`. L'avantage sur le système d'Alberti est que l'alphabet de codage change à chaque lettre du texte, ce qui brouille considérablement l'analyse des fréquences. Par contre, il s'agit un procédé rigide et finalement assez pauvre. Dès qu'il vient à être connu, les cryptogrammes n'ont plus de secret.

L'italien Giovan Batista Belaso (1505- ?), est l'auteur en 1553 d'un petit ouvrage *La Cifra del Sig* qui va proposer une évolution majeure. Il a l'idée de réaliser le chiffrement en fonction d'un mot, facilement mémorisable et modifiable, qu'il appelle le *contresigne*.

Ce contresigne peut consister en quelques mots d'italien ou de latin, ou de n'importe quelle autre langue, et les mots peuvent être en nombre réduit ou important comme on veut.

Belaso, *La Cifra del Sig.*

Le contresigne est placé exactement au-dessus du texte clair et répété si besoin est. Chaque lettre définit l'alphabet de la *tabula recta* à utiliser pour chiffrer le texte clair. Ainsi, si le message à chiffrer est « *L'armata turchesca partira a cinque di luglio* » (l'armée turque se mettra en route le cinq juillet) avec le verset latin « *Virtuti omnia parent* » (tout cède à la vertu), en utilisant la *tabula recta* de la figure 1.8, le *u* et le *v* étant confondus, cela conduira à :

```
contresigne :    V IRTUTI OMNIAPARE NTVIRTU T IOMNIA PA RENTVI
clair :          L ARMATA TURCHESCA PARTIRA A CINQUE DI LUGLIO
cryptogramme :   G TJGUNI IHFKHTSTE DTMCALA T KYAEDE SI DWTFDY
```

Tous les ingrédients d'un chiffre très sûr figurent dans ce procédé. Il sera popularisé par Vigenère et on connaît ce procédé aujourd'hui sous le nom de *Chiffre de Vigenère*. Il sera réinventé plusieurs fois par la suite avec des variantes, par le Belge Grondsfeld en 1734, par l'Anglais Beaufort en 1810, par l'Anglais Twaite en 1854. Il sera longtemps considéré comme indécryptable. Il faudra attendre la fin du XIX^{e} siècle pour qu'apparaissent ses premières cryptanalyses.

Pourtant, il ne sera finalement que très peu utilisé, son inconvénient étant sa difficulté de mise en œuvre et sa grande sensibilité aux erreurs. L'usage du chiffrement homophonique et des répertoires de codes perdurera jusqu'aux années 1970. François de Callières, ambassadeur de Louis XIV, a écrit en 1716 un manuel devenu classique à l'usage des diplomates, *De la manière de négocier avec les Souverains*. Dans cet ouvrage, il classe la commodité d'emploi comme une propriété essentielle d'un bon chiffre.

On ne parle point de certains chiffres inventés par des régents de collège et faits sur des règles d'algèbre ou d'arithmétique, qui sont impraticables à cause de leur trop grande longueur et de leurs difficultés dans l'exécution, mais des chiffres communs dont se servent tous les négociateurs et dont on peut écrire une dépêche presque aussi vite qu'avec des lettres ordinaires.

L'utilisation du chiffrement polyalphabétique ne se généralisera qu'à partir du début du XX^{e} siècle, lorsque les machines à rotors réaliseront ce type de chiffrement de manière mécanique ou électro-mécanique, se prémunissant des erreurs de codage et de décodage.

5 Les machines à chiffrer

La difficulté de mettre en œuvre le chiffre polyalphabétique a conduit les cryptographes à imaginer très rapidement des outils pour faciliter les opérations de chiffrement et de déchiffrement. La réglette de Saint-Cyr (Fig. 1.9) est de ceux-là. Elle doit son nom à l'académie militaire française qui porte ce nom, où elle était en usage entre 1880 et le début du XX^{e} siècle.

FIGURE 1.9. La réglette de Saint-Cyr est constituée d'un coulisseau au centre que l'on déplace selon une loi convenue entre les lettres du texte clair en majuscule et celles, en minuscule, qui figureront dans le cryptogramme.

5.1 Le cylindre chiffrant de Jefferson

Le futur président des États-Unis d'Amérique, Thomas Jefferson (1743-1826), était écrivain, agriculteur, architecte, diplomate et homme politique. Alors qu'il était secrétaire d'État de George Washington, il mit au point un dispositif mécanique qu'il a appelé le *Wheel Cipher*, constitué de 26 disques sur la tranche desquels était imprimé un alphabet désordonné. Pour chiffrer un message, on fait tourner les roues de manière à faire apparaître le message.

Le cryptogramme est constitué de l'une quelconque des séquences des autres lettres. Pour déchiffrer, il suffit de disposer du même cylindre constitué des mêmes 26 disques, d'aligner le cryptogramme et de lire le seul texte qui semble avoir un sens parmi les autres alignements. On peut changer de clé en changeant l'ordre des cylindres.

Ce dispositif a lui-même été réinventé par Étienne Bazeries en 1891 (Fig. 1.10) et par le colonel italien Durcos en 1900. Une variante améliorée en 1917 par le colonel Joseph O. Mauborgne, chef du *Signal Corps*, le cylindre M-94, a été utilisé par l'armée américaine entre 1922 et 1942.

5.2 Les machines électro-mécaniques à rotor

Le chiffrement polyalphabétique ne sera vraiment utilisé qu'au début du XX^{e} siècle avec l'apparition des machines électro-mécaniques à rotor. Presque simultanément, elles ont été présentées par quatre inventeurs de pays différents. L'Américain Edward Hugh Hebern (1869-1952) dépose un brevet en 1918 pour

FIGURE 1.10. Le cylindre chiffrant de Bazerie (Espace Ferrié). En 1891, Étienne Bazerie a réinventé le cylindre chiffrant, auparavant mis au point par le futur président des États-Unis d'Amérique, Thomas Jefferson, autour de 1793.

proposer sa machine à l'armée américaine qui refuse l'offre pour des raisons de vulnérabilité. Le Hollandais Hugo Alexsander Koch (1870-1928) et le Suédois Arvid Damm (?-1927) déposent chacun un brevet en 1919. L'industriel suédois Boris Hagelin (1925-1983) fonde alors la société *Aktiebolaget Cryptograph* pour exploiter le brevet d'Arvid Damm à partir de 1925. Cette société deviendra la société suisse *Crypto AG* encore en activité aujourd'hui, qui équipera de nombreuses armées occidentales, dont l'armée française.

La machine de ce type la plus connue est sans conteste la machine *Enigma*, dont l'allemand Arthur Scherbius (1878-1929) dépose le brevet en 1918. Il fonde la société *Chiffriermaschinen* en 1923 pour commercialiser sa machine qu'il propose d'abord aux milieux financiers et aux banques qui ne l'adopteront pas. C'est l'armée allemande, consciente de la faiblesse du chiffrement ADFGVX, qui verra l'intérêt de cette nouvelle machine qui allie la force du chiffrement polyalphabétique et une grande simplicité d'utilisation. L'utilisateur actionne une touche sur un clavier, et une lampe s'allume, indiquant la lettre correspondante du cryptogramme. La *Reichmarine* l'adoptera en 1926, la *Reichwehr* en 1928 et enfin la *Luftwaffe* en 1935. La doctrine militaire allemande de cette époque, la *blitzkrieg*

FIGURE 1.11. La machine Enigma : cette vue ouverte de la machine Enigma fait apparaître les trois rotors ainsi que les lampes qui indiquent la lettre qui se substitue à celle qui est actionnée sur le clavier.

(guerre éclair), est une guerre de mouvement qui s'appuie sur l'effet de surprise d'une attaque rapide et synchronisée des différentes forces : infanterie, unités mécanisées et aviation. Les communications par radio en sont un élément essentiel, ainsi que leur chiffrement en raison de la dispersion des ondes électromagnétiques qui les rend par nature sensibles à l'interception par l'ennemi. L'armée allemande se dotera d'un important système de communications chiffrées entre les différentes unités et leur commandement.

Principe de fonctionnement

Le cœur des machines électro-mécaniques sont les rotors, qui sont des cylindres rotatifs bordés de 26 contacts représentant chacun une lettre de l'alphabet. Ils sont traversés par des fils électriques qui réalisent une permutation entre les contacts de chaque bord. Plusieurs rotors sont mis en série pour multiplier les permutations ainsi composées. Un réflecteur compose les substitutions avec les substitutions réciproques, rendant l'opération de chiffrement identique à l'opération de déchiffrement, ce qui évite d'avoir à inverser l'ordre des rotors selon que l'on chiffre ou que l'on déchiffre.

La clé de chiffrement est constituée du choix des rotors, de leur position initiale ainsi que d'un branchement défini sur un tableau de fiches situé à l'avant de la machine qui réalise une permutation initiale. À chaque lettre du message, les rotors tournent à la manière d'un compteur, changeant ainsi la substitution qui s'opère sur l'alphabet, comme cela est illustré sur la figure 1.12. Le nombre de substitutions que ce système rend possible est considérable. Le paragraphe 4.4 page 94 décrit comment les cryptanalystes polonais et anglais sont venus à bout des messages chiffrés avec cette machine.

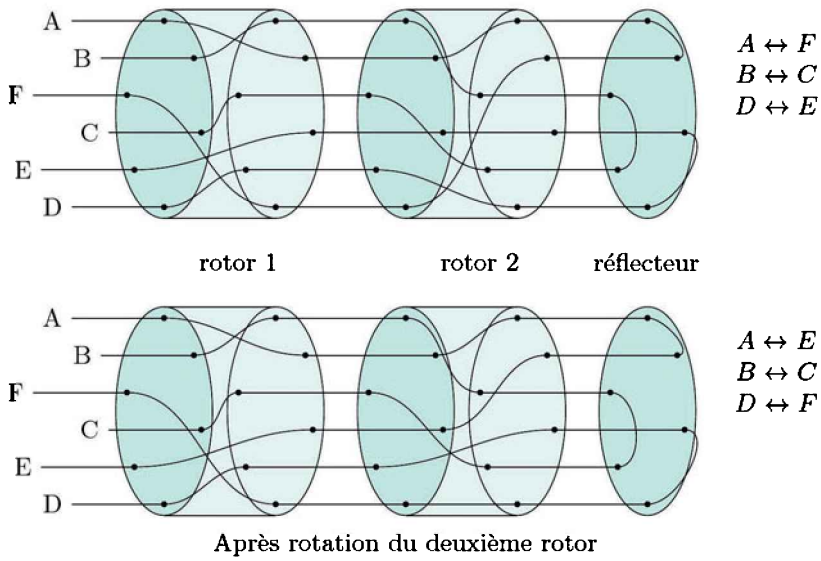


FIGURE 1.12. Principe de fonctionnement des machines à rotor illustré sur une machine à deux rotors sur l'alphabet ABCDEF. Chaque lettre de texte provoque la rotation des rotors, ce qui change à chaque fois la permutation opérée.

5 La stéganographie

La stéganographie n'est pas à proprement parler un procédé cryptographique. Le texte existe dans sa forme claire, mais il est noyé dans d'autres informations qui le rendent invisible. Elle a été utilisée par les espions en temps de guerre pour camoufler des messages stratégiques importants sous l'apparence de nouvelles anodines. À l'image de ce câble suivant a été transmis par un espion allemand lors du premier conflit mondial :

President's embargo ruling should have immediate notice, grave situation affecting international laws. Statement fore-shadows ruin of many

neutral. Yellow journals unifying national excitement immensely. (La décision d'embargo du président devrait avoir effet immédiat. Situation sérieuse mettant en cause les lois internationales. Cette déclaration présage la ruine de nombreux pays neutres. La presse à sensation unifie énormément le sentiment national.)

Le message caché apparaît en prenant la première lettre de chaque mot : *Pershing sails from NY june 1* (Pershing embarque de NY le 1er juin). John Pershing est le général commandant le corps expéditionnaire américain en Europe (AEF *American Expeditionnary Force*) suite à l'entrée en guerre des États-Unis d'Amérique contre l'Empire allemand de Guillaume II le 6 avril 1917. Un second message a confirmé l'information :

Apparently neutral's protest is thoroughly discounted and ignored. Ismam hard hit. Blockade issue affects pretext for embargo on by-products, ejecting suets and vegetable oils. (Apparemment la protestation des pays neutres est totalement écartée et ignorée. Ismam frappe fort. L'issue du blocus donne le prétexte pour un embargo sur certains produits, sauf suifs et huiles végétales)

Cette fois, c'est la deuxième lettre de chaque mot qui dévoile le même message. Contrairement à ce que semble avoir appris notre espion, Pershing a en fait quitté New-York le 28 mai 1917.

Des techniques du ressort de la stéganographie sont toujours utilisées aujourd'hui pour dissimuler des informations dans les contenus multimédia et tenter de lutter contre les copies illégales de musique ou de vidéo.

2

La cryptographie symétrique moderne

Ce chapitre présente les systèmes de chiffrement symétriques modernes, appelés ainsi car ils reposent sur le partage de façon symétrique de la même clé secrète par les deux correspondants. Après une introduction historique, on y expose les propriétés d'un système de confidentialité, qui ont été pour la première fois énoncées par Shannon à la suite des travaux qu'il a effectués pendant le second conflit mondial. Les deux principaux procédés que sont le chiffrement à flot et le chiffrement par bloc sont présentés. Le chapitre se termine par la description des principaux standards de chiffrement actuels que sont le DES et AES, ainsi que leurs modes d'utilisation.

1 La naissance de la cryptographie moderne

La révolution industrielle a vu se multiplier les échanges dès la fin du XVIII^{e} siècle et dans le courant du XIX^{e} siècle. Des réseaux de communication se mettent en place et se développent : le chemin de fer, les canaux de navigation, mais aussi le réseau télégraphique. Avec le déploiement du télégraphe de Claude Chappe (1763-1879) à partir de 1794, puis du télégraphe électrique (1845) et du téléphone (1879), on assiste à un changement d'échelle dans la quantité des messages échangés. Le développement de la cryptologie accompagne celui de ces réseaux. L'extension du télégraphe va conduire à de nouvelles exigences de confidentialité.

Des mesures devront être prises pour parer à une sérieuse objection que l'on soulève à propos des communications privées par télégraphe – la violation du secret – car, dans tous les cas, une demi-douzaine de personnes sont amenées à connaître chaque mot adressé par une personne à une autre. Les employés de la Compagnie anglaise du télégraphe s'engagent au secret sous serment, mais nous écrivons souvent des choses que nous ne supporterions pas de voir lues par d'autres avant nous. C'est encore un grave défaut du télégraphe, et il faut y remédier d'une manière ou d'une autre.

Quaterly Review, 1853

En outre, le recours à la télégraphie a imposé de revoir les méthodes de chiffrement. L'emploi d'un code alphabétique comme le code morse interdit l'utilisation de symboles graphiques dont l'usage était pourtant répandu dans la cryptographie traditionnelle. L'invention d'un chiffre compatible avec le télégraphe s'imposait. Avec le développement de la télégraphie sans fil (TSF), le renseignement par écoute va prendre une dimension stratégique. Le besoin de chiffrer les communications deviendra manifeste.

C'est dans ce contexte, et à la suite de la défaite militaire française contre la Prusse en 1870, qu'en 1883, le linguiste français d'origine hollandaise, Auguste Kerckhoffs (1835-1901), écrit un article, *La cryptographie militaire*, qui aura une grande portée dans le monde de la cryptologie et s'imposera rapidement comme une référence. Kerckhoffs y écrit en particulier :

L'Administration doit absolument renoncer aux méthodes secrètes, et établir en principe qu'elle n'acceptera qu'un procédé qui puisse être enseigné au grand jour dans nos écoles militaires, que nos élèves seront libres de communiquer à qui leur plaira, et que nos voisins pourront même copier et adopter, si cela leur convient. Je dirai plus : ce ne sera que lorsque nos officiers auront étudié les principes de la cryptographie et appris l'art de déchiffrer, qu'ils seront en état d'éviter les nombreuses bévues qui compromettent la clef des meilleurs chiffres, et auxquelles sont nécessairement exposés tous les profanes.

Dans une section nommée *Desiderata de la cryptographie militaire*, il expose les caractéristiques du chiffrement dans un système de communication. Ces caractéristiques requises sont aujourd'hui connues sous le nom de *principe de Kerckhoffs*.

Il faut bien distinguer entre un système d'écriture chiffrée, imaginé pour un échange momentané de lettres entre quelques personnes isolées, et une méthode de cryptographie destinée à régler pour un temps illimité la correspondance des différents chefs d'armée entre eux. Ceux-ci, en effet, ne

peuvent à leur gré et à un moment donné, modifier leurs conventions ; de plus, ils ne doivent jamais garder sur eux aucun objet ou écrit qui soit de nature à éclairer l'ennemi sur le sens des dépêches secrètes qui pourraient tomber entre ses mains. Un grand nombre de combinaisons ingénieuses peuvent répondre au but qu'on veut atteindre dans le premier cas ; dans le second, il faut un système remplissant certaines conditions exceptionnelles, conditions que je résumerai sous les six chefs suivants :

1. *le système doit être matériellement, sinon mathématiquement, indéchiffrable ;*
2. *il faut qu'il n'exige pas le secret, et qu'il puisse sans inconvénient tomber entre les mains de l'ennemi ;*
3. *la clef doit pouvoir en être communiquée et retenue sans le secours de notes écrites, et être changée ou modifiée au gré des correspondants ;*
4. *il faut qu'il soit applicable à la correspondance télégraphique ;*
5. *il faut qu'il soit portatif, et que son maniement ou son fonctionnement n'exige pas le concours de plusieurs personnes ;*
6. *enfin, il est nécessaire, vu les circonstances qui en commandent l'application, que le système soit d'un usage facile, ne demandant ni tension d'esprit, ni la connaissance d'une longue série de règles à observer.*

La cryptographie doit désormais assurer la sécurité du système de communication, et non plus seulement garantir la discrétion des échanges entre des acteurs particuliers comme Alice et Bob. Elle ne doit pas compliquer la tâche de ceux qui l'utilisent, comme le faisait déjà remarquer François de Caillères en 1716 (voir page 15). Et surtout, la sécurité d'un chiffre ne doit pas reposer sur le secret du procédé, mais seulement sur une clé, facilement mémorisable et modifiable. Il est au contraire largement admis qu'un procédé public, soumis à l'analyse acharnée d'une communauté ouverte de cryptanalystes, et auxquels il finit par résister, sera réputé plus sûr qu'un procédé privé et confidentiel, donc peu étudié, et qui ne manquera pas d'être dévoilé un jour ou l'autre. La publicité du procédé est un argument de sécurité.

2 Les systèmes de confidentialité

Un système de confidentialité est un ensemble de protocoles et d'algorithmes qui permettent à deux protagonistes d'échanger des informations avec discrétion, sans crainte d'être écoutés, y compris lorsqu'ils échangent sur un canal public accessible à tous comme les ondes radio ou le réseau internet. Ceux-ci se mettent

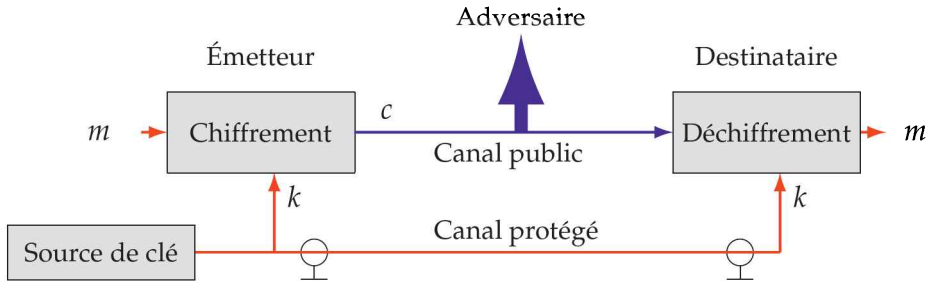


FIGURE 2.1. Schéma d'un système de confidentialité. Les protagonistes échangent préalablement une clé sur un canal protégé. Grâce à cette clé, le chiffrement transforme le message m en un cryptogramme c, incompréhensible pour l'adversaire. Le destinataire utilise la même clé pour déchiffrer le cryptogramme et retrouver le message. Les données sensibles sont celles qui transitent sur les liaisons rouges. Les données qui transitent sur les liaisons bleues sont protégées et peuvent être sans dommage observées.

préalablement d'accord sur une clé secrète et sur le procédé de transformation des messages. Le procédé est *a priori* connu de tous. Avant l'invention des mécanismes à clé publique dans les années 1970, les procédés symétriques étaient les seuls connus. Ils peuvent assurer la confidentialité et l'authentification entre des personnes qui se font confiance et ils sont toujours largement employés en raison de leur rapidité d'exécution.

La figure 2.1 montre les trois acteurs du système : l'émetteur, le destinataire et l'adversaire. Un mécanisme de chiffrement permet à l'émetteur de transformer un message en clair en un cryptogramme à l'aide d'une clé. À l'autre bout, avec la même clé, le déchiffrement permet au destinataire de reconstituer le message en clair à partir du cryptogramme. Le procédé de chiffrement peut être *probabiliste*, ce qui signifie qu'un même message peut ne pas toujours donner le même cryptogramme. Par contre, le déchiffrement est impérativement déterministe. Quel que soit le cryptogramme produit, le déchiffrement doit toujours conduire au message en clair dont il est issu.

La clé est une donnée particulièrement sensible. C'est sur son secret que repose la confidentialité des messages. Si elle venait à être divulguée, l'adversaire aurait sans problème accès au message. Le transport sécurisé des clés est un élément crucial pour la sécurité du chiffrement symétrique. Sur la figure 2.1, cela est schématisé par un câble blindé qui illustre que la clé n'est connue que par les deux protagonistes. C'est précisément pour résoudre ce problème que les systèmes à clé publique, présentés au chapitre suivant, ont été élaborés.

5 Diffusion et confusion

La conception d'un mécanisme de transformation du message en clair en un cryptogramme qui devra rester d'une obscurité totale à qui l'observe relève d'un

art que les cryptologues ont élaboré en s'appuyant sur leur longue expérience. Claude Shannon a énoncé les principes de diffusion et de confusion auxquels ils doivent obéir. Ils sont un guide pour le cryptologue dans son délicat travail de conception.

3.1 Le principe de diffusion

Le principe de diffusion énonce que les statistiques du texte clair doivent se propager sur l'ensemble de tout le cryptogramme, obligeant le cryptanalyste à intercepter et analyser une très grande quantité de données pour conduire son décryptement. Par exemple, la grande fréquence de la lettre *e* dans la langue française ne doit pas avoir un effet réduit à quelques symboles seulement du cryptogramme, mais au contraire être diffusée autant que possible à la totalité du cryptogramme. Ainsi, le cryptanalyste aura la plus grande difficulté à en tirer la moindre déduction. Les procédés traditionnels de substitution simple ne satisfont pas ce principe car les propriétés statistiques du texte clair n'y ont qu'un effet local sur le cryptogramme. La fréquence d'une lettre du texte clair vaut exactement la fréquence de la lettre qui lui correspond dans le cryptogramme. Les procédés bigrammiques souffrent, à un degré moindre, du même inconvénient : les statistiques des bigrammes du texte clair se retrouvent sur les bigrammes du cryptogramme. Le chiffre polyalphabétique avec clé répétée cycliquement ne diffuse les statistiques que sur une longueur du cryptogramme égale à la taille du mot clé.

> *Le respect du principe de diffusion doit conduire de façon idéale à un cryptogramme statistiquement indiscernable d'une suite aléatoire de symboles.*

3.2 Le principe de confusion

Le principe de confusion énonce que les relations entre le message en clair et le cryptogramme doivent être complexes. La mise en équation du procédé doit aboutir à de très gros systèmes d'équations, où toutes les variables dépendent de toutes les autres en des relations si confuses que le travail de résolution est pratiquement impossible. Le procédé ne doit pas pouvoir se modéliser simplement. En ce sens, le chiffre de Hill, en opérant par des transformations linéaires, ne satisfait pas ce principe. Le respect du principe de confusion impose l'utilisation de relations hautement non linéaires.

En première analyse, si le concepteur s'appuie sur une structure algébrique pour concevoir son procédé de chiffrement, la diffusion est assurée par les fonctions linéaires et la confusion par des fonctions non linéaires, par exemple de degré élevé. Tout l'art du concepteur est de jongler avec équilibre entre linéaire et non linéaire.

4 Le chiffrement à flot

Les procédés de chiffrement symétriques se classent en deux grandes familles : le chiffrement à flot, objet du présent paragraphe et le chiffrement par bloc, décrit dans le paragraphe suivant. Le *chiffrement à flot* tire son nom de l'anglais *stream cipher*. Il est appelé aussi *chiffrement à la volée*. Il consiste à combiner chaque symbole du message en clair avec un symbole d'une séquence pseudo-aléatoire, reproduite de manière identique par l'émetteur et le destinataire, et qui a toute l'apparence d'une suite erratique et désordonnée.

4.1 Le système de Vernam

Cette méthode tire son origine d'un brevet, déposé en 1917 par l'ingénieur en télécommunications américain Gilbert Vernam. Il était alors chargé de la sécurité des téléscripteurs dans la section *telegraph* de la société AT&T (*American Telephone & Telegraph*).

Un téléscripteur est un appareil qui permet de transmettre et d'imprimer un texte à l'aide d'un mécanisme de machine à écrire. Le texte est saisi sur un clavier par un opérateur, puis mémorisé et transporté sous forme de ruban perforé. Chaque caractère est codé par un ensemble de cinq marques, matérialisées par la présence ou l'absence d'un trou sur le ruban de papier. Un système de cinq aiguilles aimantées permet de traduire ce code par le passage ou non d'un courant électrique. Une organisation des cinq marques peut représenter une lettre ou un chiffre selon un codage appelé *code Baudot*, en hommage à Émile Baudot (1845-1903) qui l'a inventé en 1874. Un code spécial active le mode chiffre, et un autre active le mode lettre.

Vernam va mettre au point une technique permettant d'utiliser directement le téléscripteur pour chiffrer et déchiffrer les messages sans intervention humaine. Selon les termes de son brevet :

> *Les messages peuvent être transmis et reçus en clair, ou codés de manière connue, mais les impulsions du signal sont si altérées avant leur transmission sur la ligne qu'elles sont inintelligibles à quiconque les intercepte.*

Ainsi, l'opérateur n'a plus à se préoccuper du chiffrement qui se trouve intégré dans la chaîne de transmission. Le système de Vernam repose sur l'utilisation d'un second ruban perforé en guise de clé. Chaque marque du message est combinée avec la marque située à la même position dans le ruban de clé. En désignant par • la présence d'une marque et par son absence, la combinaison est définie par les règles suivantes :

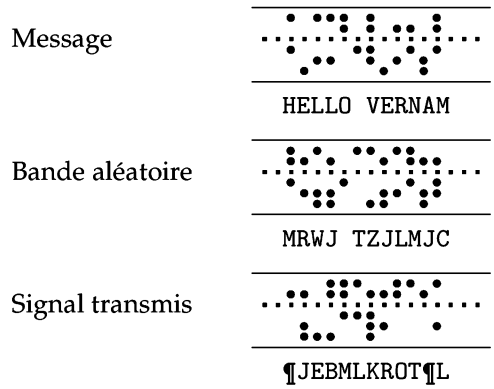
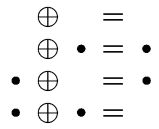


FIGURE 1.9. Le système de Vernam. La première bande perforée contient le message en clair. Les signaux sont combinés avec ceux d'une deuxième bande perforée contenant des caractères aléatoires. Le résultat de la combinaison est un signal chiffré, illustré ici par une troisième bande. Ce signal est transmis par le système télégraphique. À la réception, une bande perforée identique à la bande aléatoire utilisée à l'émission permet de reconstituer le message en clair à partir du signal reçu.



Cette opération a ceci de particulier que l'opération réciproque a exactement la même table. Avec cette combinaison, le chiffrement est identique au déchiffrement, pourvu qu'on utilise le même ruban. Il n'y a donc qu'une seule réalisation électromécanique à prévoir. Le message en clair sera obtenu en combinant de manière identique la clé au cryptogramme.

Ce système sera rapidement adopté par AT&T, mais aussi par l'armée américaine dès 1918. Le chef du *Signal Corps*, Joseph Mauborgne, énoncera que la seule clé sûre est une clé aléatoire de même longueur que le message lui-même. Il faudra attendre le développement de la théorie de l'information par Shannon en 1948 pour avoir une preuve de cette assertion.

Une bande aléatoire ne doit absolument pas être utilisée plus d'une seule fois, sous peine de compromettre tous les messages qu'elle a chiffrés. Ce qui va faire obstacle au déploiement de ce système est l'énorme quantité de bandes perforées à transmettre préalablement pour servir de clé. Il sera réservé aux communications très sensibles, présentant un haut degré de confidentialité, comme par exemple ce qu'on a appelé le *téléphone rouge* – qui était en fait un téléscripteur – mis en place entre les présidences américaines et soviétiques après la crise des missiles de Cuba de 1962. Les bandes aléatoires étaient alors transmises par

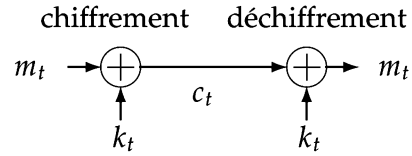


FIGURE 1.9. Schéma d'un chiffrement à flot. À chaque instant t, le symbole c_t du cryptogramme est obtenu par addition du symbole du message en clair m_t avec un symbole d'une suite chiffrante k_t que les deux correspondants savent reproduire. L'opération de déchiffrement est la même que l'opération de chiffrement.

la valise diplomatique, puis détruites après chaque utilisation. Les Allemands vont développer les machines de Lorenz, SZ40 et SZ42, utilisées pendant la seconde guerre mondiale, reposant sur un principe similaire, mais où les caractères chiffrants sont calculés à la volée à partir d'un procédé électromécanique de génération pseudo-aléatoire.

Aujourd'hui, dans un système de chiffrement à flot, le message en clair consiste en une suite de symboles binaires transmis un à un, comme un flot de données (Fig. 2.3). À chaque instant t, le symbole m_t du message en clair est additionné modulo 2 avec un terme d'une suite chiffrante k_t que les deux correspondants sont chacun capables de reproduire par un calcul à partir de leur clé secrète. Le cryptogramme est le résultat du calcul $c_t = m_t + k_t$ modulo 2. Si le symbole de la suite chiffrante vaut 0, le symbole de cryptogramme est le même que le symbole de clair et, s'il vaut 1, le symbole de cryptogramme est inversé par rapport au clair. La suite chiffrante masque toute information du message en clair en changeant chaque symbole de manière erratique. Par exemple :

Message : $m =$ `00110001010011101 0011`
Suite chiffrante : $k =$ `11010111001001111 0010`
Cryptogramme : $c =$ `11100110011010010 0001`

Au déchiffrement, le message en clair est retrouvé par la même opération en échangeant les rôles des symboles du message en clair et du cryptogramme : $m_t = c_t + k_t$ modulo 2.

4.2 Masque jetable

Les systèmes de chiffrement à flot sont classés selon la manière dont la suite chiffrante est produite. S'il s'agit d'une séquence de symboles binaires parfaitement aléatoire et utilisée une seule fois, on parle de *masque jetable* (*one time pad*). Elle peut être produite par un générateur physique d'aléa, reposant par exemple sur l'échantillonnage du bruit thermique engendré par un composant électronique. Ainsi, pour n'importe quel cryptogramme, tous les messages en clair sont également probables. Si le cryptogramme est `0101110101`, il est tout

aussi probable qu'il provienne du message en clair 1010110010 avec la suite chiffrante 1111000111 que du message en clair 0100111110 avec la suite chiffrante 00001001011, ou d'ailleurs de n'importe quel autre message en clair. Le cryptogramme n'apporte rigoureusement aucune information sur le contenu du message en clair. Ce chiffrement atteint la *sécurité inconditionnelle*. Un adversaire, même tout-puissant, n'a pas d'autre moyen de décrypter que de choisir un clair au hasard et de compter sur sa chance. Mais les correspondants doivent échanger de façon sûre une très grande quantité de symboles aléatoires, ce qui complique sa mise en œuvre.

4.3 Masque pseudo-aléatoire

Le masque jetable peut être simulé avec un générateur pseudo-aléatoire. À partir d'une clé de taille réduite, par exemple de 128 symboles binaires, le générateur pseudo-aléatoire produit par le calcul une longue séquence de symboles qui a toute l'apparence d'une suite aléatoire, totalement imprévisible pour quiconque ignore la clé, y compris s'il connaît les premiers termes. Les deux correspondants sont capables de calculer la même séquence à partir de la clé secrète qu'ils partagent.

Produire par le calcul des séquences de symboles qui semblent être le résultat du hasard n'est pas aussi facile qu'on pourrait le penser de prime abord. Une méthode possible consiste à utiliser un *automate*, comme illustré sur la figure 2.4. Il s'agit d'un dispositif constitué de :

- une mémoire qui contient un état, appelé *état interne* ;
- une *fonction de transition* qui détermine la façon dont évolue l'état interne au fil des itérations ;
- une *fonction de sortie* qui extrait une partie de l'information de l'état interne pour élaborer le symbole produit.

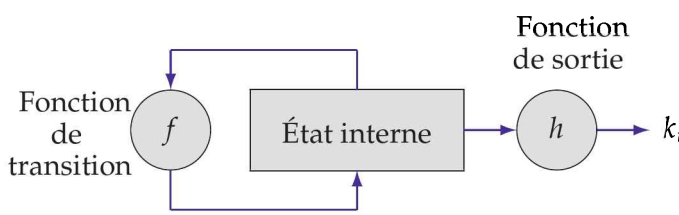


FIGURE 1.9. Automate générateur de pseudo-aléa. La fonction de transition fait évoluer l'état interne d'une itération à l'autre. La fonction de sortie détermine le symbole produit à partir de l'état interne à l'instant courant.

Pour une utilisation cryptographique, la clé secrète réside dans l'état initial de l'automate, dans la fonction de transition, ou dans la fonction de sortie. Pour interdire que plusieurs messages ne soient chiffrés avec la même séquence pseudo-aléatoire, il est nécessaire d'introduire une *clé de message*, un élément additionnel qui va diversifier la suite produite à chaque message. Cet élément est en général transmis en tête du message.

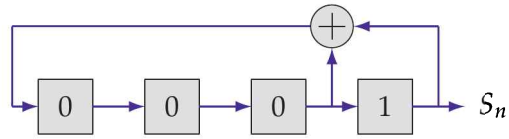
4.4 La période

La mémoire d'un automate est finie. Si elle est capable de mémoriser n symboles binaires, le nombre d'états internes différents vaut 2^n. Cette quantité peut certes atteindre de très grandes valeurs, mais elle reste finie. Quelle que soit la façon dont l'état interne évolue, celui-ci revient immanquablement vers une valeur déjà atteinte auparavant. À partir de ce moment, les mêmes calculs sont reproduits et les mêmes symboles générés. La période est le temps au bout duquel les mêmes symboles sont systématiquement répétés. Elle doit être assez grande pour ne pas risquer de reproduire la même suite chiffrante. Le message en clair serait alors chiffré plusieurs fois de la même façon, en contradiction avec le principe de diffusion. Si le début du message est connu, comme la date ou une formule de politesse, le cryptanalyste peut alors avoir accès à d'autres portions du message plus sensibles. Il est primordial que la période dépasse largement la taille des messages à chiffrer.

L'algèbre permet de construire des automates de grande période. Considérons un nombre entier m assez grand. La période des automates dont les états internes sont les entiers inférieurs à m et dont la transition d'un état à l'état suivant se fait selon une fonction affine $f(x) = ax + b$ modulo m dépend des valeurs de a et b, mais est assez facilement calculable. On peut même choisir les entiers a et b pour atteindre la période maximale, égale à $m - 1$. Les générateurs pseudo-aléatoires de ce type s'appellent des *générateurs congruentiels*. Ils réalisent la plupart des fonctions de génération aléatoire des ordinateurs. Par exemple, le générateur dont l'état interne est un entier modulo 17 initialisé à $x = 1$, et dont la transition est définie par la fonction $x \longmapsto 3x + 1 \mod 17$, produit la séquence suivante qui est de période 16 :

$$1, 4, 13, 6, 2, 7, 5, 16, 15, 12, 3, 10, 14, 9, 11, 0, 1, 4 \ldots$$

La fonction de transition d'un générateur congruentiel comprend une multiplication, qui est une opération assez complexe à réaliser. Pour cette raison, on préfère opérer dans l'algèbre des polynômes, qui a des propriétés similaires, mais dont la réalisation en circuits électroniques s'avère bien plus simple.



$$P(X) = 1 + X + X^4$$

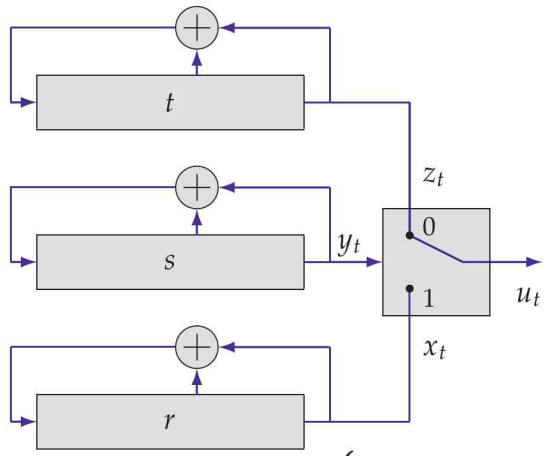
$$S = \underbrace{100010011010111}_{\text{suite de période 15}}1000100\cdots$$

FIGURE 2.5. Registre à décalage rebouclé linéairement (LFSR, *Linear Feedback Shift Register*). L'état interne est constitué de n cellules, ici $n = 4$, chacune mémorisant une donnée binaire. Elles sont décalées sur la droite à chaque itération. Le rebouclage est une fonction linéaire de l'état, c'est-à-dire la somme modulo 2 des contenus de certaines cellules. Il est représenté par un polynôme. Lorsque ce polynôme est bien choisi, et lorsque l'état initial n'est pas entièrement à zéro, la période peut aller jusqu'à $2^n - 1$, pour un registre de n cellules. Un polynôme qui réalise cette période maximale s'appelle un *polynôme primitif*.

La figure 2.5 représente ce qu'on appelle un *registre à décalage rebouclé linéairement*. Les statistiques des suites produites par de tels dispositifs sont très bonnes, car si le registre est bien construit, toutes les figures de la taille n du registre apparaissent exactement une fois au cours de la période, sauf la figure de n zéros qui correspond à un état stable. Malheureusement, ces suites sont très faciles à repérer, et même si l'on ignore les connexions du rebouclage, il suffit de connaître un nombre de termes égal à deux fois sa taille pour reconstituer à la fois l'état initial et le schéma des connexions du registre.

4.5 Comment rendre les suites plus complexes ?

Étant donné une suite binaire périodique, on peut toujours construire un registre linéaire à décalage qui la produit. La taille du plus petit registre linéaire capable de la produire s'appelle sa *complexité linéaire*. Cette quantité mesure en quelque sorte l'effort à fournir pour reconstituer la suite chiffrante en la synthétisant par un registre linéaire. Le concepteur cherchera à rendre sa valeur la plus élevée possible. La complexité linéaire d'une suite produite par un registre linéaire de dimension n ne dépasse pas n, mais il est possible de combiner plusieurs de ces suites pour en obtenir une qui soit plus complexe. Supposons que nous disposions de deux séquences produites par des registres linéaires de tailles respectives r et s que l'on choisira sans diviseur commun. Nous supposerons aussi que les polynômes de ces registres sont primitifs et n'ont, eux non plus, pas de diviseur commun. Toute combinaison de ces deux suites peut être décrite à partir des opérations d'addition ou de multiplication modulo 2. On peut démontrer les résultats suivants :



$$u_t = x_t y_t \oplus y_t z_t \oplus z_t = \begin{cases} z_t & \text{si } y_t = 0 \\ x_t & \text{si } y_t = 1 \end{cases}$$

FIGURE 2.6. Un générateur typique de suite chiffrante : le générateur de Geffe (1973). Il est constitué de trois registres rebouclés linéairement de tailles respectives r, s et t, dont les polynômes de connexion sont primitifs (voir Fig. 2.5) et tels que les périodes des trois registres sont deux à deux premières entre elles. La période de la suite (u_t) est le produit des périodes des trois registres : $(2^r - 1)(2^s - 1)(2^t - 1)$. La fonction de filtrage $f(x,y,z) = xy \oplus zy \oplus y$ est équilibrée, ce qui assure de bonnes propriétés statistiques à la suite produite. La complexité linéaire de cette dernière vaut $rs + st + t$. Toutefois, ce générateur souffre encore de certaines faiblesses. Dans trois cas sur quatre, $u_t = z_t$ et dans trois cas sur quatre aussi, $u_t = x_t$. Un cryptanalyste habile ne manquera pas d'exploiter ces corrélations.

- la suite obtenue en additionnant modulo 2 les termes des deux suites est de période $(2^r - 1)(2^s - 1)$ et sa complexité linéaire est $r + s$;
- la suite obtenue en multipliant les termes des deux suites est toujours de période $(2^r - 1)(2^s - 1)$ mais cette fois, la complexité linéaire, égale à $r \times s$, a considérablement augmenté. Hélas, elle présente un biais rédhibitoire. En moyenne trois quarts des termes valent 0 pour seulement un quart de 1 !

L'emploi de la multiplication s'avère donc cryptologiquement très intéressant, mais il faut corriger le biais statistique qui en résulte. Cela peut se faire en combinant plus de deux registres, avec une fonction équilibrée, c'est-à-dire qui prend autant de fois la valeur 0 que la valeur 1, et qui s'exprime comme un polynôme de plusieurs variables de degré élevé. L'architecture typique d'un générateur de suites chiffrantes comprend plusieurs registres linéaires pour assurer une très longue période et de bonnes statistiques, associés à une ou plusieurs fonctions non linéaires destinées à augmenter la complexité de la suite produite.

4.6 Chiffrement autosynchronisant

Lors d'une communication protégée par un chiffrement à flot, les deux correspondants doivent être parfaitement synchronisés. Le symbole reçu par le destinataire doit être déchiffré avec le même symbole chiffrant que celui dont s'est servi l'expéditeur. Le moindre décalage à la réception est fatal au déchiffrement de tout le reste du message. Le destinataire ne peut alors plus rien déchiffrer correctement.

Pour pallier ce défaut, il existe des mécanismes dits *autosynchronisants*, où le générateur pseudo-aléatoire de l'émetteur et du destinataire sont couplés, ce qui permet de se re-synchroniser en cas de perte d'un symbole de cryptogramme. Dans ces systèmes, le symbole chiffrant k_t est le résultat d'un calcul qui fait intervenir la clé secrète et un certain nombre fixé de symboles du cryptogramme. Si le destinataire reçoit ces symboles correctement, il est alors assuré de disposer du bon symbole chiffrant et se re-synchronise automatiquement.

5 Le chiffrement par bloc

Le chiffrement par bloc est la seconde grande famille de procédés de chiffrement symétrique. Il est le prolongement des substitutions simples. Le message est partagé en blocs de même taille, par exemple 64 ou 128 symboles binaires, une transformation inversible dépendante de la clé opère sur chaque bloc pour fournir un bloc de cryptogramme. La transformation inverse est utilisée au déchiffrement. Il s'agit finalement d'une substitution sur un alphabet de taille considérable constitué de tous les mots binaires de taille 64 ou 128.

5.1 Architecture générale d'un chiffrement par bloc

Sur un chiffrement par bloc, la propriété de diffusion signifie qu'un changement, ne serait-ce que d'un seul symbole binaire de la clé ou du message en clair, doit provoquer le changement en moyenne d'un symbole sur deux du cryptogramme. La propriété de confusion signifie que la relation de dépendance du cryptogramme vis-à-vis du message en clair et de la clé doit être assez complexe pour interdire au cryptanalyste toute résolution des équations définissant la fonction de calcul. Aucune réalisation simple n'atteint ces propriétés. C'est pourquoi un calcul par bloc itère des calculs simples sur plusieurs tours. L'architecture typique d'un chiffrement par bloc comprend les éléments suivants (Fig. 2.7) :

1. une fonction de tour qui opère sur un bloc et qui est assez simple pour être matériellement réalisable ;
2. un dispositif qui, à partir de la clé, produit plusieurs *sous-clés* qui serviront de paramètres à chaque fonction de tour.

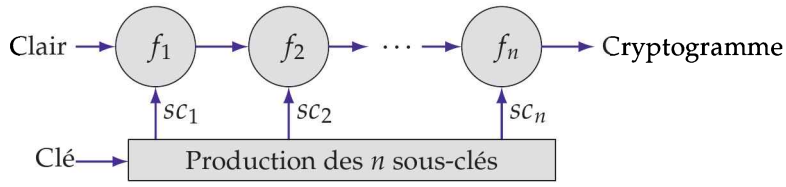


FIGURE 2.5. Architecture typique d'un chiffrement par bloc : la clé est déclinée en n sous-clés $sc_1, sc_2, \ldots, sc_n$, et le calcul est itéré sur n tours.

Le nombre n de tours de calcul doit être suffisant pour assurer les bonnes propriétés de confusion et de diffusion. Mais il ne doit pas être trop élevé pour ne pas pénaliser la rapidité du calcul. Tout l'art du concepteur est de trouver le juste équilibre entre le nombre de tours et la complexité de chacun d'entre eux. La fonction de déchiffrement, qui calcule le message en clair à partir du cryptogramme, itère les fonctions inverses de chaque tour dans l'ordre contraire. Les deux constructions classiques de fonctions inversibles sont les réseaux de substitutions et permutations, et le schéma de Feistel.

5.2 Les réseaux de substitutions et permutations

Une façon efficace de réaliser une fonction complexe, c'est-à-dire qui satisfait le principe de confusion, est d'utiliser une mémoire à lecture seule (ROM, *Read Only Memory*). Il s'agit d'un dispositif capable de mémoriser des données fixes inscrites en usine au moment de la fabrication. Il comprend une entrée de n symboles binaires et une sortie de m symboles binaires. L'entrée définit l'adresse de l'une des 2^n cellules, qui est décodée pour accéder aux données inscrites dans la cellule située à cette adresse. Les m symboles binaires qui y sont inscrits sont alors dirigés vers la sortie. Une fonction inversible incluse dans un tel dispositif est appelée une *boîte de substitution*.

Toute fonction arbitraire est réalisable ainsi. Malheureusement, le nombre de cellules croît exponentiellement avec la taille de l'adresse, ce qui limite le nombre d'entrées à une dizaine de symboles binaires. Lorsque, comme en cryptographie, on souhaite réaliser des fonctions inversibles de plusieurs dizaines d'entrées, il faut combiner plusieurs boîtes de substitution. Mais il faut aussi assurer la propriété de diffusion. Une façon de réaliser un assemblage qui assure cette propriété est de permuter les connexions entre les boîtes de substitution de sorte que les sorties de chaque boîte soient connectées vers les entrées de toutes les autres boîtes, comme cela est illustré sur la figure 2.8. Ainsi, un changement sur l'une quelconque des entrées aura un impact sur toutes les sorties.

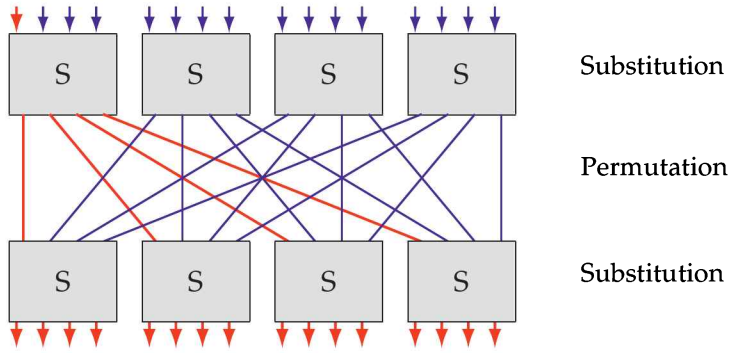
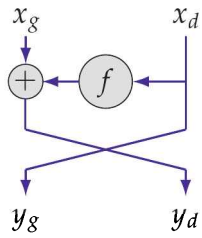


FIGURE 2.6. Réseau de substitutions et permutations assurant que chaque entrée influence toutes les sorties et que chaque sortie dépend de toutes les entrées, assurant de bonnes propriétés de diffusion. Un changement sur l'entrée marquée en rouge influence *toutes* les sorties. La transformation inverse est assurée en inversant chaque boîte de substitution.

5.3 Le schéma de Feistel

Le schéma de Feistel est une manière de réaliser une fonction inversible à partir d'une fonction qui ne l'est pas forcément. L'entrée est partagée en deux composantes de même taille : une entrée gauche x_g et une entrée droite x_d. La valeur est calculée selon le schéma de la figure 2.9.



$$\begin{cases} y_g &= x_d \\ y_d &= x_g \oplus f(x_d) \end{cases}$$

FIGURE 2.5. Schéma de Feistel permettant de réaliser une fonction inversible à partir d'une fonction f qui ne l'est pas forcément. L'opération $\oplus$ est *l'addition sans retenue* qui réalise une addition modulo 2 sur chaque symbole binaire. Cette addition a la particularité d'être identique à la soustraction sans retenue.

La fonction inverse, qui exprime x_g et x_d en fonction des sorties gauche y_g et droite y_d est donnée par :

$$\begin{cases} x_d &= y_g \\ x_g &= y_d \oplus f(y_g) \end{cases}$$

Dans un schéma de Feistel, la sortie gauche est égale à l'entrée droite. Les propriétés de diffusion et de confusion ne sont pas assurées. Pour qu'elles le soient, il est nécessaire d'itérer plusieurs tours de manière à composer plusieurs schémas en série.

5.4 Le *Data Ecryption Standard* (DES)

Historique

L'algorithme de chiffrement DES est le résultat d'un appel à contributions lancé en 1973 par le NBS (*National Bureau of Standards*) pour un algorithme de chiffrement utilisable par les entreprises qui ont besoin de protéger les fichiers et les communications sensibles. Cet appel d'offres a conduit à la standardisation en 1976 d'un algorithme de chiffrement à usage civil qui a été publié en janvier 1977. Il s'agit de la première publication à grande échelle d'un algorithme cryptographique et elle marque l'entrée de la cryptologie dans le domaine public.

L'entreprise IBM (*International Business Machines*) disposait alors d'une famille d'algorithmes, appelée *Lucifer* qu'elle a proposée pour cet usage. Ces algorithmes avaient été mis au point à partir du début des années 1970 par Horst Feistel. Une des versions de Lucifer est proposée pour concourir à l'appel d'offres et sera retenue pour devenir le *Data Encryption Standard* (DES). Il s'agit d'un schéma de Feistel à 16 tours, précédé et suivi d'une permutation des composantes du bloc de données. Il opère sur des blocs de 64 symboles binaires (Fig. 2.10).

La NSA (*National Security Agency*) a imposé des modifications sur l'algorithme Lucifer d'IBM, en particulier en réduisant la taille de clé à 56 symboles binaires au lieu des 112 que proposait IBM. Cette taille autorisait à l'époque la recherche exhaustive des 2^{56} clés par des institutions disposant de très puissants moyens de calcul, et permettait ainsi un contrôle des communications chiffrées. La NSA a également exigé une modification des boîtes de substitutions non linéaires. Cette exigence a fait peser un soupçon sur le DES. La NSA y a-t-elle introduit une trappe de manière à pouvoir être seule à résoudre les cryptogrammes produits ? La communauté cryptographique a recherché des attaques contre le DES et il s'est avéré que les boîtes de substitution proposées par la NSA rendait l'algorithme plus résistant, faisant du DES un algorithme très solide. Aujourd'hui encore, l'attaque la plus réaliste reste la recherche exhaustive de la clé par force brutale (voir le paragraphe 1 page 81).

Le triple DES

La taille de la clé, égale à 56 symboles binaires, convenable lors de la création du DES en 1977, est devenue insuffisante en raison des progrès des moyens de calculs. La recherche de la clé par force brutale devenait envisageable avec des moyens personnels. Pour pallier cette faiblesse, la NSA a proposé le *triple DES*. Il utilise deux clés k_1 et k_2 de 56 symboles binaires chacune, soit 112 symboles binaires en tout. La fonction de chiffrement compose la fonction DES en mode chiffrement utilisant k_1 avec la fonction DES en mode déchiffrement utilisant k_2

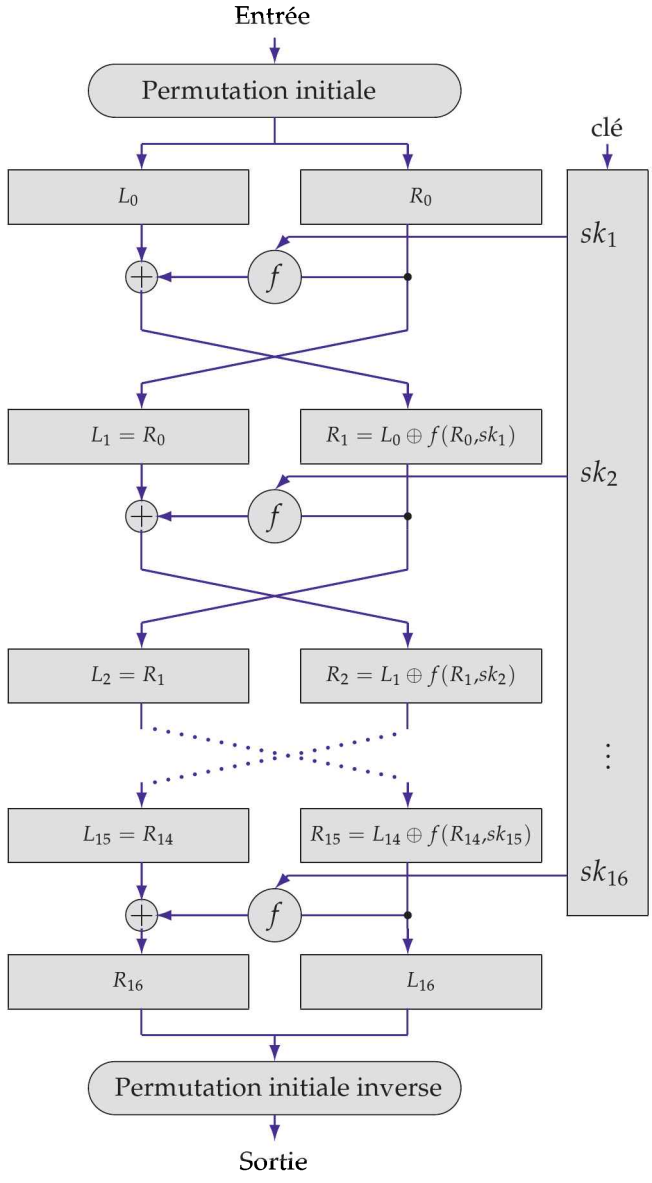


FIGURE 2.10. Architecture de l'algorithme DES. Il est composé d'un schéma de Feistel à 16 tours. Les parties gauche et droite du dernier tour ne sont pas permutées ; ainsi la fonction de déchiffrement a une structure identique à celle de la fonction de chiffrement, mais avec les sous-clés prises dans l'ordre inverse.

puis à nouveau la fonction DES en mode chiffrement utilisant k_1.

$$m \longmapsto DES_{k_1} \circ DES_{k_2}^{-1} \circ DES_{k_1}(m).$$

Lorsque les deux clés k_1 et k_2 sont égales, cela réalise un simple DES, ce qui permet de conserver une compatibilité avec les systèmes qui ne disposent encore que du simple DES.

5.5 L'*Advanced Encryption Standard* (AES)

L'algorithme AES a fait l'objet en 1997 d'un appel d'offre international du NIST (*National Institute of Standards and Technology*) pour remplacer le DES vieillissant. La réparation du DES en triple DES en a fait un procédé trop lent. L'objectif était de standardiser un nouvel algorithme de chiffrement pour le monde civil et commercial, à la fois plus sûr et plus rapide à exécuter sur les processeurs de calcul. Après l'évaluation publique de 15 candidats, le NIST a retenu en 2000 l'algorithme Rijndael, conçu par les cryptologues belges Vincent Rijmen et Joan Daemen. Il fait l'objet d'un standard FIPS 191. L'algorithme AES peut utiliser comme paramètre secret des clés de 128, 192 ou 256 symboles binaires pour chiffrer et déchiffrer des blocs de 128 symboles binaires. Il est construit autour d'un réseau de substitutions-permutations. Pour une clé de 128 symboles binaires, il itère un calcul élémentaire sur dix tours. L'itération élémentaire comprend quatre étapes.

1. La substitution des octets (*Byte Substitution*, BS) consiste à appliquer une fonction de substitution sur chacun des seize octets (Fig. 2.11). Cette fonction fortement non linéaire assure la propriété de confusion.

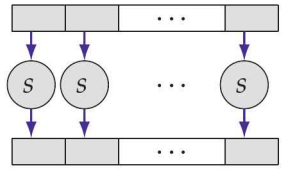


FIGURE 2.11. Substitution des octets. La transformation $x \mapsto s(x)$ est appliquée à chaque octet.

2. Le bloc de 128 symboles binaires est organisé en seize octets disposés en carré de quatre sur quatre. La rotation des lignes (*Shift Row*, SR) opère une rotation cyclique de chacune des lignes du carré, différente pour chaque ligne (Fig. 2.12).
3. Le mélange des colonnes (*Mix Column*, MC) transforme les colonnes du carré en leur appliquant une transformation linéaire. Cette transformation

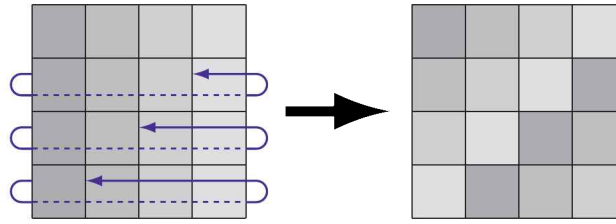


FIGURE 2.11. Rotation cyclique appliquée à chaque ligne.

est absente du dernier tour, afin que la fonction de déchiffrement soit semblable à la fonction de chiffrement (Fig. 2.13). La rotation des lignes et le mélange des colonnes sont des transformations linéaires qui assurent la propriété de diffusion.

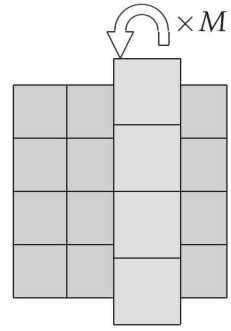


FIGURE 2.13. Mélange des colonnes : applique la transformation linéaire M sur chacune des colonnes.

4. L'addition des sous-clés (*Add Sybkey*) consiste à ajouter modulo 2, composante à composante, les sous-clés qui sont dérivées de la clé principale (Fig. 2.14).

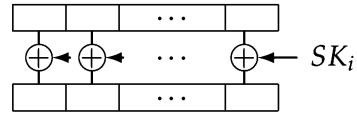


FIGURE 2.13. Addition des sous-clés à chaque octet du bloc.

5.6 Les modes opératoires

Une fonction de calcul par bloc est adaptée au chiffrement de messages dont le nombre de symboles binaires est exactement la taille du bloc. Pour chiffrer

des messages plus longs qu'un bloc, il convient d'appliquer plusieurs fois le calcul. Les modes opératoires des chiffrements par bloc ont pour objet de décrire comment procéder. Ils sont normalisés par l'institut américain NIST (*National Institute of Standards and Technology*) dans le document FIPS 81 (*Federal Information Processing Standards*) en date du 2 décembre 1980. On supposera pour simplifier que le message a une taille multiple de la taille du bloc traité par la fonction de calcul, quitte à le compléter par des symboles binaires égaux à zéro. La suite de ce paragraphe décrit les modes opératoires classiques.

Le mode dictionnaire

Le mode *dictionnaire* ou mode ECB (*Electronic CodeBook*) réalise une substitution simple. Le message est découpé en blocs de même taille et tous les blocs sont chiffrés de manière identique, indépendamment les uns des autres en appliquant sur chacun d'entre eux la fonction de calcul sur un bloc. Ce mode présente l'inconvénient de chiffrer de la même façon les blocs identiques du message en clair.

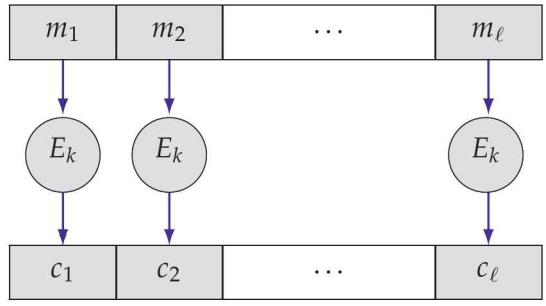


FIGURE 2.13. Le mode dictionnaire ou mode ECB (*Electronic CodeBook*). Le cryptogramme est partagé en blocs et chaque bloc est chiffré de manière indépendante. Ce mode, bien que normalisé par le NIST, n'est pas sûr !

Le mode chaîné

Le mode *chaîné*, ou mode CBC (*Cipher Block Chaining*) nécessite un bloc initial appelé IV (*Initial Vector*), et transmis avec le cryptogramme. Ce bloc initial doit être propre à chaque message. Le plus sûr est qu'il soit aléatoire et utilisé une seule fois. Ensuite, chaque bloc de clair est additionné modulo 2 au bloc de cryptogramme précédent avant d'être traité par la fonction de chiffrement par blocs, comme illustré sur la figure 2.16. Avec cette façon de procéder, le chiffrement d'un bloc dépend de tous les blocs précédents, ce qui a pour avantage de meilleures propriétés de diffusion.

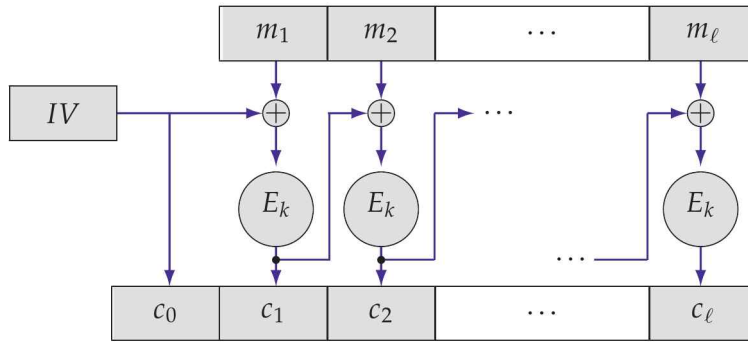


FIGURE 2.16. Mode chaîné ou mode CBC (*Cipher Block Chaining*). Le cryptogramme d'un bloc est additionné modulo 2 au bloc clair suivant de manière à faire dépendre chaque bloc du cryptogramme de tout ce qui précède. Cela nécessite un bloc initial (IV – *Initial Vector*) pour démarrer le chaînage. Contrairement au mode ECB, le mode CBC est sûr.

Sécurité comparée des modes ECB et CBC

Le cryptogramme ne doit apporter aucune information sur le message en clair. Or ce n'est pas le cas lorsqu'on utilise le mode ECB. Si deux blocs de cryptogramme sont identiques, c'est qu'ils proviennent de blocs identiques du message en clair, et cela constitue une information à la disposition du décrypteur, comme cela est illustré sur la figure 2.17. Le mode CBC, lui, n'a pas cet inconvénient.

Autres modes opératoires

Il existe d'autres modes opératoires des calculs par bloc qui réalisent en fait un chiffrement à flot. Ces modes sont présentés sur les figures 2.18 et 2.19. Ils utilisent la fonction de calcul pour produire une séquence pseudo-aléatoire qui servira à masquer le message. Ils diffèrent dans la façon dont est produite cette séquence. Le mode OFB (Fig. 2.18) utilise la fonction de calcul par bloc comme fonction de transition d'un automate qui produit un masque pseudo-aléatoire. Le mode CTR (Fig. 2.19) produit la séquence pseudo-aléatoire en appliquant la fonction de calcul à des entiers consécutifs.

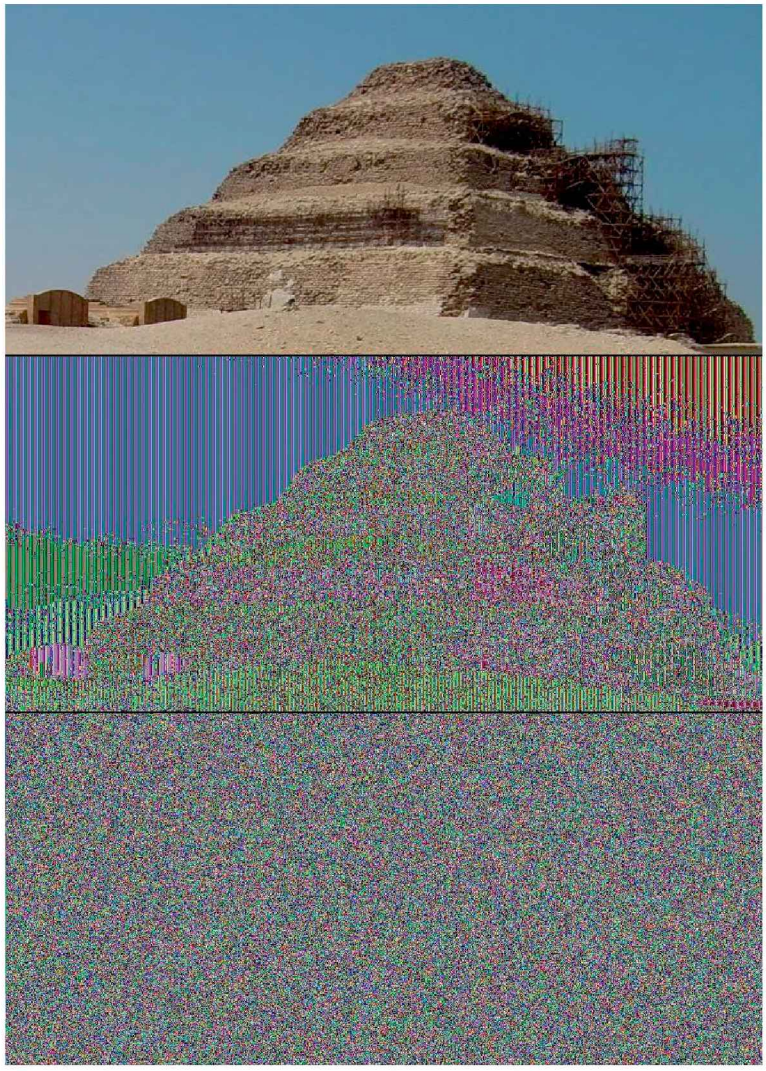


FIGURE 2.13. Cette figure montre trois images. L'image originale est en haut. Au centre elle a été chiffrée en mode ECB, et en bas en mode CBC. Le chiffrement ECB ne dissimule pas entièrement l'image originale.

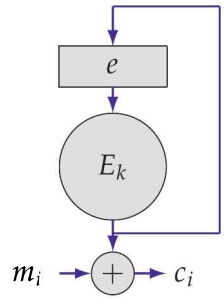


FIGURE 2.18. Le mode OFB (*Output FeedBack*) : une suite chiffrante est produite par un automate dont la fonction de mise à jour de l'état est la fonction de calcul par bloc. Cette suite masque chaque bloc du message en clair m_i pour produire un bloc du cryptogramme c_i.

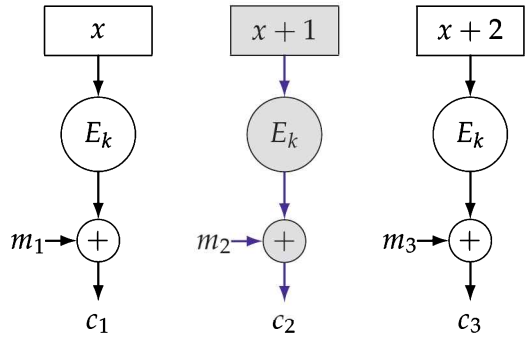


FIGURE 2.19. Le mode CTR, ou mode compteur : le masque est le résultat du chiffrement de la valeur d'un compteur initialisé à une valeur convenue x. Pour chaque bloc, la valeur du compteur est incrémentée, conduisant à un nouveau masque qui est ajouté au bloc du message en clair pour produire le bloc du cryptogramme.

3

La cryptographie à clé publique

En 1976, Whitfield Diffie et Martin Hellman, du Department of Electrical Engineering à l'université de Stanford en Californie ont publié un article intitulé *Les nouvelles orientations de la cryptographie*, qui introduit la notion de clé publique. L'article commence ainsi :

Nous sommes aujourd'hui à l'aube d'une révolution en cryptographie.

Effectivement, la révolution a bien eu lieu. Le développement des communications par ordinateurs mis en réseau a ouvert de nouveaux besoins qui ne pouvaient pas être mis en œuvre avec la cryptographie conventionnelle :

- Comment deux protagonistes qui ont besoin d'échanger de manière discrète pour la première fois, et qui ne se sont jamais rencontrés auparavant, peuvent-t-il échanger une clé sur un réseau ouvert que tout le monde peut écouter ?
- Comment trouver un équivalent numérique à tout ce qui établit la confiance dans les relations d'affaires, en particulier comment signer numériquement un document ?

Elle a été aussi une révolution des mentalités. On parle maintenant de clé publique dans un milieu habitué au secret ! Comme l'ont noté Diffie et Hellman dans leur article, la cryptologie à clé publique constitue l'aboutissement d'un mouvement qui a vu continuellement décroître la part secrète. La sécurité du chiffre de César suppose que le procédé lui-même soit tenu secret. Kerckhoffs a énoncé que le procédé doit pouvoir être connu de tous, de telle sorte que le vol ou la perte d'un appareil ne compromette pas les futurs messages chiffrés avec de nouvelles clés. Les procédés modernes de cryptographie symétrique sont sûrs y compris lorsque le texte clair de certains cryptogrammes interceptés est connu, éliminant l'inconvénient de garder secret les anciens messages. Chacun de ces développements a diminué ce qui doit rester confidentiel.

1 Les fonctions à sens unique

Une fonction est dite « *à sens unique* » lorsqu'elle est facilement calculable, mais difficile à inverser. À partir de tout élément x, il existe un algorithme efficace pour calculer la valeur de $f(x)$. Mais étant donné une valeur y dont on sait qu'elle provient d'un calcul $y = f(x)$, il est pratiquement impossible de trouver un antécédent x convenable.

Dès le XVII^{e} siècle, on a remarqué qu'autant il est facile de multiplier deux nombres, même très grands, autant il est difficile de retrouver les facteurs une fois le produit calculé (voir paragraphe 6 page 99). La factorisation d'un entier de plus de 250 chiffres reste encore hors de portée de nos capacités actuelles. Le record, en date du 7 janvier 2010, est de 232 chiffres décimaux (768 chiffres binaires). Ainsi, je peux choisir deux nombres premiers p et q très grands, par exemple d'une centaine de chiffres chacun. Je les multiplie et je publie leur produit $n = p \times q$. Si je garde secret mes facteurs p et q, je serai seul à connaître la factorisation de n. Et si je les perds, je serai incapable de les retrouver.

La notion de fonction à sens unique est pour l'instant une notion empirique. De gros progrès ont été accomplis pour factoriser les entiers, mais ce problème reste encore aujourd'hui un problème difficile pour lequel on ne connaît pas de solution efficace. Cette difficulté est-elle dans la nature même du problème ? ou bien est-elle la conséquence de notre ignorance ? On ne sait toujours pas répondre à cette question. L'existence même de fonctions à sens unique reste une hypothèse. Si ce type de fonctions existe, la multiplication des entiers en est un candidat.

Un autre exemple est la fonction exponentielle en base a. Il s'agit de la fonction qui à un entier n associe a^n, égal au produit de n facteurs égaux à a. Sur l'ensemble des entiers, cette fonction est rapidement incalculable, car le résultat croît très rapidement. Les opérations deviennent de plus en plus longues jusqu'à devenir irréalisables. En revanche, dans l'anneau des entiers modulo m, cela devient plus aisé. À chaque étape du calcul, la réduction modulo m ramène le résultat à une taille raisonnable et un calcul est envisageable, y compris pour de très grandes valeurs de l'exposant. Un exposant de plusieurs centaines de chiffres n'est vraiment pas un obstacle.

La fonction réciproque de l'exponentielle en base a s'appelle le *logarithme* en base a. Le logarithme d'un nombre y est l'exposant ℓ qui convient pour avoir $y = a^{\ell}$. Pour cette fonction, la situation s'inverse. Autant dans l'ensemble des entiers il existe des algorithmes efficaces pour calculer des valeurs approchées du logarithme de n'importe quel entier en base a, autant dans l'anneau des entiers modulo m, pour un élément y compris entre 1 et $m-1$ donné, trouver l'exposant ℓ tel que $a^{\ell} = y$, reste un problème difficile et pratiquement impossible à résoudre dès que m dépasse quelques centaines de chiffres. La fonction exponentielle sur

l'anneau des entiers modulo m est un second exemple de candidat à être une fonction à sens unique.

S'il existe des fonctions mathématiques à sens unique, alors pourquoi ne pas imaginer un mécanisme cryptographique dans lequel une clé de déchiffrement serait un élément x et la clé de chiffrement associée serait l'image $f(x)$ par une fonction à sens unique ? L'image $f(x)$ pourrait alors être publiée et l'élément x soigneusement gardé secret sans qu'il ne soit compromis par la connaissance de $f(x)$. Il n'y aurait plus de transport de secret préalable à une communication confidentielle. C'est le pas qui a été franchi avec l'invention du chiffrement à clé publique. Cette invention a aussi ouvert la voie à d'autres protocoles qui numérisent d'autres éléments de l'activité humaine, comme par exemple la signature.

Les premiers procédés à clé publique à avoir été présentés exploitent des résultats de l'arithmétique et de la théorie des nombres, présentés jusqu'alors comme des mathématiques dites « *pures* », contredisant la déclaration, en 1940, du mathématicien G. H. Hardy dans *The Mathematician's Apology* :

> *Les vraies mathématiques n'ont aucun effet sur la guerre. Personne n'a encore trouvé un objectif militaire qui serait dépendant de la théorie des nombres.*

2 Le chiffrement

2.1 L'échange de clé Diffie-Hellman

Diffie et Hellman ont publié dans leur article de 1976 un procédé qui permet à deux correspondants de partager une donnée secrète en échangeant publiquement des informations. Ceux qui observent l'échange ne peuvent pas reconstituer leur secret. Cela résout le problème du partage d'une clé secrète à distance entre deux correspondants qui ne se sont jamais rencontrés (Fig. 3.1).

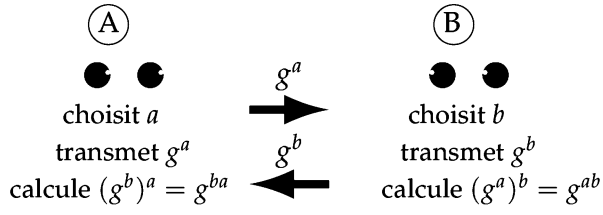


FIGURE 3.1. Échange Diffie-Hellman : deux correspondants s'accordent sur un secret g^{ab}. L'observateur de l'échange n'accède qu'à g^a et à g^b. Il n'existe aujourd'hui aucun moyen efficace pour en déduire g^{ab}. Cette donnée peut servir de clé secrète pour protéger les communications entre les deux correspondants par une cryptographie symétrique.

Pour cela, ils conviennent d'un nombre premier p, assez grand pour empêcher tout calcul de logarithme modulo p. Toutes les opérations s'effectueront modulo ce grand nombre premier. Ils conviennent également d'un nombre g qui permet d'engendrer tous les entiers modulo p. Les éléments g et p peuvent sans inconvénient être dévoilés. L'un des protagonistes choisit un nombre aléatoire a, puis calcule g^a modulo p qu'il transmet à son correspondant. Celui-ci fait de même, il choisit un nombre aléatoire b, puis calcule g^b qu'il transmet en retour à son correspondant. Maintenant, les deux protagonistes peuvent calculer g^{ab}. L'un le calcule en élevant la donnée g^b qu'il a reçue à la puissance a, et l'autre le calcule en élevant la donnée g^a qu'il a reçue à la puissance b. Ceux qui ont observé cet échange ont vu transiter g^a et g^b.

Bien sûr, si les logarithmes modulo p étaient facilement calculables, le secret g^{ab} serait accessible à tous, puisque l'observation de g^a permettrait de retrouver a, et on se retrouverait exactement dans la même situation que l'un des correspondants. Non seulement le calcul du logarithme discret reste aujourd'hui un problème difficile, mais on ne connaît toujours pas d'autre méthode pour calculer g^{ab} à partir de g^a et de g^b. Ce dernier problème s'appelle le *problème Diffie-Hellman*.

2.2 Le problème de l'intrus

L'échange de clé Diffie-Hellman résout le problème du transport des clés entre deux personnes n'ayant pas eu d'accointance préalable. Mais il n'assure pas encore que ces deux correspondants puissent communiquer en toute confiance. Il manque encore un ingrédient. Imaginons qu'un intrus se glisse entre ces deux correspondants de telle sorte que, bien qu'ils croient communiquer entre eux, ils envoient en fait leurs messages à cet intrus qui ne fait que recevoir, observer et retransmettre. Cet intrus, cet homme au milieu (*man in the middle*), cet intermédiaire indésirable, peut sans problème échanger un secret avec l'un et un autre secret avec l'autre. Les deux correspondants se sentant protégés dans leurs communications verront tous leurs échanges interceptés, déchiffrés puis réchiffrés, et finalement entièrement dévoilés à cet intrus (Fig. 3.2). Il est nécessaire de s'assurer de la provenance des messages en les authentifiant, et cela sera assuré par la signature numérique, présentée au paragraphe 3, page 57.

2.3 Le chiffrement ElGamal

Le procédé Diffie-Hellman pour l'échange de clé peut se décliner en une opération de chiffrement à clé publique, appelé *chiffrement ElGamal* du nom du cryptologue américain d'origine égyptienne, Taher ElGamal, qui l'a inventé en 1985.

On utilise un algorithme de chiffrement symétrique E, qui permet le chiffrement de tout message m avec pour clé secrète un entier compris entre 1 et $p-1$.

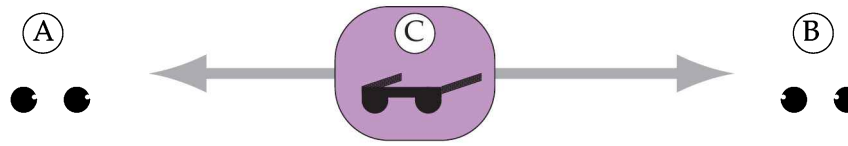


FIGURE 3.2. Le problème de l'intrus : deux correspondants croient communiquer entre eux, mais tous les deux communiquent en fait avec un intrus qui intercepte tous leurs échanges. Pour éviter cela, les correspondants doivent s'authentifier par une signature numérique.

Le destinataire dispose d'une clé privée s connue de lui seul qui consiste en un entier compris entre 1 et $p-1$. La clé publique qui lui correspond est $y = g^s$ modulo p. Rappelons qu'il est très difficile de retrouver s à partir de y. La valeur de y peut sans inconvénient être publiée.

Chiffrement d'un message

Le chiffrement du message m pour un destinataire ne fait appel qu'à sa clé publique y. Le message sera dissimulé avec l'algorithme E et une clé secrète établie uniquement à cette occasion, appelée la *clé de session*. Cette clé est calculée comme dans le protocole Diffie-Hellman.

Pour illustrer comment fonctionne ce chiffrement, considérons le nombre premier $p = 23$. L'entier $g = 5$ engendre tous les entiers modulo 23, ce qui signifie que les puissances successives de 5 modulo 23, qui sont 1, 5, 2, 10, 4, 20, 8, etc. parcourent tous les entiers de 1 à 22. Pour notre exemple, le procédé de chiffrement E sera un simple décalage de César. Chiffrons le message « *ElGamal* » pour un destinataire dont la clé privée est $s = 13$. La clé publique correspondante est $y = 5^{13} = 21$ modulo 23. Les étapes du chiffrement sont les suivantes :

- choisir un exposant a au hasard entre 1 et $p-1$, par exemple $a = 10$;
- calculer la clé de session avec la clé publique du destinataire avec la formule $k = y^a$ modulo p, soit $k = 21^{10} = 12$ modulo 23 ;
- se servir de cette clé de session pour chiffrer le message avec la formule $c = E_k(m)$. Dans notre exemple, ce sera le résultat d'un décalage de douze positions dans l'alphabet, soit « *QxSmymx* » ;
- calculer la donnée $z = g^a$ modulo p qui permettra au destinataire de calculer la clé de session. Dans notre exemple, z vaut $5^{10} = 9$ modulo 23. Le cryptogramme transmis au destinataire est le couple (z,c), qui ici est (9,« *QxSmymx* »).

Dans le cryptogramme (z,c), la composante c contient l'information du message, dissimulé par chiffrement avec la clé de session, et la donnée z permet au seul détenteur de la clé privée de retrouver la clé de session.

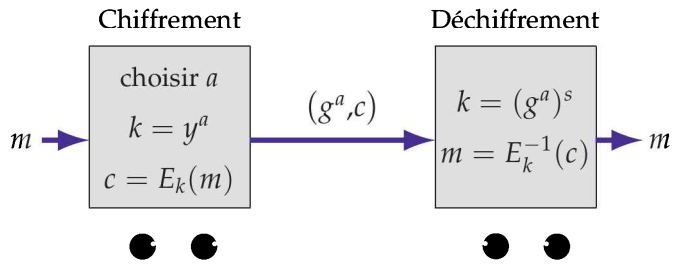


FIGURE 3.2. Le système ElGamal. La clé publique est constituée d'un nombre premier p, d'un générateur g des entiers modulo p et de la valeur $y = g^s$ modulo p, où s est la clé privée du destinataire. Pour chiffrer un message m, l'émetteur choisit un entier aléatoire a entre 1 et $p - 1$, calcule la clé de session $k = y^a$ modulo p et chiffre son message avec pour obtenir un chiffré c. Le cryptogramme est constitué du couple (g^a,c). Le destinataire calcule la clé de session par $k = (g^a)^s$ modulo p, ce qui lui permet de déchiffrer c et de retrouver m.

Déchiffrement du cryptogramme

À la réception des éléments (z,c), le destinataire peut calculer la clé de session grâce à sa clé privée s par la formule $k = z^s$ qui vaut bien $(g^a)^s = g^{as} = y^a = k$. Une fois la clé de session connue, il lui suffit d'appliquer l'algorithme de déchiffrement au message chiffré c avec cette clé pour retrouver le message en clair m.

Dans notre exemple, le destinataire a reçu le cryptogramme $(9,$« *QxSmymx* »$)$. Sa clé privée lui permet de retrouver la clé de session en calculant $k = 9^{13} = 12$ modulo 23. Cette clé indique le décalage inverse à faire opérer sur le chiffré « *QxSmymx* » pour retrouver le message en clair « *ElGamal* ».

Une trappe secrète pour déchiffrer

Un mécanisme de chiffrement à clé publique réalise ce qu'on appelle une *fonction à sens unique avec trappe secrète*. Tout émetteur peut calculer le cryptogramme d'un message donné mais, pour inverser ce calcul, il faut disposer d'une information supplémentaire qu'on ne doit pas révéler : la clé privée. Symboliquement, cette clé privée est une trappe secrète par laquelle s'opère l'inversion du chiffrement. L'image du cadenas est souvent utilisée pour illustrer ce principe. La clé publique est un cadenas ouvert, mis à la disposition de tous, que tout le monde peut fermer pour protéger la confidentialité d'un message. La clé privée est celle qui ouvre le cadenas, que seul détient le destinataire, avec laquelle il peut avoir accès au message. Le chiffrement à clé publique s'apparente à la dissimulation du message dans une boîte fermée avec un cadenas (Fig. 3.4).

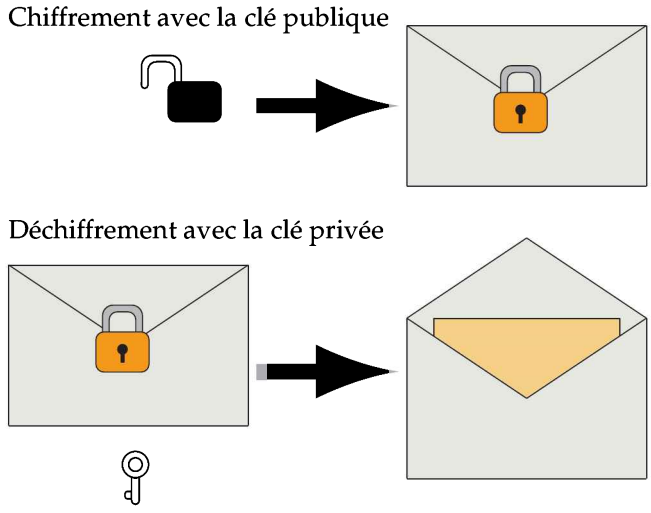


FIGURE 3.4. Illustration du chiffrement à clé publique : la clé publique est un cadenas ouvert que tout expéditeur peut fermer pour protéger un message à l'attention du destinataire. La clé privée, que seul le destinataire détient, est celle qui peut ouvrir le cadenas.

2.4 Le chiffrement de Rabin

Le chiffrement de Rabin est un procédé de chiffrement à clé publique dont la trappe est la connaissance de la factorisation d'un grand entier. On peut même montrer qu'il est aussi difficile à résoudre que les entiers sont difficiles à factoriser.

- La clé privée est un couple d'entiers premiers p et q.
- La clé publique est le produit $n = p \times q$, appelé le *module* public. Les facteurs p et q ne doivent bien sûr pas être révélés.

Les entiers p et q doivent être choisis assez grands pour que la factorisation de l'entier n soit un problème insurmontable. Le chiffrement de Rabin opère sur un message qui est codé par un entier compris entre 0 et $n - 1$. Il consiste à l'élever au carré modulo n. Cette opération est simple à effectuer, rendant ce mécanisme particulièrement adapté lorsqu'il doit être réalisé sur un dispositif ne disposant que d'une très faible puissance de calcul. Le déchiffrement est l'extraction d'une racine carrée du cryptogramme. Si la factorisation de n est connue, il suffit d'extraire les racines carrées modulo les deux facteurs p et q, ce qui est un problème facile, puis de conclure avec le théorème chinois des restes.

Encadré 3.1. Extraire une racine carrée modulo un nombre premier p.

Si on sait que l'entier a est le carré d'un autre entier b modulo p et, si par ailleurs $p+1$ est multiple de 4, le calcul de $y = a^{\frac{p+1}{4}}$ modulo p donne directement une racine carrée de a modulo p. Pour le vérifier, élevons y au carré modulo p : $y^2 = a^{\frac{p+1}{2}} = b^{p+1} = b^{p-1} \times b^2$. En se souvenant du petit théorème de Fermat qui énonce que $b^{p-1} = 1$ modulo p dès que p est un nombre premier et que b n'est pas multiple de p, on trouve bien $y^2 = 1 \times b^2 = a$ modulo p.

Si $p+1$ n'est pas multiple de 4, il existe aussi une méthode efficace. Il faut choisir un entier x tel que $c = x^2 - a$ ne soit pas un carré modulo p, puis considérer un nombre i qui est une racine carrée imaginaire de c, tout comme une racine carrée imaginaire de (-1) définit les nombres complexes. Dans cette algèbre, le calcul $y = (x+i)^{\frac{p+1}{2}}$ modulo p fournit la racine carrée cherchée.

Lorsque n est le produit de deux nombres premiers impairs, un carré modulo n a quatre racines carrées. Par exemple, l'entier 26 a quatre racines carrées modulo 55 qui sont 9, 24, 31 et 46 (voir encadré 3.3 page 53). Pour un cryptogramme donné, il y a donc quatre messages en clair possibles. Pour la mise en œuvre pratique de ce procédé, on doit adjoindre une information permettant d'effectuer le bon choix parmi ces quatre possibilités.

Encadré 3.2. Le théorème chinois des restes.

Le traité Juzhang Suhashu, dont la date est estimée entre 280 et 473, pose le problème suivant :

> *« Combien l'armée de Han Xing comporte-t-elle de soldats si, rangés par 3 colonnes, il reste deux soldats, rangés par 5 colonnes, il reste trois soldats et, rangés par 7 colonnes, il reste deux soldats ? »*

Exprimés en termes contemporains, le problème de notre général chinois est un système d'équations congruentielles. Si un entier est x connu modulo a et modulo b et si, au surplus, les entiers a et b n'ont pas de diviseur commun, alors il est connu modulo le produit ab.

Le théorème de Bézout énonce l'existence de deux entiers u et v vérifiant $au + bv = 1$, qu'on peut trouver avec l'algorithme d'Euclide étendu. Si notre entier inconnu x est congru à x_a modulo a et à x_b modulo b, alors la solution à notre problème est donnée par la formule $x = x_a bv + x_b au$. En réduisant cette égalité modulo a puis modulo b, on observe qu'elle convient.

Appliquons cela à l'armée de Han Xing. L'énoncé nous apprend que $x_3 = 2$, $x_5 = 3$ et $x_7 = 2$. Une relation de Bézout entre 3 et 5 est $3 \times 2 - 5 \times 1 = 1$. Elle permet de trouver $x_{15} = 3 \times 6 - 2 \times 5 = 8$. Une relation de Bézout entre 15 et 7 est $15 \times 1 - 2 \times 7 = 1$. Elle permet de trouver $x_{105} = 2 \times 15 - 8 \times 14 = -82$, qui est congru à 23 modulo 105. Si l'on suppose qu'elle en comprend moins de 105, l'armée de Han Xing comprend 23 soldats. Heureuse époque !

Un argument de sécurité du système de Rabin

Extraire les racines carrées modulo n se réalise efficacement dès lors que la factorisation de l'entier n est connue. Mais n'existe-t-il pas un procédé qui se passerait de la connaissance de cette factorisation ? On montre en fait que si l'on sait extraire les racines carrées modulo n, alors on peut utiliser ce savoir pour factoriser l'entier n. Supposons par exemple qu'un calculateur prodige soit doué du pouvoir d'extraire les racines carrées modulo $n = 119$. Il choisit un entier x au hasard, par exemple $x = 15$, puis il l'élève au carré modulo 119. Il obtient $x^2 = 15^2 = 106$ modulo 119. Son don de calculateur lui apprend qu'une racine carrée de 106 modulo 119 est $y = 36$, ce qu'on peut vérifier immédiatement en élevant 36 au carré. Il se trouve que x et y ont le même carré modulo 119, mais x et y ne sont ni égaux ni opposés. Cela signifie que la différence $x^2 - y^2$ est un multiple de n sans que ni $x - y$, ni $x + y$ ne le soient. Les deux facteurs de n ne peuvent donc pas tous deux figurer dans $x - y$, ni dans $x + y$. Ils sont répartis entre $x - y$ et $x + y$. Le calcul du pgcd de n et de l'un d'entre eux fournit le facteur de n tant cherché. En effet, $x - y = 70$ et le calcul de $\text{pgcd}(119,70) = 7$ donne un facteur de 119. Cette méthode n'a pu aboutir que parce que le don de calculateur a révélé une autre racine carrée que 36 ou son opposé modulo 119, qui est 83. Dans ce cas, il aurait simplement recommencé avec une autre valeur de x prise au hasard.

En conclusion, il est tout aussi difficile de décrypter le système de Rabin que de factoriser son module public :

Tant que la factorisation sera difficile, le système de Rabin sera difficile à résoudre.

Encadré 3.3. Attention aux messages courts !.

Extraire la racine carrée d'un carré parfait est un problème facile qui a une solution efficace. C'est l'extraction *modulo n* qui est difficile. On devra éviter les messages clairs codés par des entiers si petits que leur élévation au carré ne requiert pas de réduction modulo n. Par exemple, si l'on doit chiffrer les messages $m = 7$ ou $m = -7$ avec un module public égal à $5 \times 11 = 55$, le cryptogramme sera $c = 49$. Extraire sa racine carrée entière pour retrouver $m = 7$ est à la portée de tous.

Le cryptogramme du message $m = 9$ est $c = 9^2$ qui est congru à 26 modulo 55. Cette fois, pour retrouver le message m à partir du cryptogramme, il faut la factorisation du module. Le cryptogramme 26 est congru à 1 modulo 5 et à 4 modulo 11 qui sont les carrés respectifs de 1 et de 2. En utilisant la relation de Bézout $11 \times 1 - 5 \times 2 = 1$, on trouve les quatre racines carrées de 26 modulo 55 qui sont $\pm 11 \times 1 \pm 10 \times 2$, soit 9, 24, 31 et 46.

L'élévation au carré modulo le produit n de deux grands nombres premiers p et q illustre à nouveau la notion de fonction à sens unique avec trappe secrète.

Nous pouvons tous calculer n'importe quel carré avec la seule connaissance du module n. Mais sans autre information, l'opération inverse, à savoir l'extraction des racines carrées modulo n reste inaccessible. Il faudrait pour cela connaître les facteurs premiers de n. Cette information supplémentaire, détenue uniquement par celui qui aura multiplié p et q pour diffuser n, est la trappe, le passage secret, la porte dérobée qui permet de retrouver la valeur de x à partir de son carré y.

2.5 Le RSA

Le RSA, du nom de ses trois auteurs Ronald Rivest, Adi Shamir et Leonard Adleman, est sans doute le plus utilisé des procédés à clé publique (Fig. 3.5). Comme le chiffrement de Rabin, sa sécurité repose sur la difficulté de factoriser les grands entiers.

- La clé privée est constituée de deux grands nombres premiers p et q.
- La clé publique est leur produit $n = p \times q$, qui est le module public, associé à un exposant public e, choisi premier avec $p - 1$ et $q - 1$. Bien sûr, les facteurs p et q sont maintenus secrets.

Un message est codé par un entier compris entre 0 et $n - 1$. Le chiffrement consiste à élever cet entier à la puissance e modulo n. Le cryptogramme est calculé avec la formule $c = m^e$ modulo n. Seules les données publiques sont requises pour le calculer. Grâce à la connaissance des facteurs du module public n, le destinataire est capable de déterminer un exposant d qui lui permettra d'extraire la racine e-ième du cryptogramme et de retrouver ainsi le message clair. L'extraction de la racine e-ième est réalisée en élevant le cryptogramme à la puissance d, soit $c^d = m$ modulo n. L'exposant d qui permet le déchiffrement est appelé l'*exposant privé*.

Contrairement au système de Rabin, on ne sait pas s'il existe un moyen de déchiffrer qui se passerait de la connaissance de la factorisation du module n. Le problème RSA, qui est le problème de l'extraction des racines e–ièmes modulo n, est un problème *a priori* plus facile que celui de la factorisation des entiers.

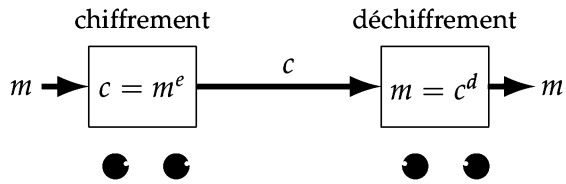


FIGURE 3.2. Le RSA. La clé consiste en un entier public n, un exposant public e et un exposant privé d. Le chiffrement d'un message m, qui est un entier compris entre 1 et $n - 1$, est élevé à la puissance exposant public modulo n. Le déchiffrement du cryptogramme ainsi obtenu consiste en son élévation à la puissance exposant privé modulo n.

Encadré 3.4. Calcul de l'exposant privé RSA.

Avec les facteurs p et q du module public n, on peut calculer $\lambda(n)$ qui est le plus petit multiple commun à $p-1$ et $q-1$. Rappelons que le petit théorème de Fermat énonce que pour tout nombre premier p, tout entier x non multiple de p élevé à la puissance $p-1$ modulo p vaut 1. On trouve également 1 si l'on élève x à la puissance $q-1$ modulo q. Comme $x^{\lambda(n)}$ est congru à 1, à la fois modulo p et modulo q, le théorème chinois des restes conduit à $x^{\lambda(n)} = 1$ modulo n.

L'exposant public e étant choisi premier avec $p-1$ et $q-1$, l'algorithme d'Euclide étendu permet de trouver deux entiers d et f qui satisfont la relation de Bézout $ed + f\lambda(n) = 1$. L'entier d fait figure d'inverse de e modulo $\lambda(n)$. Ainsi, pour tout entier x, on a $x^{ed} = x^{1-f\lambda(n)} = x$ modulo n.

En élevant un entier à la puissance e (chiffrement), puis à la puissance d (déchiffrement), on retrouve bien l'entier de départ. L'entier d convient comme exposant privé.

Prenons par exemple $n = 5 \times 11$. La valeur de $\lambda(n)$ est le plus petit multiple commun à $5-1=4$ et à $11-1=10$. C'est 20. Prenons $e=3$ comme exposant public. Il convient puisque 3 et 20 n'ont pas de diviseur commun autre que 1. Comme $3 \times 7 = 21$ qui est congru à 1 modulo 20, l'exposant privé $d=7$ convient.

Le cryptogramme du message $m=47$ est $c = 47^3$ modulo 55, qui vaut 38.

Pour déchiffrer le cryptogramme $c=38$, on l'élève à la puissance 7 modulo 55. On retrouve bien $38^7 = 47$ modulo 55.

Publié pour la première fois en août 1977 dans la rubrique des jeux mathématiques de Martin Gardner de la revue *Scientific American*, le RSA a essuyé de nombreux assauts, mais résiste toujours, à condition toutefois de suivre certaines précautions d'emploi.

- La connaissance de l'exposant privé permet de factoriser le module. On ne peut donc pas utiliser le même module pour des destinataires différents, chacun pourrait déchiffrer les messages destinés aux autres qui partagent le même module.
- Un exposant public égal à 3 est très populaire, car élever un entier au cube modulo n est une opération beaucoup plus rapide que si l'exposant est un nombre plus grand. Cependant, dans ce cas, il faut s'interdire de chiffrer des messages représentés par un entier petit dont l'élévation au cube ne demanderait pas de réduction modulo n, car extraire une racine cubique d'un entier est facile. La difficulté, rappelons le, c'est l'extraction des racines cubiques *modulo* n.
- Avec un exposant public égal à 3, il ne faut pas non plus chiffrer le même message à trois destinataires différents. Le théorème chinois des restes – encore lui ! – permet de connaître le cube du message, et l'extraction de sa racine cubique conduit au message clair.
- De manière symétrique, si l'exposant privé est trop petit, alors il existe une méthode pour le retrouver par un calcul simple sur les données publiques n

et e. Il ne faut pas que l'exposant privé soit plus petit que la racine quatrième du module.
- Enfin, la taille du module n doit être suffisante pour que cet entier reste hors de portée des meilleurs algorithmes de factorisation.

Si toutes ces mises en garde sont prises en compte, le RSA reste encore aujourd'hui un procédé sûr.

2.6 Le système de McEliece

Les systèmes à clé publique présentés jusqu'à présent sont construits à partir de la théorie des nombres. Celui présenté dans ce paragraphe est dû à Robert McEliece, mathématicien et professeur à l'Electrical Engineering au California Institute of Technology. Il ne repose pas sur l'arithmétique comme les précédents, mais sur la théorie des codes correcteurs d'erreurs. Ces codes ont été introduits pour lutter contre les parasites et le bruit de fond qui perturbent les communications radio. L'information est codée par des symboles binaires. On adjoint à cette information une redondance, constituée de symboles supplémentaires, de telle sorte que, si certains symboles se trouvent altérés lors de la transmission, un algorithme de décodage permette quand même de reconstituer l'information initiale.

Les codes les plus simples à mettre en œuvre sont des codes linéaires. L'information est portée par un mot binaire de longueur k, c'est-à-dire un vecteur constitué de k composantes égales à 0 ou 1, appelé *vecteur d'information*. Le codage consiste à calculer la redondance comme une fonction linéaire du vecteur d'information. Ce qu'on appelle le code est l'ensemble de tous les vecteurs de dimension n produits à partir des 2^k vecteurs d'information. Les vecteurs qui appartiennent au code ne recouvrent qu'une partie infime des 2^n vecteurs binaires possibles. Lorsque ces données sont transmises, certaines composantes peuvent être altérées, et le vecteur reçu n'est alors plus identique au vecteur émis. L'opération de décodage consiste à trouver l'élément du code qui a le plus de symboles en commun avec le vecteur reçu. Ce sera celui-ci qui aura été le plus vraisemblablement émis. La capacité de décodage d'un code est le nombre maximal de symboles qui peuvent être altérés tout en permettant encore de retrouver l'information initiale.

Décoder est en général un problème difficile, exigeant quasiment une recherche exhaustive de tous les motifs d'erreur. Elle devient impraticable si la dimension du code est trop grande. Heureusement, il existe des codes suffisamment structurés pour qu'on puisse décoder de manière efficace. McEliece a eu l'idée d'utiliser ce principe pour réaliser un chiffrement à clé publique.
- La clé privée est un code dont on connaît un algorithme de décodage.
- La clé publique est l'ensemble des formules linéaires de calcul de la redondance.

Pour certaines familles de codes, appelés codes alternants, l'algorithme de décodage existe, mais ne se déduit pas facilement de ces formules. Chiffrer une information consiste à la coder, puis à lui adjoindre une erreur aléatoire compatible avec les capacités de décodage du code. Pour déchiffrer le cryptogramme, on lui applique l'algorithme de décodage. Cela permet d'éliminer l'erreur transmise et de retrouver l'information du message en clair. L'information restera inaccessible à qui ne sait pas décoder.

Le système de McEliece n'est pas utilisé en pratique, en particulier parce que ses clés publiques ont une taille prohibitive de plusieurs centaines de kilo-octets. Il est cependant très étudié, car il constitue l'une des rares alternatives envisageables au cas où des progrès notables dans la factorisation des entiers ou dans le calcul des logarithmes rendraient les systèmes existants obsolètes, par exemple si des avancées technologiques majeures rendaient possible la réalisation d'un calculateur quantique (voir paragraphe 2 page 159). Pour cette raison, ce type de cryptographie est appelée *post-quantique* (*post-quantum cryptography*).

2 La signature numérique

Une signature non contestable est au cœur de bien des relations contractuelles. L'avènement de la cryptographie à clé publique a permis la signature numérique des documents. Ce processus est par nature asymétrique : le signataire signe le document en y adjoignant une donnée liée à celui-ci qu'il est seul à pouvoir calculer et que tous peuvent vérifier. Cette fonction est également indispensable pour authentifier les correspondants qui échangent une clé avec le protocole Diffie-Hellman afin de se prémunir du problème de l'intrus. Les signatures numériques servent également à certifier l'origine des clés publiques dans les infrastructures de gestion des clés publiques (voir le paragraphe 1 page 109).

La signature numérique associe deux fonctions : la fonction de calcul de la signature qui dépend du document et d'une clé privée, et la fonction de vérification qui, à partir du document, de sa signature et d'une clé publique, renvoie une réponse binaire qui atteste ou non de la validité de la signature (Fig. 3.6). Ainsi, la signature à clé publique assure :

- le service d'intégrité : toute modification du document entraîne l'invalidité de la signature ;
- le service d'authentification : la vérification de la signature assure qu'elle a été produite par le seul détenteur de la clé privée ;
- le service de non-répudiation : la signature ne peut être produite par personne d'autre que le détenteur de la clé privée ; le signataire ne peut pas nier avoir signé le document.

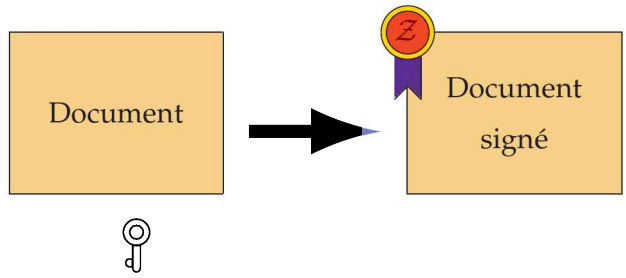


FIGURE 3.6. Signature à clé publique. La production de la signature nécessite une clé privée et change à chaque document. Sa vérification ne requiert qu'une clé publique accessible à tous.

Le processus de signature numérique suit la méthode dite *hache-puis-signe* qui opère en deux temps. Tout d'abord, le document à signer est réduit en une donnée de taille fixe par une fonction appelée *fonction de hachage* pour produire ce qu'on appelle une *empreinte* du document. Cette empreinte est ensuite soumise à la fonction de signature pour produire la donnée appelée *signature*, qui est destinée à accompagner le document afin d'authentifier le signataire. Le destinataire, pour vérifier cette signature, procède de manière similaire. Le document est soumis à la fonction de hachage, puis l'empreinte calculée est présentée à la fonction de vérification (Fig. 3.7).

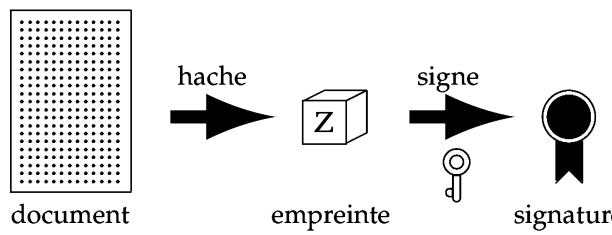


FIGURE 3.1. La méthode *hache-puis-signe*. Le document à signer est haché pour produire une empreinte de taille fixe. Ensuite il est fait appel à la clé privée pour signer cette empreinte et produire la signature du document. La fonction de hachage est publique et ne fait appel à aucun paramètre secret.

3.1 Fonction de hachage

Pour les applications cryptographiques, une fonction de hachage idéale est ce qu'on appelle un *oracle aléatoire*. L'empreinte d'un nouveau message doit sembler avoir été tirée au hasard, sans aucune relation avec les valeurs déjà calculées sur des messages antérieurs. Cette situation idéale n'est pas réaliste. En pratique, on se contente des propriétés suivantes :

1. l'empreinte d'un message doit être calculable de manière efficace ;

2. une fonction de hachage doit être à sens unique. Pour presque toute empreinte, il ne doit pas être possible de trouver un message lui correspondant. Les seules valeurs pour lesquelles cela doit rester possible sont les empreintes de messages déjà calculées ;
3. une fonction de hachage doit aussi résister au *second antécédent*. Pour un message m donné ayant pour empreinte h, il ne doit pas être possible de trouver un second message m_1 ayant la même empreinte que m ;
4. enfin, une fonction de hachage doit résister aux collisions, ce qui signifie qu'il doit être pratiquement impossible de trouver deux messages ayant la même empreinte.

En particulier, la taille de l'empreinte doit être suffisante pour qu'il n'y ait qu'une chance infime de résoudre les problèmes ci-dessus en piochant la réponse au hasard. Ceci impose en pratique une taille supérieure à 128, voire 160 symboles binaires.

Encadré 3.5. Le paradoxe des anniversaires.

La probabilité que deux personnes aient le même jour anniversaire au sein d'une assemblée est supérieure à 0,5 dès que l'assemblée comprend plus de 23 personnes. Cette valeur étonnamment basse a donné son nom au paradoxe des anniversaires.

Plus généralement, la probabilité d'obtenir deux objets identiques, c'est-à-dire une collision, lors du tirage avec remise de m objets dans une urne qui en contient n, vaut environ $e^{-\frac{m^2}{2n}}$. La valeur devient significative dès que le nombre d'objets tirés approche la racine carrée du nombre total d'objets. *A contrario*, si on veut rendre négligeable la probabilité de collision, le nombre d'objets doit dépasser le carré du nombre de tirages.

Pour trouver une collision dans une fonction de hachage, un adversaire peut calculer n empreintes de n messages différents. S'il en existe deux égales parmi les n valeurs calculées, on aura trouvé une collision à la fonction de hachage. Si on veut que la probabilité soit inférieure à un millionième lors du tirage de mille milliards de messages au hasard, il faut que les empreintes comprennent plus de 90 symboles binaires.

Les fonctions de hachage standard ont pour noms SHA-1, SHA-256, SHA-512 (*Secure Hash Algorithm*), MD-5 (*Message Digest*), RIPEMD (*RACE Integrity Primitives Evaluation Message Digest*, RACE – *Research for Advanced Communications in Europe* étant un programme de recherche européen lancé en 1995 sur les télécommunications). Il existe également un mode opératoire des fonctions de chiffrement par bloc comme le DES ou l'AES pour définir une fonction de hachage. Le 2 octobre 2012, le NIST américain a standardisé la fonction de hachage *Keccak* comme nouveau standard SHA-3 en remplacement des précédents.

3.2 Signature RSA

La signature RSA est la première à avoir été proposée dès 1978. Elle consiste à appliquer la fonction RSA décrite pour le chiffrement au paragraphe 2.5 page 54 en utilisant l'exposant privé pour produire la signature et l'exposant public pour la vérifier.

Si la clé privée RSA est constituée du module n et de l'exposant privé d, la signature RSA de l'empreinte h d'un document est calculée par :

$$\sigma = s_{\text{RSA}}(h,d,n) = h^d \mod n.$$

Si la clé publique est constituée du même module n et de l'exposant public e, la signature σ est vérifiée en calculant $h' = \sigma^e \mod n$ et en vérifiant si cette quantité est ou n'est pas égale à l'empreinte h du message. Si $h = h'$, la signature σ est considérée comme valide et, dans le cas contraire, elle est considérée comme invalide.

Encadré 3.6. Exemple de signature RSA.

Soit $p = 113$ et $q = 131$ deux nombres premiers. Ces facteurs engendrent un module RSA égal à $n = p \times q = 14\,803$. Soit $e = 3$ l'exposant public. Le plus petit multiple commun à $p-1$ et $q-1$ est $\lambda(n) = \text{ppcm}(p-1,q-1) = 3\,640$. L'exposant privé correspondant à e est l'inverse de e modulo $\lambda(n) = 3\,640$. Il vaut $d = 2\,427$.

Si l'empreinte d'un document à signer est $h = 10\,000$, la signature du document vaut $\sigma = h^d \mod n = 10\,000^{2\,427} \mod 14\,803 = 618$.

Pour vérifier la signature, on calcule $\sigma^e \mod n = 618^3 \mod 14\,803 = 10\,000$ et on vérifie que la valeur obtenue vaut bien l'empreinte h du message.

La signature RSA est très utilisée, en particulier dans les cartes bancaires de paiement pour vérifier la validité des données qu'elles contiennent (voir paragraphe 2 page 110).

3.3 *Digital Signature Algorithm*, DSA

L'algorithme de signature numérique DSA est un algorithme standardisé par le NIST américain sous la norme FIPS 186-2. Il repose sur le problème du logarithme. Les paramètres de cet algorithme sont les données publiques suivantes :

- un grand nombre premier p, par exemple de 1 024 chiffres binaires ;
- un diviseur q de $p-1$ de 160 chiffres binaires ;
- un élément g de l'ensemble des entiers modulo p, différent de 1 et tel que $g^q = 1$ modulo p. On sait qu'il existe toujours un élément ayant cette propriété. Considérons par exemple $p = 23$ et le diviseur $q = 11$ de 22. L'élément $g = 2$ convient, car $g^{11} = 1$ modulo 23.

La clé privée est un nombre s compris entre 2 et $q-1$. La clé publique correspondante est g^s modulo p. Retrouver la clé privée à partir de la clé publique revient à calculer son logarithme en base g modulo p, et l'entier p est choisi assez grand pour que cela soit considéré comme pratiquement impossible.

La signature d'un document consiste à effectuer les opérations suivantes :

1. Choisir un nombre aléatoire r, compris entre 1 et $q-1$.
2. Calculer l'empreinte $h(m)$ du message à signer.
3. Calculer $z = g^r$ modulo p. On pose $x = z \mod q$.
4. Calculer $y = \dfrac{h(m)+xs}{r}$ modulo q.

La signature du document m est constituée du couple $\sigma = (x,y)$. L'élément y est une façon de masquer l'empreinte $h(m)$ par un calcul qui fait appel à la clé privée. Comme cette clé privée ne doit pas être dévoilée, elle est camouflée par la division par r qui est un élément aléatoire. La composante x de la signature est un élément qui permet de vérifier la validité de la signature en invoquant uniquement la clé publique. Pour vérifier la signature $\sigma = (x,y)$ d'un document m, on procède aux opérations suivantes :

1. Calculer $\tilde{z} = g^{h(m)/y} \times \left(g^s\right)^{x/y}$ modulo p. Ce calcul ne fait appel qu'aux éléments x et y de la signature et aux paramètres publics g et g^s. Les quotients $h(m)/y$ et x/y présents dans les exposants sont calculés modulo q. Pour une signature correcte, la valeur de ce calcul est $\tilde{z} = g^r$ modulo p.
2. La signature est valide si $\tilde{z} \mod q = x$ et invalide dans le cas contraire.

Encadré 3.7. Un exemple de signature DSA.

Posons comme paramètres connus de tous $p = 23$, $q = 11$ et $g = 2$. Si la clé privée du signataire est $s = 5$, la clé publique est $2^5 = 9$ modulo 23. Supposons que l'empreinte du message à signer est $h(m) = 10$. Pour signer, le signataire choisit un nombre aléatoire compris entre 1 et 11, par exemple $r = 7$, puis il calcule $z = g^r = 2^7 = 13$ modulo 23. La valeur de x est $x = z \mod q = 13 \mod 11 = 2$. Le second calcul est $y = \dfrac{10+2\times 5}{7}$ modulo 11, soit $y = 6$. La signature du document est constituée du couple :

$$\sigma = (2,6).$$

Pour vérifier la signature, le destinataire calcule $\tilde{z} = 2^{10/6} \times 9^{2/6} = 2^9 \times 9^4$ modulo 23, soit $\tilde{z} = 13$. La signature est valide, car $13 \mod 11 = 2$ qui est la première composante de la signature.

Pour assurer la sécurité de cette signature, c'est-à-dire pour empêcher un faussaire de pouvoir forger une fausse signature, il est important que la donnée

aléatoire r ne soit ni prédictible, ni révélée, ni utilisée plusieurs fois pour signer des messages différents.

4 L'authentification

« *Introduisez votre code confidentiel à l'abri des regards indiscrets.* »

Le moyen le plus simple de prouver son identité est de révéler un secret qu'on est supposé être seul à détenir. Malheureusement, dès que le secret est révélé, ce n'en est plus un. Nous faisons tous confiance aux distributeurs automatiques de billets pour effacer ce que nous leur dévoilons le temps d'un retrait. Dans les communications par ordinateur, le service d'authentification est essentiel, davantage même que celui de confidentialité. Lors d'un achat sur internet, il est primordial, pour que la transaction se fasse en toute confiance, que les partenaires sachent avec certitude à qui ils s'adressent. Le contenu de la transaction, lui, est rarement secret.

4.1 L'authentification dynamique

La signature numérique d'un document est un premier niveau d'authentification. La signature d'un document identifie le signataire, mais pas celui qui remet le document. Pour s'assurer qu'on s'adresse à la bonne personne, il est nécessaire de procéder à un échange interactif fonctionnant sur le schéma *défi/réponse*.

Un système d'authentification met en jeu deux acteurs : un *vérificateur* qui veut s'assurer de l'identité de son interlocuteur et le *prouveur* qui, comme ce nom l'indique, doit prouver son identité. Ce dernier dispose pour cela d'une information secrète, en l'occurrence une clé privée, à l'aide de laquelle il peut signer toute donnée, par exemple son propre nom. Mais cela ne suffit pas. Il faut montrer qu'on est capable de signer n'importe quelle donnée. Aussi, lors d'une authentification dynamique, le vérificateur lance-t-il au prouveur le défi de signer une donnée tirée au hasard pour la circonstance. Le prouveur répond en la signant. Si la signature est valide, le vérificateur est assuré que le prouveur dispose bien de la clé privée qui a permis d'élaborer la signature. N'importe quel procédé de signature est utilisable à cette fin : le RSA, le DSA, etc.

« – *Es-tu capable de signer* `KINDER` ?
– *Bien sûr ! la signature est* `ABCDUL`
– *C'est bon, tu peux passer !* »

Ce dialogue est interactif. La fois suivante, ce sera un autre défi imprévisible qui sera posé. Si une oreille indiscrète intercepte et note soigneusement cet échange, cela ne lui sera très probablement d'aucune utilité pour tenter de s'authentifier ultérieurement.

Avec l'authentification dynamique, le vérificateur s'assure que le correspondant dispose bien de l'élément secret qui lui permet d'élaborer les réponses. Mais le secret s'use à force de s'en servir. Plus on s'authentifie, et plus la probabilité qu'un défi ait déjà été posé augmente. Pour pallier cela, les cryptologues ont élaboré l'authentification sans divulgation. Il s'agit d'un échange qui permet de prouver la détention d'un secret sans divulguer aucune connaissance sur ce secret (*Zero Knowledge*).

Le protocole décrit ci-après est dû aux cryptologues Amos Fiat et Adi Shamir qui l'ont proposé en 1987. Rappelons que, lorsque l'entier n est le produit de deux nombres premiers p et q, il est tout aussi difficile de factoriser l'entier n que d'extraire les racines carrées modulo n. Le prouveur est identifié par un entier i qui code son identité. Une autorité, seule détentrice de la factorisation de l'entier n, distribue aux individus un secret s égal à une racine carrée de leur identité modulo n. Un individu voulant prouver qu'il est bien i doit prouver qu'il détient, sans le révéler, le secret s d'une racine carrée de son identité i modulo n.

Pour cela, il tire un nombre au hasard r qui masquera son secret en fournissant $r \times s$ au vérificateur. Mais pour pouvoir vérifier que cette information correspond bien à l'identité i, il faut connaître r^2. L'idée est donc de transmettre $x = r^2$, puis $y = r \times s$. À la réception de ces données, le vérificateur peut élever y au carré modulo n, ce qui lui donnera $y^2 = (r \times s)^2 = r^2 \times s^2 = x \times i$. Ainsi, si $y^2 = x \times i$, le vérificateur pourra être convaincu que son interlocuteur dispose bien du secret s, égal à une racine carrée de i modulo n. Il est bien celui qu'il prétend être en montrant qu'il détient l'information secrète sans la révéler.

Le problème avec cette façon de faire est que le prouveur peut tricher. Au lieu de procéder comme indiqué ci-dessus, il peut satisfaire le vérificateur en lui fournissant des données x et y qui vérifient simplement $y^2 = x \times i$. Pour cela, il tire y au hasard, et calcule $x = y^2/i$ modulo n. Avec ces données, le vérificateur sera satisfait et n'aura pas vu la tricherie. Pour l'empêcher, le vérificateur doit demander de façon imprévisible de révéler le masque r, ce qui lui permet de vérifier que la donnée x résulte bien du carré de r.

On aboutit au protocole à trois passes au cours duquel le vérificateur demande aléatoirement au prouveur de montrer soit qu'il connaît le secret, soit qu'il ne triche pas :

1. *Engagement :* le prouveur choisit r au hasard, et fournit au vérificateur $x = r^2$;
2. *Défi :* le vérificateur met le prouveur au défi de révéler soit r, soit $r \times s$;
3. *Réponse :* le prouveur fournit la réponse demandée au vérificateur.

La stratégie d'un tricheur est d'essayer d'anticiper le défi posé par le vérificateur au moment de l'engagement :

- s'il devine qu'on lui demandera de révéler $r \times s$, alors il peut s'engager sur $x = y^2/i$. La réponse y conviendra au vérificateur ;
- s'il devine qu'on lui demandera de révéler r, alors il peut s'engager sur $x = r^2$. La réponse r conviendra au vérificateur.

Un tricheur a une chance sur deux de réussir son test, mais aussi une chance sur deux d'échouer car, s'il a mal anticipé le défi, il ne pourra pas y répondre correctement sans avoir à calculer une racine carrée modulo n dans l'ignorance de sa factorisation, ce qui est inaccessible avec nos moyens actuels. Le prouveur n'a finalement qu'une chance sur deux de fournir une réponse qui satisfera le vérificateur. Ce dernier doit itérer ces trois passes jusqu'à être convaincu de l'identité de son correspondant. Un tricheur ne pourra pas prouver longtemps qu'il connaît le secret sans le détenir réellement.

Une propriété remarquable de ce protocole est qu'un échange d'authentification ne divulgue rigoureusement aucune information sur le secret détenu par le prouveur. En effet, cet échange peut parfaitement être réalisé par un simulateur qui ne connaît pas le secret. Il suffit de connaître le défi pour s'engager de manière à pouvoir répondre sans faire appel au secret. Et comme on peut créer un échange fictif, mais valide, sans connaître le secret, c'est que l'échange réel entre le prouveur et son vérificateur ne divulgue, lui non plus, aucune information. C'est pour cette raison que ce type de protocole à trois passes est appelé « sans divulgation de connaissance ».

5 Les courbes elliptiques

5.1 Une loi de groupe intéressante

Jusqu'aux début des années 1980, tous les algorithmes connus de calcul du logarithme modulo p avaient une complexité exponentielle, ce qui convenait pour assurer la sécurité des mécanismes à clé publique reposant sur la difficulté de ce problème. En 1983, la découverte d'un algorithme sous-exponentiel de calcul de logarithme dans l'anneau des entiers modulo p a conduit les cryptographes à se tourner vers d'autres structures mathématiques. Le groupe des points d'une courbe elliptique a été proposé en 1985 comme alternative. L'avantage apporté par l'utilisation de cette structure est que, encore aujourd'hui, le meilleur algorithme de calcul du logarithme sur une courbe elliptique a une complexité exponentielle. Pour une sécurité équivalente, les paramètres peuvent avoir une taille bien moindre : 256 symboles binaires au lieu de 2 048, ce qui conduit à des clés plus courtes et des calculs cinq à dix fois plus rapides.

Encadré 3.8. Théorie des groupes.

En algèbre, un groupe est un ensemble E, muni d'une opération interne, notée ici $*$, et qui vérifie les propriétés suivantes :

- l'opération est associative : $(x * y) * z = x * (y * z)$;
- elle admet un élément neutre, noté ici e, qui vérifie $x * e = e * x = x$ pour tout élément x de E ;
- tout élément x de E a un symétrique x^{-1} qui est tel que $x * x^{-1} = x^{-1} * x = e$.

Si l'opération $*$ est commutative, le groupe est dit commutatif.

Les entiers non nuls modulo p, munis de la multiplication, forment un groupe commutatif. Ce groupe a le premier été utilisé en cryptographie à clé publique pour réaliser des opérations de chiffrement. La sécurité repose sur la difficulté de résoudre le problème du logarithme. Un autre groupe populaire chez les cryptographes est l'ensemble des points d'une courbe elliptique.

Les courbes elliptiques ont leur origine dans l'étude des aires des arcs d'ellipses. Ce sont des courbes définies par une équation polynomiale du troisième degré de la forme $y^2 = x^3 + ax + b$. On adjoint un point supplémentaire, appelé point à l'infini et noté P_∞, qu'il faut imaginer comme le point d'intersection des droites verticales du plan. On exige également que le discriminant de la courbe, $\Delta = 4a^3 + 27b^2$ ne soit pas nul, ce qui assure que la courbe est lisse, c'est-à-dire ne comporte aucun croisement ni rebroussement ou autre singularité.

Pour une utilisation cryptographique, les coordonnées, ainsi que les coefficients a et b de l'équation de la courbe ne sont pas des nombres réels, mais des entiers modulo un grand nombre premier. Mais cela ne change pas grand chose quant aux formules qui sont applicables. La propriété remarquable de ces courbes, qui fait qu'elles se prêtent à un usage cryptographique, est l'existence d'une façon de composer deux points pour en obtenir un troisième. Cette opération est notée comme une addition. Elle a des propriétés très similaires à la multiplication des entiers modulo p.

Comme la courbe est définie par une équation du troisième degré, toute droite qui passe par deux points de la courbe la coupe nécessairement en un troisième point. La définition de la somme de deux points obéit aux règles suivantes :

- le point à l'infini est un élément neutre ;
- trois points alignés sur la courbe ont une somme nulle.

La figure 3.8 montre deux courbes dessinées dans le plan réel ainsi que l'illustration de la somme de deux points. Une droite verticale qui passe par deux points de la courbe la coupe encore au point à l'infini. Deux points sont opposés lorsqu'ils sont symétriques par rapport à l'axe horizontal.

L'ensemble des points d'une courbe elliptique, muni d'une addition ainsi définie, forme un groupe commutatif. On peut additionner des points avec des propriétés similaires à celles de l'addition des nombres. La somme de deux points s'obtient à l'aide de formules d'addition qui opèrent sur les coordonnées des

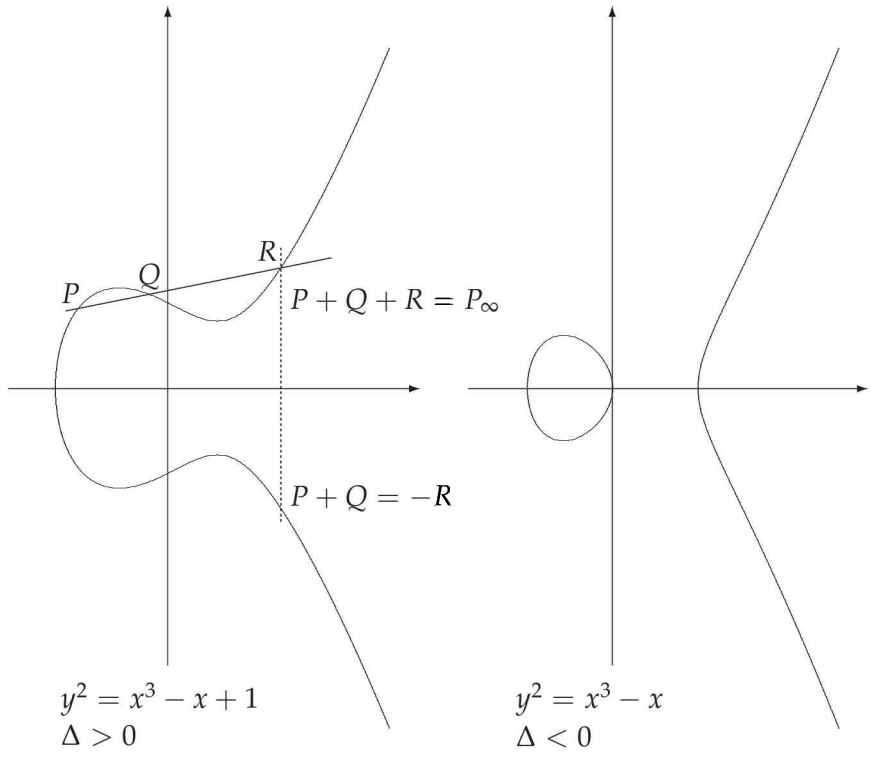


FIGURE 3.8. Représentation de courbes elliptiques dans le plan réel et illustration de la loi d'addition des points. Pour additionner les points P et Q de la courbe, prolonger la droite (PQ). Elle coupe la courbe au point R. La somme de P et Q est le point symétrique de R par rapport à l'axe horizontal.

points. Multiplier un point par un entier est l'analogue de l'élévation d'un nombre à une puissance : $\ell P = \underbrace{P + \cdots + P}_{\ell \text{ fois}}$. On continue d'appeler *logarithme* l'opération qui consiste à retrouver le multiplicateur ℓ à partir de la donnée d'un point Q sur la courbe, égal à ℓP.

Tous les mécanismes à clé publique qui utilisent la fonction puissance, comme l'échange Diffie-Helman, le chiffrement ElGamal et la signature DSA peuvent se transposer de manière analogue au groupe des points d'une courbe elliptique. Ils sont notés en adjoignant le préfixe EC pour *Elliptic Curve* : ECDSA, EC-DH, EC-ElGamal, etc.

5.2 La cryptographie bilinéaire

Les courbes elliptiques ont une arithmétique particulièrement riche. Outre la loi de groupe, elles disposent d'une fonction d'appariement des points avec

des propriétés de bilinéarité qui ont permis de réaliser de nouvelles fonctions cryptographiques.

Encadré 3.9. L'appariement de Weil sur une courbe elliptique.

Sur une courbe elliptique, un point de n-torsion est un point qui vérifie $nP = P_\infty$. La somme de deux tels points est aussi un point de n-torsion, de telle sorte qu'ils forment un sous-groupe de la courbe, appelé groupe de n-torsion.

Il existe sur le groupe de n-torsion d'une courbe elliptique une fonction e_n, appelée appariement de Weil, qui à deux points associe une valeur qui est une racine n^e de l'unité, c'est-à-dire qui élevée à la puissance n vaut 1.

L'appariement de Weil a les propriétés remarquables suivantes :

- il est bilinéaire. Pour tout point P, Q et R de n torsion, $e_n(P+Q,R) = e_n(P,R) \times e_n(Q,R)$ et $e_n(P,Q+R) = e_n(P,Q) \times e_n(P,R)$;
- il est normal, c'est-à-dire pour tout point P de n-torsion, $e_n(P,P) = 1$;
- il est antisymétrique, c'est-à-dire pour tout point P et Q de n-torsion, $e_n(P,Q) = e_n(Q,P)^{-1}$.

Cet appariement est exploité pour réaliser de nouvelles fonctions cryptographiques dans un domaine en pleine expansion appelé la cryptographie bilinéaire.

5.3 Le chiffrement avec l'identité

Dans les systèmes à clé publique classiques, la clé privée est choisie *a priori*, et la clé publique est le résultat de l'application d'une fonction à sens unique sur la clé privée. À l'opposé, dans un système de chiffrement avec l'identité, c'est la clé publique qui est choisie, et la clé privée calculée. Cette notion a été introduite dans son principe dès 1984 par Shamir, mais ce n'est qu'en 2001 que des réalisations effectives ont été proposées. L'une d'entre elles utilise les propriétés de bilinéarité des fonctions d'appariement sur une courbe elliptique.

La clé publique étant choisie, elle peut être tout simplement le nom du destinataire, ou n'importe quelle donnée qui est indubitablement liée à celui-ci, comme son numéro de téléphone ou son adresse mail. On peut aussi imaginer toute sorte d'autres informations adjointes à l'identité du destinataire comme par exemple des dates de validité, des informations de délégation, des informations de gestion des droits, etc.

Mais pour que la clé privée reste privée, son calcul doit reposer sur un secret. Ce mécanisme ne peut fonctionner qu'avec une autorité qui, à partir d'une clé privée centrale détenue par elle seule, calcule les clés privées des usagers et les leur transmet. Le chiffrement vers un destinataire utilise la clé publique de l'autorité centrale. L'avantage est que cela dispense de déployer une infrastructure de gestion des clés publiques pour certifier les clés publiques des destinataires (voir paragraphe 1 page 109). Mais l'autorité centrale qui

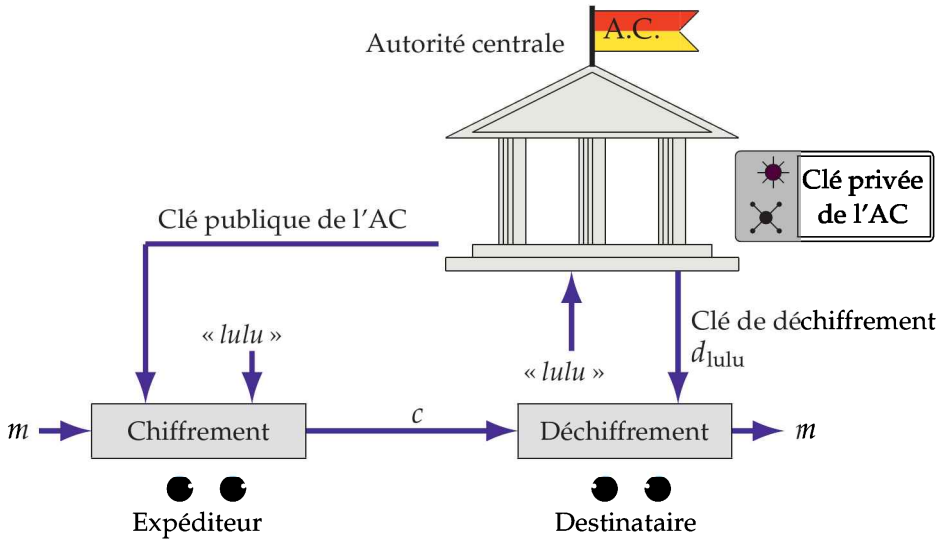


FIGURE 3.8. Le chiffrement avec l'identité. Une autorité centrale délivre, grâce à sa clé privée, des clés de déchiffrement aux destinataires construites à partir de leur identité. Les expéditeurs chiffrent avec l'identité du destinataire et la clé publique de l'autorité centrale. Les destinataires déchiffrent avec leur propre clé privée de déchiffrement fournie par l'autorité centrale.

délivre les clés privées peut déchiffrer tous les messages. Ce n'est clairement pas envisageable dans un système ouvert comme l'internet, mais cela peut être acceptable au sein d'une organisation placée sous l'autorité d'un chef ou d'un directeur.

5.4 Un exemple concret

Les ingrédients pour construire un schéma de chiffrement avec l'identité sont :

- un nombre premier q ;
- une courbe elliptique bien choisie ;
- un point P de cette courbe elliptique qui est d'ordre q, c'est-à-dire tel que $q\,P = P_\infty$;
- une application bilinéaire convenable β qui prend deux multiples de P et leur associe une racine q^{e} de l'unité.
  La bilinéarité signifie que pour tout couple de points A et B, tous deux multiples de P, et tout couple d'entiers a et b, on a $\beta(aA,bB) = \beta(A,B)^{ab}$;
- une fonction h qui permet de réduire les noms des destinataires en un point de la courbe, multiple de P, qui l'identifiera. Par exemple $Q_{\text{lulu}} = h(lulu)$;
- un algorithme de chiffrement symétrique E qui opère avec des clés qui sont racines q^{e} de l'unité.

Encadré 3.10. Une fonction bilinéaire convenable avec l'appariement de Weil.

Considérons un nombre premier p qui s'exprime comme $p = q(q-1)+1$ où l'entier q est également premier. Par exemple $p = 43$ s'exprime comme $43 = 7 \times 6 + 1$ avec $q = 7$ qui est aussi premier. Dans ce cas, l'entier $j = q - 1$ est une racine cubique de 1 modulo p, c'est-à-dire vérifie $j^3 = 1$ modulo p. Dans notre exemple, on a $j = 6$ et $6^3 = 216 = 5 \times 43 + 1$ vaut bien 1 modulo 43.

Avec un tel entier p, on peut toujours trouver une courbe elliptique $\mathcal{E}$ d'équation $y^2 = x^3 + b$, qui contient exactement q^2 points dont les coordonnées sont des entiers modulo p. Avec $p = 43$, la courbe d'équation $y^2 = x^3 + 3$ contient 36 points aux coordonnées modulo 43. Et les points de cette courbe sont tous des points de q–torsion, c'est-à-dire vérifient $qP = P_\infty$.

On définit l'application φ sur la courbe $\mathcal{E}$ en associant au point P de coordonnées (x,y) le point $\varphi(x,y)$ de coordonnées (jx,y). Ce point appartient aussi à la courbe $\mathcal{E}$. L'application φ a la propriété remarquable d'être compatible avec la loi d'addition des points. Elle vérifie $\varphi(P+Q) = \varphi(P) + \varphi(Q)$. L'application β définie par :

$$\beta(P,Q) = e_q(P,\varphi(Q)),$$

où e_q est l'appariement de Weil, convient pour réaliser le chiffrement avec l'identité.

Calcul des clés de déchiffrement des destinataires

Les clés de déchiffrement des destinataires sont calculées par l'autorité centrale. Celle-ci dispose d'une clé privée s qui est un entier s compris entre 1 et $q-1$. La clé publique correspondante est le point $R = s\,P$. La clé de déchiffrement du destinataire D, identifié par le point Q_D, est le point $D_D = s\,Q_D$.

Chiffrement d'un message

Pour chiffrer un message à destination du destinataire D, identifié par le point Q_D, l'expéditeur utilise la clé publique R de l'autorité centrale. Les étapes du chiffrement sont les suivantes :

1. Choisir un nombre r au hasard compris entre 1 et $q-1$.
2. Calculer la clé de session $k = \beta(Q_D,R)^r$. En raison de la bilinéarité de la fonction β, cette clé est aussi égale à $\beta(Q_D,sP)^r = \beta(Q_D,P)^{rs}$.
3. Chiffrer le message m avec l'algorithme symétrique E en utilisant la clé de session k. On obtient le message chiffré par la relation $c = E_k(m)$.
4. Le cryptogramme est constitué du couple $(r\,P,c)$.

Dans ce cryptogramme, la donnée c contient l'information du message, dissimulée par chiffrement avec l'algorithme E et la clé de session k. Le préfixe $r\,P$ est une information additionnelle, un indice, qui permettra au destinataire et à lui seul de reconstituer la clé de session.

Déchiffrement d'un message

Le destinataire reçoit un couple (T,c). Il doit tout d'abord retrouver la clé de session. Pour cela, il dispose de sa clé de déchiffrement D_D. Il calcule $\beta(D_D,T)$. En raison de la bilinéarité de la fonction β, cette valeur est égale à $\beta(sQ_D,rP) = \beta(Q_D,P)^{rs}$ qui a bien la même valeur que la clé de session k calculée par l'expéditeur. Une fois cette clé de session connue, il ne lui reste plus qu'à l'utiliser pour déchiffrer c avec l'algorithme symétrique E. Il retrouve le message clair m.

6 L'algorithmique de la cryptographie à clé publique

L'usage généralisé de la cryptographie à clé publique a créé de nouveaux besoins en calculs. Il faut maintenant être capable d'effectuer des opérations arithmétiques sur des nombres de plusieurs centaines de chiffres avec des dispositifs matériels aussi petits qu'une carte à puce. La génération sans cesse croissante des couples clé publique-clé privée, a engendré un nouveau besoin : il est nécessaire de produire rapidement et en grande quantité des nombres premiers aléatoires. Pour réaliser cela sans que l'utilisateur ne soit importuné par des temps d'attente prohibitifs, il a fallu développer une algorithmique efficace.

6.1 Multiplier

La multiplication modulo un entier n est l'opération de base utilisée dans tous les mécanismes à clé publique reposant sur l'arithmétique. Étant donné deux entiers a et b, il s'agit de trouver la valeur de $a \times b \mod n$. Le résultat est la réduction modulo n du produit de a par b. Les algorithmes utilisés par les ordinateurs pour effectuer les quatre opérations sont très proches de ceux que nous avons tous appris à l'école. Nous connaissons les tables de multiplication sur les nombres de 0 à 9. Et lorsque nous avons à multiplier un nombre de plusieurs chiffres, nous savons comment nous ramener à une série de multiplications que nous connaissons par les tables. La façon d'opérer des ordinateurs est très proche dans son principe. Les premiers micro-ordinateurs opéraient sur les mots constitués de huit symboles binaires, appelés *octets* et qui peuvent représenter un nombre compris entre 0 et 255. Aujourd'hui, les machines travaillent sur des mots de 32, voire 64, symboles binaires, ce qui change l'échelle des nombres manipulés, mais pas le principe des calculs. Les opérations élémentaires sur les mots-machine sont réalisées par le matériel du cœur de calcul, sous forme de câblage de circuits logiques. Elles correspondent aux tables d'addition et de multiplication que nous avons apprises. Nous savons calculer dans la numération en base dix, les

processeurs opèrent dans une base de numération égale à 2^8 ou 2^{32}, voire 2^{64} selon leur architecture matérielle.

Les nombres dépassant la base de numération s'écrivent comme une liste de chiffres, et pour multiplier deux tels nombres un algorithme réalise la combinaison appropriée d'opérations élémentaires. Pour multiplier deux nombres de k chiffres, nous réalisons k^2 multiplications élémentaires et quelques additions. Pour évaluer la complexité du calcul, les additions peuvent être ignorées, car ce sont des opérations bien plus rapides que les multiplications. Nous en concluons que l'algorithme de multiplication scolaire a une complexité quadratique. En 1960, le mathématicien russe Anatolii Alexevich Karatsuba a réduit la complexité de la multiplication à $k^{1,58\cdots}$ à partir d'une idée de Gauss pour multiplier les nombres complexes. Mais cet algorithme nécessite de réaliser davantage d'additions que l'algorithme scolaire et n'est intéressant qu'à partir de nombres de plusieurs centaines de chiffres.

Encadré 3.11. La multiplication de Karatsuba.

Multiplions deux nombres de deux chiffres : $(10a + b)$ par $(10c + d)$. La méthode usuelle repose sur l'égalité $(10a + b)(10c + d) = 100ac + 10(ad + bc) + bd$. En effectuant les quatre multiplications ac, ad, bc et bd, le résultat est acquis. L'idée de Karatsuba est d'obtenir le nombre des dizaines $ad + bc$ en une seule multiplication au lieu de deux, en exploitant l'égalité $ad + bc = (a + b)(c + d) - ac - bd$. Les produits ac et bd étant déjà calculés, seule la multiplication $(a + b)(c + d)$ est nécessaire.

Le produit de deux nombres de deux chiffres s'obtient en seulement trois multiplications élémentaires au lieu de quatre. Cette performance est obtenue au prix de deux additions et deux soustractions supplémentaires, dont la complexité est d'autant plus négligeable devant celle des multiplications que les nombres traités sont grands.

Poursuivant l'idée de Karatsuba, le Russe André Toom et le Canadien Stephen Cook ont encore amélioré l'efficacité de la multiplication des très grands nombres pour obtenir une complexité asymptotique en $k^{1+\varepsilon}$, où ε est un réel positif qu'on peut rendre aussi proche de zéro que l'on veut, mais intéressant pour des nombres d'autant plus grands que ε est petit.

Le meilleur algorithme connu actuellement pour multiplier deux grands entiers est dû à Arnold Schönage et Vloker Strassen en 1971, et repose sur la transformation de Fourier. L'application de cette transformation sur une table de k chiffres conduit à une représentation des nombres pour lesquels le produit de deux nombres s'obtient simplement en multipliant chaque composante de la table, ce qui ne nécessite que k multiplications. L'étape dominante du calcul réside alors dans celui de la transformation de Fourier. Un algorithme rapide le réalise en $k \ln k \ln \ln k$ opérations. Une telle complexité est dite *quasi linéaire*.

Pour réaliser la multiplication modulo n, il faut encore réduire le résultat modulo n, et pour cela, effectuer la division par n pour ne retenir que le reste. Notre expérience d'écolier nous a appris la difficulté de la division. Il n'y a pas de méthode directe pour obtenir le quotient. Il nous faut tout d'abord estimer le quotient, puis le corriger lorsqu'il a été mal estimé. Une algorithmique plus fine peut limiter les corrections dans l'estimation du quotient, mais ne les élimine pas, ce qui rend l'opération de division relativement lente. Pour cette raison, deux approches ont été développées pour remplacer la division par des multiplications.

Une première approche, due à Paul Barrett en 1986, consiste grossièrement à effectuer une multiplication par l'inverse de l'entier n pour remplacer la division par n. Dans une opération cryptologique, toutes les opérations se font modulo le même entier, et le calcul de son inverse peut être réalisé une fois pour toute. La réduction modulo n se réalise alors en effectuant deux multiplications.

Encadré 3.12. Réduction de Barrett.

Considérons une base de numération B et un nombre x qui s'exprime avec k chiffres dans cette base. Cela signifie que l'entier x est compris entre 0 et $B^k - 1$. Appelons n' le quotient de B^k par n. Alors la partie entière de $(x \times n')/B^n$ est une estimation du quotient de x par n à une unité près par défaut. Cela signifie que pour calculer le reste, la division de x par n, il suffit de calculer ce quotient estimé $\widetilde{q}$ et d'en déduire une estimation du reste par $\widetilde{r} = x - \widetilde{q}n$. Si le reste estimé est supérieur à n, alors une simple correction est nécessaire et la véritable valeur du reste est donnée par $r = \widetilde{r} - n$.

Par exemple, en base $B = 10$, divisons 33 par 8. Le nombre 33 comprend deux chiffres. Les premiers chiffres de l'inverse de 8 sont donnés par le quotient euclidien de 100 par 8 qui vaut 12.

- Le quotient estimé $\widetilde{q}$ est le quotient de 33×12 par 100, soit $\widetilde{q} = 3$.
- Le reste estimé est $\widetilde{r} = 33 - 3 \times 8 = 9$.
- Comme ce reste estimé est supérieur ou égal à 8, il faut corriger. Le reste vaut finalement $r = \widetilde{r} - 8 = 1$.

Noter que dans le calcul du quotient estimé, la division par B^n ne demande aucun calcul. Comme le diviseur est une puissance de la base de numération, elle ne consiste qu'en un décalage des chiffres, tout comme la division par 100 se fait de tête.

Une autre approche pour réaliser la multiplication de deux entiers modulo n sans effectuer de division a été introduite en 1985 par Peter Montgomery et est largement utilisée dans la plupart des réalisations de la cryptographie à clé publique en raison de sa grande efficacité. Elle repose sur une représentation des entiers modulo n qui n'est pas la représentation usuelle.

Notons B une base de numération sans diviseur commun avec l'entier n. Sur un ordinateur, B sera une puissance de 2 et conviendra pour les applications

cryptographiques où l'entier n est toujours un nombre impair, soit parce qu'il est un grand nombre premier, soit parce qu'il est le produit de deux grands nombres premiers. La base de numération B est par conséquent inversible modulo n et la multiplication par B modulo n réalise un codage des entiers modulo n. Ce qu'on appelle la *représentation de Montgomery* de l'entier x est l'entier $\widetilde{x}$ égal à $B \times x$ mod n. Réciproquement, l'entier x obtenu à partir de $\widetilde{x}$ s'appelle la *réduction de Montgomery* de $\widetilde{x}$. Réduire un nombre consiste à le diviser par B modulo n, ce qu'on note $\texttt{RED}(\widetilde{x}) = x = \widetilde{x}/B \mod n$. Il existe une méthode assez simple, présentée sur l'encadré 3.13 qui ne requiert aucune division, mais seulement deux multiplications pour calculer la réduction de Montgomery d'un entier donné.

La figure 3.10 indique les représentations de Montgomery $\widetilde{x}$ des entiers x modulo 7 en base 10. Cela définit une opération notée $*$ qui a les mêmes propriétés que la multiplication modulo n. Le produit de 3 par 5 vaut 15 qui est congru à 1 modulo 7. En transcrivant dans la représentation de Montgomery, l'opération devient $2 * 1 = 3$. Ce résultat est la réduction du produit 2×1, qui vaut 3 comme l'indique l'encadré 3.13.

Encadré 3.13. La réduction de Montgomery.

Pour réaliser la réduction de Montgomery modulo n, il faut une base de numération B qui ne partage aucun facteur premier avec n. On peut alors précalculer un entier n' qui est l'opposé de l'inverse de n modulo B. Par exemple, en base $B = 10$, pour réduire modulo $n = 7$, il faut précalculer $n' = -1/7$ modulo 10 qui vaut 7, car $7 \times 7 = 49$ est bien égal à -1 modulo 10.

Pour réduire un entier $\widetilde{x}$, on pose $q = n'\widetilde{x}$ modulo B. La quantité $\widetilde{x} + qn$ est congrue à 0 modulo B, ce qui signifie qu'elle est multiple de B. Le quotient par B donne le résultat x cherché. Toutefois, si $\widetilde{x}$ est inférieur à n^2, on a seulement $x < 2n$, ce qui peut dans certains cas nécessiter une soustraction supplémentaire pour obtenir un résultat dans l'intervalle $0, \ldots, n-1$.

Par exemple, toujours avec $B = 10$ et $n = 7$, posons $\widetilde{x} = 2$. On a $q = 7 \times 2$ vaut 4 modulo 10. La quantité $2 + 4 \times 7 = 30$ est bien multiple de 10 et le résultat cherché est $x = 3$. Le résultat a été obtenu en deux multiplications, l'une pour calculer l'entier q, et l'autre pour calculer la valeur de x. La division par la base n'est qu'un décalage des chiffres. Elle est immédiate.

6.3 Élever à la puissance *k*

Le RSA et le chiffrement ElGamal, comme tous les algorithmes cryptographiques qui exploitent l'arithmétique, utilisent l'exponentiation modulo n qui consiste à élever une donnée x à une certaine puissance modulo n. Il s'agit maintenant de trouver une méthode efficace pour calculer cette puissance en minimisant le

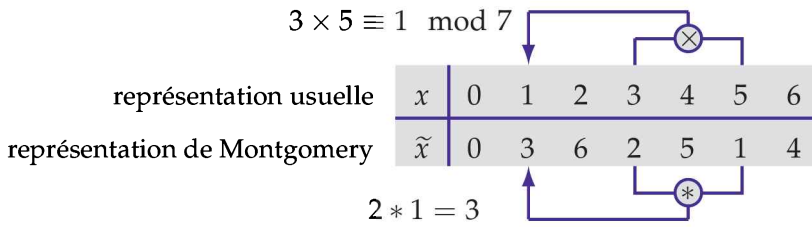


FIGURE 3.10. Représentation de Montgomery des entiers modulo 7 en base 10. La représentation d'un entier x est donnée par la formule $\widetilde{x} = 10 \times x \mod 7$. Pour réaliser l'opération inverse, c'est-à-dire retrouver x à partir de $\widetilde{x}$, il existe une méthode qui ne nécessite aucune division. Cette correspondance définit une opération $*$ similaire à la multiplication modulo 7.

nombre de multiplications. Élever à la puissance k peut se faire avec $k-1$ multiplications, mais cela est loin d'être satisfaisant. La figure 3.11 montre comment calculer x^{13} avec seulement deux élévations au carré et trois multiplications. La suite des exposants des puissances de x successivement calculées est :

$$1 \rightarrow 2 \rightarrow 3 \rightarrow 5 \rightarrow 10 \rightarrow 13$$

et constitue ce qu'on appelle *une chaîne d'additions de 1 à 13 de longueur 5*. L'entier 1 est toujours le point de départ, l'entier 13 est la cible de la chaîne, la valeur 5 est le nombre de flèches. Chaque terme est, soit la somme de deux termes précédents, soit le double d'un terme précédent.

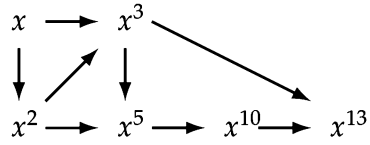


FIGURE 3.11. Élévation de x à la puissance 13 en deux élévations au carré et trois multiplications. On élève x au carré, pour obtenir x^2, qu'on multiplie par x pour obtenir x^3. La valeur x^5 s'obtient en multipliant x^2 par x^3, celle de x^{10} en élevant x^5 au carré, et finalement x^{13} est le résultat de la multiplication de x^3 par x^{10}.

Une chaîne d'addition de cible n décrit exactement un algorithme d'élévation à la puissance n à partir de multiplications et d'élévations au carré. Chaque flèche d'une chaîne d'addition correspond à une multiplication ou à une élévation au carré dans l'algorithme d'exponentiation qu'elle décrit. Trouver un algorithme d'exponentiation qui nécessite le moins possible de multiplications revient à trouver une chaîne d'addition qui cible l'exposant et qui soit aussi courte que possible. On ne connaît pas actuellement d'algorithme efficace pour résoudre ce problème en toute généralité, mais seulement des méthodes heuristiques. Citons deux d'entre elles.

Chaîne d'additions obtenue avec l'écriture binaire de la cible

Une façon systématique d'obtenir des chaînes d'additions assez courtes est d'utiliser l'écriture de la cible dans la numération en base 2. Partant du premier 1, on parcourt les chiffres de gauche à droite. Lorsqu'on rencontre un 0, on réalise un doublement, et lorsqu'on rencontre un 1, on réalise un doublement suivi de l'addition d'une unité. Prenons par exemple la cible $n = 13$ dont l'écriture binaire est 1101. On obtient la chaîne suivante :

$$\underbrace{1}_{1}\underbrace{\rightarrow 2 \rightarrow 3}_{1}\underbrace{\rightarrow 6}_{0}\underbrace{\rightarrow 12 \rightarrow 13}_{1}$$

Pour une cible dont l'écriture binaire comprend i chiffres 1 et j chiffres 0, la longueur de la chaîne est $2i + j - 2$. Elle est la plus longue lorsque l'écriture binaire ne comprend que des 1. Pour $n = 15$, dont l'écriture binaire est 1111, on obtient la chaîne de longueur 6 :

$$\underbrace{1}_{1}\underbrace{\rightarrow 2 \rightarrow 3}_{1}\underbrace{\rightarrow 6 \rightarrow 7}_{1}\underbrace{\rightarrow 14 \rightarrow 15}_{1}$$

Chaîne d'additions obtenue avec la factorisation de la cible

On peut tirer profit de la factorisation de la cible pour composer les chaînes d'additions pour chacun des facteurs. Par exemple, pour $n = 15$, on peut composer une chaîne d'additions de cible 3, par exemple $1 \rightarrow 2 \rightarrow 3$ avec une chaîne d'addition de cible 5, par exemple $1 \rightarrow 2 \rightarrow 4 \rightarrow 5$. Pour cela, on multiplie par 3 chacun des termes de la chaîne de cible 5 et on la concatène à la chaîne de cible 3 :

$$1 \rightarrow 2 \rightarrow \underbrace{3 \rightarrow 6 \rightarrow 12 \rightarrow 15}_{\text{triple de la chaîne pour 5}}$$

Cette chaîne d'addition, de longueur cinq, est plus courte que celle obtenue avec la décomposition binaire. Elle permet de réaliser l'élévation à la puissance 15 avec seulement cinq multiplications.

6.4 Produire des nombres premiers

Produire à grande échelle des nombres premiers de grande taille devient une nécessité pour alimenter la plupart des mécanismes à clé publique. La méthode la plus employée, qui permet d'obtenir des nombres premiers quelconques sans propriété particulière, est de partir d'un nombre aléatoire, et de chercher le prochain nombre premier qui le suit. Pour cela, il faut un test qui, à partir d'un

entier, sait décider si cet entier est ou n'est pas un nombre premier. Le test qui consiste à diviser par tous les nombres qui lui sont inférieurs n'est bien sûr pas satisfaisant en raison de sa très faible efficacité pour les grand nombres. On peut utiliser cette méthode pour vérifier que l'entier testé n'est pas divisible par de petits nombres premiers comme 1, 3, 5, 7, jusqu'à une certaine borne, pour éliminer rapidement les cas évidents, mais il convient de compléter cette méthode par un test qui donnera une réponse plus complète.

Les tests probabilistes

En 2002, les mathématiciens indiens Manindra Agrawal, Neeraj Kayal et Nitin Saxena ont découvert un test efficace de primalité qui permet de répondre avec certitude. En leur honneur, ce test porte le nom de *test AKS*. Il reste cependant encore d'une complexité trop importante pour être pratiquable. On utilise toujours aujourd'hui des tests de primalité plus efficaces, mais qui ne donnent qu'une réponse incertaine. Il subsiste une très faible probabilité que le test déclare un nombre premier alors qu'il ne l'est pas. Le degré d'incertitude résiduelle est si faible que tout le monde s'en satisfait. Ces tests reposent sur une propriété satisfaite par les nombres premiers qu'on peut énoncer ainsi :

si l'entier n est premier, alors la propriété $P(n)$ est vérifiée.

Bien sûr, si la propriété $P(n)$ n'est pas satisfaite, on sait avec certitude que le nombre n n'est pas premier – on dit dans ce cas que n est *composé* – mais il subsiste des cas où la propriété $P(n)$ est vérifiée pour des nombres composés. On parle de test probabiliste dont la réponse la plus adéquate est « *ce nombre est probablement premier* ». Les mathématiciens continuent à chercher des propriétés qui réduisent l'incertitude de la réponse. Il existe même certains tests pour lesquels on ne sait ni prouver que leur réponse est certaine, ni trouver de contre-exemple qui infirmerait leur caractère déterministe.

Un premier test repose sur le petit théorème de Fermat qui énonce que, pour tout entier a premier avec n, on a :

si n est un nombre premier, alors $a^{n-1} = 1$ modulo n.

Cette propriété se décline immédiatement en un test de primalité. Soit n un entier à propos duquel on souhaite savoir s'il est premier ou composé. Choisissons un entier a entre 1 et $n - 1$ dont on aura préalablement vérifié avec l'algorithme d'Euclide qu'il est premier avec l'entier n. Si tel n'est pas le cas, nous pouvons déjà affirmer que l'entier n est composé. Calculons maintenant a^{n-1} modulo n. Si le résultat vaut 1, alors nous pourrons dire que l'entier n est probablement

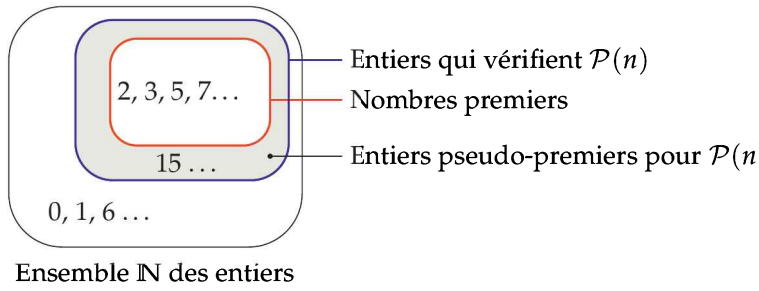


FIGURE 3.12. Illustration des tests probabilistes de primalité. On utilise une propriété $\mathcal{P}(n)$, satisfaite pour tous les nombres premiers. Elle sera d'autant plus efficace que l'ensemble marqué en gris des entiers composés qui la vérifient sera réduit. Pour certaines propriétés, on ne sait ni trouver de contre-exemple, ni prouver qu'il n'en existe pas.

premier, mais si le résultat prend une autre valeur, alors nous aurons la certitude que l'entier n est composé. Au besoin, tant que le résultat vaut 1, on répète la procédure avec plusieurs valeurs de la base a jusqu'à être suffisamment convaincu de la primalité de l'entier n.

Pour illustrer la méthode, essayons de prouver la primalité de $n = 15$. Si l'on choisit une base $a = 4$, on calcule 4^{14} qui est congru à 1 modulo 15. L'entier 15 est probablement premier ! Mais ce résultat est entaché d'une certaine incertitude. N'étant pas convaincu, essayons avec une autre base, par exemple $a = 2$. Cette fois 2^{14} est congru à 4 modulo 15, ce qui nous assure que l'entier 15 n'est pas premier. Cependant, comme il passe le test avec la base $a = 4$, nous dirons que l'entier 15 est *pseudo-premier en base 4*.

L'existence de nombres pseudo-premiers, qui passent le test et qui pourtant ne sont pas premiers, est le grave inconvénient de ces tests probabilistes. Pour le test de Fermat, il existe même des entiers, appelés *nombres de Carmichael* qui sont composés, mais qui pourtant passent le test pour toute base qui ne partage aucun diviseur commun avec eux. Le plus petit de ces nombres est 561 qui vaut $3 \times 11 \times 17$, et pour qui a^{560} est congru à 1 modulo 561 pour tout entier a qui n'a ni 3, ni 11, ni 17 comme diviseur.

Une amélioration du test de Fermat est le test de Miller-Rabin. Prenons par exemple la base $a = 2$ et le nombre de Carmichael $n = 561$. Examinons l'élévation de a à la puissance 560. L'exposant est un nombre pair. La procédure d'élévation à la puissance 560 se terminera par une succession d'élévations au carré :

$$2 \to \cdots \to 2^{35} = 263 \underset{x^2}{\to} 2^{70} = 166 \underset{x^2}{\to} 2^{140} = 67 \underset{x^2}{\to} 2^{280} = 1 \underset{x^2}{\to} 2^{560} = 1$$

Lors de ce calcul, il apparaît que le carré de 67 est congru à 1 modulo 561. Or si 561 était un nombre premier, l'entier 1 n'aurait que deux racines carrées modulo 561 : 1 et -1, c'est-à-dire 1 et 560. On observe ici que 67 en est une aussi. Non seulement nous pouvons affirmer que l'entier 561 n'est pas premier, mais cela permet même de le factoriser comme indiqué au paragraphe 2.4 page 51. En effet, le calcul $\text{pgcd}(561,67-1) = 11$ permet d'exhiber un facteur.

Le test de Miller-Rabin en base a consiste à affirmer qu'un entier n est probablement premier si, d'une part, la valeur de a^{n-1} est congrue à 1 modulo n et si, en plus, les élévations au carré successives à la fin de la procédure d'élévation à la puissance $n-1$ ne font jamais apparaître de racine carrée de 1 autre que 1 et -1. Un nombre sera dit *pseudo-premier fort en base a* s'il est composé et s'il passe quand même le test de Miller-Rabin en base a. Malheureusement, de tels nombres existent, comme par exemple $n = 121$ qui, bien qu'il soit composé, puisque $121 = 11 \times 11$, passe le test de Miller-Rabin en base 3 :

$$3 \to \cdots \to 3^{15} = 1 \underset{x^2}{\to} 3^{30} = 1 \underset{x^2}{\to} 3^{60} = 1 \underset{x^2}{\to} 3^{120} = 1$$

Bien que 121 soit composé, le détail de l'élévation de 3 à la puissance 120 modulo 121 ne fait pas apparaître de racine carrée de 1 autre que 1 ou -1.

Un test pratiquement déterministe

Présentons pour terminer un dernier test particulièrement performant reposant sur la fameuse suite de Fibonacci. Le premier terme de cette suite vaut 0, le deuxième vaut 1 et les autres sont égaux à la somme des deux précédents :

$$0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377\ldots$$

Il est possible de définir la suite de Fibonacci modulo un entier n simplement en effectuant l'addition modulo n. Par exemple la suite de Fibonacci modulo 13 est

$$0, 1, 1, 2, 3, 5, 8, 0, 8, 8, 3, 11, 1, 12, 0\ldots$$

Une propriété remarquable de la suite de Fibonacci modulo n est la suivante : si n est un nombre premier qui est congru à 2 ou à -2 modulo 5, ce qui est le cas pour $n = 13$, alors, en numérotant les termes à partir de zéro, le terme numéro $n+1$ est toujours congru à 0 modulo n. Par exemple, dans la suite ci-dessus, on observe que le terme numéro quatorze est bien congru à 0 modulo 13.

Cette propriété conduit à un test de primalité pour les entiers n qui sont congrus à ± 2 modulo 5. On calcule le terme numéro $n+1$ modulo n comme indiqué dans l'encadré 3.14. Si sa valeur est 0, alors l'entier n est probablement

premier et, sinon, il est composé. Il existe des nombres composés qui passent ce test et qui sont appelés tout naturellement des nombres *pseudo-premiers de Fibonacci*. Le premier d'entre eux est $n = 323$.

Encadré 3.14. Calcul direct des termes de la suite de Fibonacci.

Calculer le k^e terme de la suite de Fibonacci en calculant tous les termes à partir des deux premiers est totalement inefficace. Il existe heureusement une méthode directe reposant sur une algèbre un peu particulière. Soit n un entier qui est congru à ± 2 modulo 5. Cette condition signifie qu'il n'existe pas d'entier modulo n dont le carré vaut 5. Notons α une quantité imaginaire qui vérifie $\alpha^2 = 5$, analogue à l'imaginaire i des nombres complexes dont le carré vaut -1. Considérons les suites dont chaque terme à partir du troisième est la somme des deux précédents. Une telle suite est entièrement définie par ses deux premiers termes. Parmi ces suites, il en existe deux qui ont des propriétés particulièrement intéressantes. Il s'agit de la suite dont les deux premiers termes sont 1 et $\varphi = (1 + \alpha)/2$, et de celle dont les deux premiers termes sont 1 et $\overline{\varphi} = (1 - \alpha)/2$. Ces deux suites ont en effet la particularité d'être des suites géométriques. Chaque terme s'exprime en fonction du précédent en le multipliant par φ pour la première suite et par $\overline{\varphi}$ pour la deuxième

$$1, \quad \varphi, \quad \varphi^2, \quad \varphi^3, \quad \ldots, \quad \varphi^n, \quad \ldots$$
$$1, \quad \overline{\varphi}, \quad \overline{\varphi}^2, \quad \overline{\varphi}^3, \quad \ldots, \quad \overline{\varphi}^n, \quad \ldots$$

Le terme numéro n de la suite de Fibonacci s'exprime alors simplement par :

$$F_n = \frac{\varphi^n + \overline{\varphi}^n}{\alpha}$$

Calculer directement le terme F_n se réalise alors avec un algorithme d'élévation à la puissance n comme ceux décrits au paragraphe 6.3 page 73, qu'on appliquera à φ et à $\overline{\varphi}$.

Mais le plus intéressant est que la combinaison de ce test et du test de Fermat en base 2 est pratiquement déterministe. On ne sait, ni prouver que la combinaison de ces deux tests conduit à une réponse certaine sur la primalité de l'entier testé, ni exhiber de contre-exemple. Le mathématicien américain Carl Pomerance a déclaré en 1984 offrir une prime de 20 dollars à qui en fournirait un avec sa factorisation. Cette prime a été portée à 620 dollars en 1994. Selon le Canadien Jeff Gilchrist, une recherche systématique toujours en cours en 2012 n'en a toujours pas trouvé jusqu'à 10^{19}. La prime proposée par Carl Pomerance tient aussi pour qui apportera la preuve de la non-existence de tels nombres. Avis aux amateurs !

4

La cryptanalyse

La cryptanalyse est le chiffre d'attaque. L'objectif du cryptanalyste, qui ne voit que le cryptogramme, est de retrouver des informations sur le message en clair alors qu'il ne dispose pas de la clé de déchiffrement. Son travail est le complément indispensable de celui du concepteur. Ce n'est que lorsqu'un procédé résiste à toute tentative de cryptanalyse qu'il est considéré comme sûr. Ce chapitre passe en revue les principales méthodes d'attaque, y compris les attaques physiques sur le dispositif de calcul, qui font aujourd'hui l'objet de nombreuses recherches.

1 La force brutale

Dans le domaine de la cryptologie, la réalité dépasse parfois de très loin la fiction. Dans un roman publié en 2007, *Forteresse digitale*, Dan Brown décrit l'inquiétude de l'agence américaine de sécurité NSA face à la création d'un code indécryptable. Voici un extrait de ce roman :

> *Dans un premier temps, les clés secrètes des utilisateurs étaient suffisamment courtes pour être « devinées » par les ordinateurs de la NSA. Pour décrypter une clé secrète à dix chiffres, il suffisait de programmer la machine pour qu'elle essaie toutes les combinaisons possibles entre 0000000000 et 9999999999. Tôt ou tard, l'ordinateur tombait sur la bonne séquence, cette façon de procéder par élimination était surnommée « l'attaque de force brute ». Cela prenait parfois beaucoup de temps, mais le résultat était garanti.*
>
> *(...) Les mots de passe devinrent de plus en plus lourds. Le temps nécessaire pour deviner les combinaisons se chiffra en semaines, puis en mois, puis en années.*

> *Dans les années quatre-vingt-dix, les clés dépassaient les cinquante caractères, (...) le nombre de possibilités avoisinait les* 10^{120} *– un suivi de cent vingt zéros. La probabilité mathématique de tomber sur le code exact revenait à trouver le bon grain de sable sur une plage de plus de quatre kilomètres de long.*

Le travail pour retrouver un grain de sable dans une si longue plage apparaît sans doute considérable à l'auteur qui a voulu impressionner le lecteur avec cette image. La réalité est toute autre. Le nombre annoncé par l'auteur, 10^{120}, dépasse de loin tout ce qui est imaginable. Dès le III^{e} siècle avant notre ère, Archimède, dans un petit ouvrage appelé *l'arénaire*, a lui aussi utilisé la métaphore des grains de sable pour appréhender les grandeurs considérables. Il a évalué combien l'Univers tout entier pouvait contenir de grains de sable, à supposer qu'il en soit entièrement rempli. Il a utilisé pour cela le modèle héliocentrique d'Aristarque de Samos, qui « *suppose que la grandeur de la sphère* (des étoiles fixes) *dans laquelle il veut que la terre se meuve est égale à la sphère que nous appelons le monde.* ». Il a dû imaginer un système de numération capable de traiter des grandeurs gigantesques. Après une longue série de calculs, il conclut, par des arguments géométriques irréfutables, que « *le nombre des grains de sable contenus dans une sphère aussi grande que la sphère des étoiles fixes, supposée par Aristarque, est plus petit que mille myriades des nombres huitièmes* ». Cela signifie, en numération scientifique moderne, que si l'Univers tout entier était supposé entièrement remplis de grains de sable, le nombre de ceux-ci ne saurait excéder 10^{63}.

Encadré 4.1. Le système de numération d'Archimède.

La numération grecque dispose de la *myriade*, qui désigne dix mille. Elle peut aussi *répéter une myriade jusqu'à dix mille myriades*. Cela reste insuffisant pour représenter le nombre de grains de sable dont on peut remplir l'Univers. Pour cela, Archimède met au point un nouveau système qu'il décrit ainsi :

> *Que les nombres (...) qui vont jusqu'à une myriade de myriade soient appelés nombres premiers, et qu'une myriade de myriades de nombres premiers soient appelés l'unité des nombres seconds. Comptons par ces unités, et par les dizaines, les centaines, les milles, les myriades de ces mêmes unités, jusqu'à une myriade de myriades. Qu'une myriade de myriades des nombres seconds soit appelée l'unité des nombres troisièmes (...)*

En notation scientifique actuelle, les nombres premiers d'Archimède s'étendent de 1 à 10^{8}, les nombres seconds de 10^{8} à 10^{16}, les nombres troisièmes de 10^{16} à 10^{24}, et ainsi de suite par ce qu'il appelle des octades. Les nombres huitièmes s'étendent de 10^{56} à 10^{64}.

La cryptologie manipule des nombres si grands que notre intuition fait défaut pour se les représenter. Voici quelques éléments pour se familiariser avec eux :

- $2^{80} = 10^{24}$: nombre de dés à coudre d'eau dans tous les océans terrestres ;
- 10^{35} : nombre de nanosecondes depuis la création de l'Univers, dont l'âge est aujourd'hui estimé à plus de treize milliards d'années ;
- 10^{80} : nombre d'atomes contenus dans cent milliards de galaxies, contenant chacune cent milliards d'étoiles grosses en moyenne comme cent fois notre soleil.

À la lumière de ces ordres de grandeur, le lecteur est invité à relire l'extrait du roman de Dan Brown présenté au début de ce chapitre.

2 La loi de Moore

Les progrès réalisés dans les performances des moyens de calcul permettent d'envisager une attaque par force brutale sur des tailles de données sans cesse croissantes. Depuis l'apparition du calcul automatique, ces progrès sont considérables. Il a fallu 70 heures à l'ordinateur ENIAC en 1949 pour calculer deux mille décimales du nombre π, alors qu'aujourd'hui, le moindre calculateur embarqué dans un téléphone portable peut mener le même calcul en une fraction de seconde. En 1977, la revue *Scientific American* publiait le principe du système RSA sous le titre *Un nouveau moyen de chiffrement qu'on mettrait des millions d'années à casser*. Pourtant, l'entier de 129 chiffres sur lequel il reposait a été factorisé en 1994, c'est-à-dire bien avant les millions d'années annoncés. Pourquoi une telle avance ? La raison essentielle en est l'étonnant progrès de la puissance de calcul des ordinateurs. Ce progrès suit une loi empirique, connue sous le nom de *loi de Moore*, énoncée pour la première fois dans le numéro du 19 avril 1965 de la revue *Electronics* par Gordon E. Moore, alors directeur de recherche et développement à l'entreprise américaine Fairchild qui a commercialisé les premiers circuits intégrés.

> *La complexité à coût minimal des composants a été environ multipliée chaque année par un facteur deux. Sans doute, à court terme, ce taux de croissance se maintiendra, si toutefois il n'augmente pas. À long terme, le taux de croissance est un peu plus incertain, mais il n'y a aucune raison de croire qu'il ne se maintiendra pas pendant au moins dix ans. Cela signifie qu'en 1975, le nombre de composants par circuit intégré au moindre coût sera de* 65 000.

Pour illustrer la véracité de cette prédiction, rappelons que le processeur Motorola 68 000, qui doit son nom au nombre de transistors qu'il contient est apparu en 1979.

La finesse de gravure d'un circuit intégré est la largeur la plus fine d'une bande de silicium qu'il est possible d'y graver. Plus la gravure est fine, et plus il est possible de concentrer un grand nombre de transistors, et donc de fonctions de calcul sur une surface de silicium donnée. La finesse de gravure des premiers circuits était de l'ordre de la dizaine de microns ($1\mu = 10^{-6}$ m). Elle se mesure aujourd'hui en nanomètres ($1\text{ nm} = 10^{-12}$ m). Le tableau suivant montre l'évolution de ce paramètre qui mesure les progrès dans l'intégration des circuits.

1971	1980	1989	1998	2007	2010	2014
10 μ	3 μ	800 nm	250 nm	90 nm	32 nm	(15 nm)

Jusqu'en 2007, l'évolution de la finesse de gravure s'accompagnait également d'une augmentation de la fréquence de fonctionnement, jusqu'à 4 Ghz environ, ce qui a un impact direct sur la vitesse des calculs. Depuis 2007, la fréquence n'augmente quasiment plus, car la consommation électrique croît avec la fréquence de fonctionnement du circuit et la dissipation de chaleur qui résulterait d'une fréquence plus élevée pourrait faire fondre le circuit. Les gains sur la finesse de gravure permettent toutefois une intégration plus grande, ouvrant la voie à des circuits qui comprennent deux à huit processeurs, pouvant mener plusieurs calculs simultanément.

Encadré 4.2. À qui profite la loi de Moore ?

Dans la lutte sans merci qui oppose le cryptographe au cryptanalyste, on croit souvent que c'est ce dernier qui tire le plus grand profit des progrès en moyens de calcul. Ceux-ci rendent envisageable l'utilisation de la force brutale. Par exemple, la cryptanalyse des machines de Lorenz pendant le deuxième conflit mondial n'a été possible que grâce à la construction d'une machine gigantesque, le *Colossus*, exploitant la vitesse offerte par la nouvelle technologie électronique des tubes à vide, à l'époque utilisés depuis peu pour accélérer les commutateurs téléphoniques.

Pourtant, un examen plus attentif montre que c'est au contraire au cryptographe, au chiffre de défense, que bénéficient ces progrès. Pour fixer les idées, supposons qu'à un moment donné, on utilise des clés RSA de 200 chiffres. Si, grâce aux progrès technologiques, la puissance des moyens de calcul est doublée, la taille des clés pourra être augmentée de 25 % sans que l'utilisateur n'y voie le moindre changement. Elles pourront comporter 250 chiffres. Mais, selon la formule de l'encadré 4.4 page 102, le travail du cryptanalyste pour les factoriser devra alors être multiplié par 36. Avec ses moyens qui n'auront que doublé, il aura perdu un facteur 18 dans cette opération. Et ce sera pire la prochaine fois ! Avec une puissance encore doublée, les clés pourront être portées à 312 chiffres, mais cette fois-ci, le travail du cryptanalyste devra encore être multiplié par 58.

C'est le chiffre de défense qui bénéficie des progrès du calcul !

2 La résolution des substitutions simples

Les progrès de la cryptanalyse ne sont pas dus qu'à la force. L'ingéniosité des hommes y a aussi largement contribué. L'écrivain américain Edgar Allan Poe, qui était féru de cryptographie, a écrit une nouvelle parue en 1843, *Le Scarabée d'or*, où une chasse au trésor repose sur le cryptogramme que voici :

53‡‡†305))6*;4826)4.)4‡);806*;48†8¶60))85;1‡(;:‡
8†83(88)5†;46(;88*96*?;8)*‡(;485);5*†2:*‡(;4956*2(5*-4)
8¶8*;4069285);)6†8)4‡‡;1(‡9;48081;8:8‡1;48†85;4)485†
528806*81(‡9;48;(88;4(‡?34;48)4‡161;:188:‡?;

Une partie de la nouvelle est consacrée à la description de la façon de résoudre ce cryptogramme. Cela a été conduit selon les méthodes traditionnelles de la cryptanalyse, établies pour la première fois dans le monde arabe aux VIII^e^ et IX^e^ siècles. Deux types de méthodes sont au cœur du décryptement :

- les *méthodes quantitatives* consistent à compter la fréquence de chaque lettre dans le texte ainsi que les groupements de deux ou trois lettres, pour les comparer aux fréquences des lettres dans la langue du message ;
- les *méthodes qualitatives* sont faites de réflexion et d'intuition. Il s'agit d'étudier les mots probables et les combinaisons possibles et impossibles de lettres. Les groupements *es*, *en*, *nt*, *ion* sont fréquents en français. Les groupements *kn*, *gh* ou *tw* y sont très rares alors qu'ils sont courants en anglais. Dans ces deux langues, un *u* suit presque toujours un *q*.

La première étape de la cryptanalyse du scarabée d'or est l'analyse des fréquences par comptage des caractères. Le « 8 » est le plus fréquent et apparaît 33 fois, ensuite le « ; » apparaît 26 fois, le « 4 » apparaît 19 fois, le « ‡ » et le «) » 16 fois chacun, etc. La nouvelle décrit le travail acharné du crypanalyste :

> *Donc 8 représentera e. Maintenant, de tous les mots de la langue, « the » est le plus utilisé ; conséquemment, il nous faut voir si nous ne trouverons pas répétée plusieurs fois la même combinaison de trois caractères, ce 8 étant le dernier des trois. Si nous trouvons des répétitions de ce genre, elle représenteront très probablement le mot « the ».*

Une recherche intuitive de type linguistique reposant sur l'étude des mots probables, aidée par la connaissance des caractéristiques de la langue et recoupée avec les informations déjà acquises, conduit au décryptement complet (solution page 174).

Une méthode infaillible pour trouver les voyelles. Les cryptanalystes ont toujours cherché à développer les méthodes quantitatives, afin de faciliter au

maximum la tâche qualitative, qui est de loin la plus délicate et que Vigenère qualifiait de *rompement de cerveau*. François Viète, décrypteur du roi Henri IV, devait, en pleine guerre de religions, décrypter un nombre de plus en plus important de dépêches chiffrées en provenance du roi Philippe II d'Espagne. Il a établi une méthode analytique pour commencer ce travail, méthode qu'il a qualifiée d'infaillible.

Règle infaillible Parmi trois lettres consécutives, on trouve toujours une ou plusieurs des cinq voyelles *A*, *E*, *I*, *O* ou *U*.

Cette propriété est presque toujours satisfaite en espagnol. Même si elle l'est moins en français, sa vraisemblance suffit pour commencer la résolution. Le premier travail de Viète face à un cryptogramme sera de rechercher les voyelles par un travail systématique qui pouvait être confié à un exécutant.

Il a finalement été très tôt établi que les substitutions simples tombent toujours sous les coups de l'analyse des fréquences, plus ou moins rapidement selon l'habileté du cryptanalyste.

> *Je vous en ai dit assez pour vous convaincre que des chiffres de cette nature sont faciles à résoudre...*
>
> Edgar Allan Poe, *Le Scarabée d'or*.

4 La cryptanalyse du chiffrement polyalphabétique

Le chiffre de Vigenère, avec une clé cyclique, est resté indécrypté pendant plusieurs siècles. Il a fallu attendre le milieu du XIX^{e} siècle pour commencer à le voir vaciller. Citons trois acteurs de ce décryptement.

4.1 Charles Babbage

Charles Babbage (1791-1871) est honoré par les informaticiens comme l'auteur des plans de la machine analytique, première machine mécanique de calcul numérique conçue pour être programmable, et qui a d'étonnantes similarités avec l'architecture Von Neumann des ordinateurs actuels. Il a eu une activité constante et reconnue en cryptologie de 1831 jusqu'en 1870, en particulier en cryptanalyse. Il dira dans une autobiographie en 1864 :

> *Le décryptement est l'un des arts les plus fascinants, et je crains d'y avoir dépensé plus de temps qu'il ne le mérite.*

En 1846, il résout un défi, lancé par son neveu Henry Holliers, qui consiste à décrypter le message montré sur la figure 4.1.

La méthode qui apparaît sur les notes de Babbage est faite d'analyse de mots probables et de recherche intuitive dans un travail où la signification du message

```
PYRI ULOFV
   POVVMGN MK UO GOWR HW LQ PGFJHYQ
OJAV MSN WIJHEEHPR BRVGRUHEGK, EFF WJSR RVY
CPOY VSP, PX OKLN PI XXYSNLA SELF XG
FEEWTALV LJIU, WR MOI EGAP HMFL ML YINZ
TNGDDG YQIV UYEAP-BQL
WJQV PGYK STRITLMHFOFI
    EWTAWK
         TIEJC
```

FIGURE 4.1. Message que Henry Hollier met au défi son oncle Charles Babbage de décrypter. Ce dernier en viendra à bout le 19 mars 1854, après plus d'un mois de travail acharné.

occupe une grande place. Il procède par exemple à de nombreux essais sur le mot composé UYEAP-BQL. Après avoir essayé sans succès *water-dog*, *fools-up*, *birth-day*, *wool-cap*, *money-bag*, *night-cap*, *small-pox*, *heavy-wet*, *death-bed*, *house-dog*, *under-cut*, il trouve *brain-box* qui est le bon résultat. Puis, quittant le terrain des devinettes, il se tournera sur les trigrammes les plus fréquents : *tea*, *the*, *and*, etc. La mention « *cypher given me by the Duke of Somerset* » sur le manuscrit lui fait sans doute trouver SOMERSET qui est l'un des mots clés du message. Cette clé convient pour la fin du message, mais s'avérera mauvaise pour la première partie. Il trouve finalement les mots clés MURRAY et CACOETHES (solution page 174).

En 1854, Babbage intervient également comme expert pour évaluer une invention qu'un certain Mr. Twaite présente dans le *Journal of the Society of Art* comme utile pour maintenir le caractère privé des échanges télégraphiques. Il s'agit en fait d'une ré-invention du chiffre de Vigenère. Babbage montre que le procédé est déjà connu, et précise aussi qu'il est facilement décryptable. Il s'en suit un échange dans le journal où Mr. Twaite lance un défi qui est résolu par Babbage, sans toutefois que ce dernier ne précise comment il a procédé. Il est vraisemblable que Babbage a décrypté ce message d'une manière plus méthodique que celle, laborieuse, qu'il a utilisée contre le message de son neveu.

4.2 Le test de Kasiski

Friedrich Wilhelm Kasiski (1805-1881) est un officier d'infanterie prussien. Il a publié la première méthode analytique de décryptement du chiffre de Vigenère dans un petit ouvrage de 95 pages : *L'écriture secrète et l'art du déchiffrement*. Il n'était pas cryptologue professionnel. Sa découverte n'a pas été reconnue au moment de sa publication. Kasiski s'est ensuite intéressé à l'archéologie, sans se rendre compte du caractère essentiel de son travail pour la cryptanalyse.

La première tâche à accomplir pour décrypter un cryptogramme issu d'un chiffrement polyalphabétique avec clé cyclique est de trouver la longueur du cycle de la clé. La méthode décrite par Kasiski pour la trouver repose sur l'analyse des écarts entre les polygrammes répétés dans le cryptogramme. Elle est fondée sur les assertions suivantes, telles qu'elles sont énoncées en 1939 dans l'ouvrage *Élements de cryptographie* du Capitaine Baudouin :

- lorsque deux groupes de lettres sont répétés dans le message clair et s'ils sont placés de manière identique par rapport à la clé, ils seront pareillement chiffrés et, par conséquent, donneront des groupes de lettres égaux dans le cryptogramme ;
- réciproquement, dans la plupart des cas, deux groupes de lettres égaux du cryptogramme proviendront assez probablement de deux groupes de lettres égaux du message clair placés de manière identique par rapport à la clé ;
- en conséquence, l'intervalle qui sépare les groupes de lettres égaux dans le cryptogramme sera dans la plupart des cas un multiple de la taille de la clé.

Il faut donc repérer les répétitions de motifs identiques dans le cryptogramme et noter leur distance. Il se peut bien sûr que certaines répétitions soient fortuites et n'aient aucun lien avec la longueur de la clé. Toutefois, la longueur de clé la plus probable sera donnée par le diviseur le plus fréquemment observé parmi les distances des répétitions. Prenons par exemple le cryptogramme suivant :

```
twyne dhume dehgr qeeem icmgw
emldx mtsce cmeke hhdme fnnlj
xxvsa xzlqq alova tblyx iapfs
jhrcj soojv aosiu uuaxf suyrq
qotju zabpu lfzjx fs
```

Cherchons les groupes de lettres qui sont répétés dans ce cryptogramme. Le tableau 4.1 les énumère et indique leurs positions, les écarts des positions, ainsi que les diviseurs des écarts. Les premiers bigrammes répétés sont coloriés.

Le diviseur 7 est le plus fréquent. Il est donc raisonnable de faire l'hypothèse que le cryptogramme a été chiffré avec une clé cyclique de longueur 7. Ce test sur les diviseurs des écarts des répétitions pour déterminer la taille de la clé s'appelle aujourd'hui le *test de Kasiski*.

Connaissant maintenant la longueur de la clé, disposons le cryptogramme verticalement sur sept lignes. Sur chaque ligne, toutes les lettres se trouveront avoir été chiffrées avec le même décalage de type Jules César. Il reste à savoir quel décalage a été opéré sur chaque ligne. Cela peut se faire par recherche exhaustive, en regardant le décalage qui contient le plus d'*e, s, a, n, t, i, r, u, l, o*. Le

lettres répétés	positions et écarts	diviseurs des écarts
ed	$9 - 4 = 5$	5
eh	$39 - 11 = 28$	2, 4, 7, 14
rq	$98 - 14 = 84$	2, 3, 4, 6, 7, 12, 14, 21, 28, 42
me	$36 - 8 = 28$	2, 4, 7, 14
em	$25 - 18 = 7$	7
cm	$35 - 21 = 14$	2, 7
me	$43 - 36 = 7$	7
jx	$113 - 49 = 64$	2, 4, 8, 16, 32
ax	$92 - 54 = 38$	2, 19
qq	$99 - 58 = 41$	41
va	$84 - 63 = 21$	3, 7
fs	$94 - 73 = 21$	3, 7
xfs	$114 - 93 = 21$	3, 7

Tableau 4.1. Table donnant les digrammes et trigrammes répétés dans le cryptogramme ainsi que leurs positions d'apparition. La dernière colonne indique les diviseurs des écarts entre ces positions. Le diviseur le plus fréquent est le 7. Il est un candidat à la longueur du cycle de la clé répétée.

terme « *esantirulo* » est un mot facile à retenir, sorte de mot magique des apprentis cryptanalystes, constitué des lettres les plus fréquentes de la langue française. La lettre *e* a une fréquence d'environ 18 %, les lettres *s*, *a*, *n*, *t*, *i*, *r*, *u*, *l* et *o* ont chacune une fréquence de 5 à 8 %. Ces dix lettres couvrent à elles seules près de 80 % des lettres d'un texte écrit en français. On donne ci-dessous, pour chaque ligne, la lettre clé la plus probable.

```
turcdcdjzvirvurzz    r
wmqmxmmxlaacaaqaj    i
yeegmeexqtpjoxqbx    m
ndewtkfvqbfssfopf    b
eeeesensalsoistus    a
dhmmchnalyjouujl     u
hgilehlxoxhjuyuf     d
```

La figure 4.2 illustre comment la lettre clé *r* a été trouvée pour la première ligne. Le lecteur est invité à poursuivre le décryptement à son terme (solution page 175).

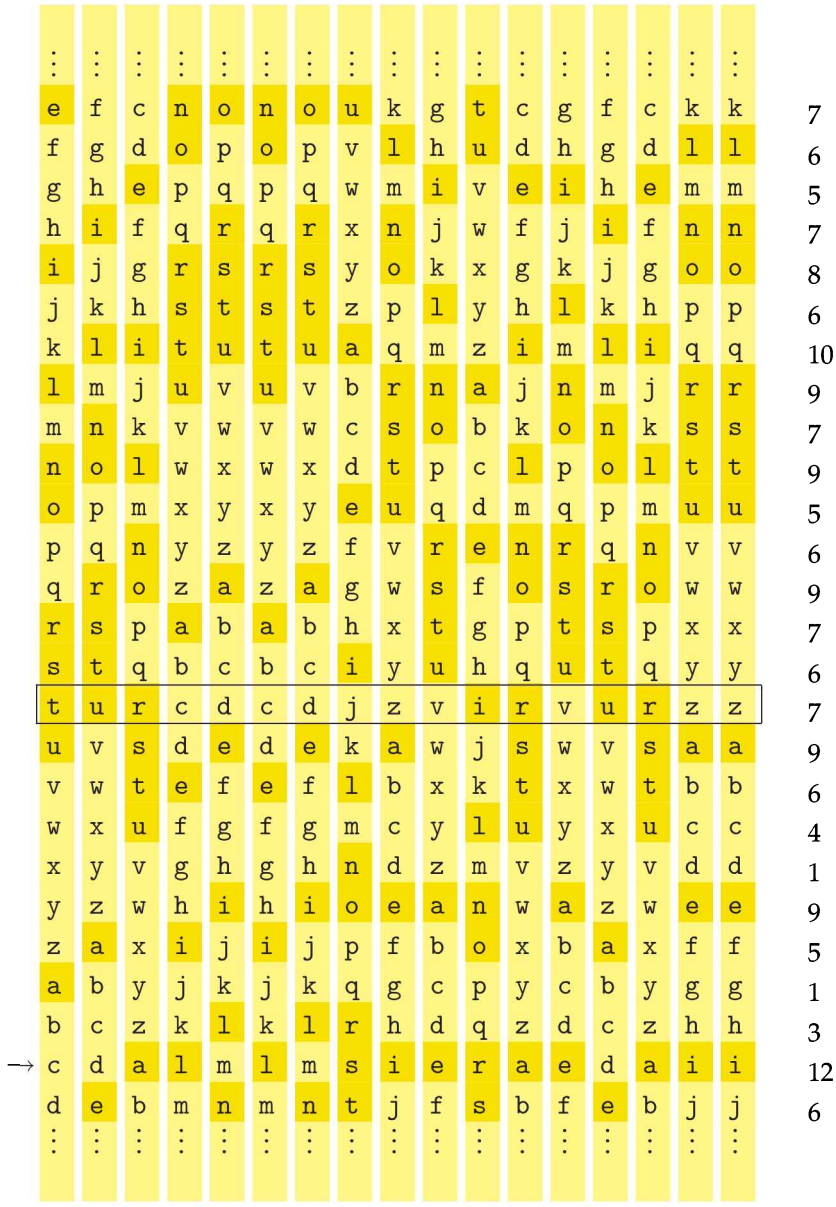


FIGURE 4.2. Pour trouver le décalage le plus probable, le cryptanalyste peut disposer de bandes de papier où est inscrit l'alphabet dans l'ordre avec les *esantirulo* surlignés. Il aligne les bandes de manière à faire apparaître les lettres du cryptogramme à résoudre, ici encadré, et il compte les *esantirulo* de chaque ligne. Celle qui en contiendra le plus sera la plus probable. En cas de doute, il peut affiner son analyse en éliminant les lignes où figurent trop de lettres rares. Ici, c'est l'avant-dernière ligne, qui contient douze *esantirulo* qui est nettement la plus probable. La septième ligne, bien qu'ayant dix *esantirulo*, est moins probable en raison des autres lettres qu'elle contient, k, q et z, qui sont bien plus rares. La première lettre du texte en clair, le c, a été changée en t dans le cryptogramme, soit un décalage de 17 rangs, ce qui correspond à la lettre clé r.

4.3 L'indice de coïncidence de William Friedman

William Friedman (1891-1969) a fait des études scientifiques de génétique à l'université Cornell, l'une des universités de la *Ivy League* dans le Nord-Est des États-Unis d'Amérique, puis a été embauché à l'institut privé de recherche Riverbank en 1915. Cette institution s'intéresse aussi à la cryptologie, en particulier pour essayer de prouver que les pièces de Shakespeare ont été écrites par Francis Bacon, théorie née de la supposition que les éditions originales des œuvres de Shakespeare contiendraient cette information, camouflée derrière la typographie des caractères. George Fabyan, fondateur et directeur de l'institut de recherche Riverbank pense que la validation de cette théorie le couvrira de gloire. Friedman a réalisé quelques études de génétique à Riverbank, mais a surtout aidé l'équipe des cryptologues qui travaillaient sur la controverse Bacon-Shakespeare pour finalement invalider cette théorie.

Il s'est retrouvé à la tête du département de cryptologie où il a principalement mené un travail de cryptanalyse. Sa publication la plus importante dans ce domaine est sans conteste *L'indice de coïncidence et ses applications à la cryptanalyse* (1920). Il introduit cette notion pour décrypter des mécanismes de chiffrement reposant sur une clé répétée de manière cyclique. Lorsque la clé est trop longue, ou lorsque le matériel intercepté est insuffisant, la méthode de Kasiski peut échouer.

> *La rareté des trigrammes, des polygrammes et même des digrammes ne constitue pas forcément un obstacle, car la répartition individuelle des lettres peut être utilisée avec une grande précision pour le même objectif qui est la détermination de la période. La méthode repose sur la construction de ce que nous avons appelé la « table de coïncidence », qui nous montrera mathématiquement la valeur la plus probable de la période.*
>
> William Friedman, *L'indice de coïncidence.*

Dans un texte écrit en français, la distribution des lettres n'est pas uniforme. Par exemple, le *a* a une fréquence d'apparition d'environ 7,68 %. La probabilité de répétition d'un *a* vaut donc $0{,}0768 \times 0{,}0768$, soit environ 0,590 %. Pour la lettre *e*, la probabilité de répétition vaut $0{,}178 \times 0{,}178$. En faisant la somme de ces probabilités pour chacune des lettres de l'alphabet, on trouve une probabilité de répétition égale à 0,0778, valeur plus élevée que celle d'un texte où les lettres sont uniformément distribuées selon le hasard.

L'indice de coïncidence d'un texte est par définition la probabilité de répétition d'une lettre. Pour un texte aléatoire, la valeur est 0,0385, et sinon, la valeur dépend

de la langue d'écriture, comme l'indique la table suivante :

Français	Anglais	Russe	Arabe	Espagnol	Allemand
0,0778	0,0667	0,0529	0,0758	0,0770	0,0762

Encadré 4.3. Coïncidence.

Une coïncidence est la répétition d'une lettre dans un texte. Dans un texte où toutes les lettres sont uniformément réparties, chaque lettre a une probabilité d'occurrence égale à $1/26$. La probabilité d'observer deux fois une lettre vaut $1/26 \times 1/26$. La probabilité d'observer la répétition d'une lettre quelconque est la somme de ces probabilités pour les vingt-six lettres. La probabilité de coïncidence théorique dans ce cas est donc égale à vingt-six fois $1/26 \times 1/26$, soit $1/26 \approx 0{,}0385$.

Cette probabilité théorique est à comparer au nombre observé de coïncidences dans un texte réel. Si dans une courte phrase, la lettre a apparaît quatre fois, on y compte six coïncidences de cette lettre. Plus généralement, pour $i = 1$ à 26, si le nombre n_i y représente le nombre d'occurrences de la lettre numéro i, le nombre de coïncidences de cette lettre sera le nombre de combinaisons de deux lettres parmi n_i. Le nombre total de coïncidences dans le texte sera la somme des coïncidences de chaque lettre, soit :

$$N = \sum_{i=1}^{26} \frac{n_i(n_i - 1)}{2}$$

Si deux portions de cryptogramme sont chiffrées avec la même lettre clé, une coïncidence sur le message clair se traduit par une coïncidence sur le cryptogramme. Ainsi, si deux portions d'un cryptogramme issu d'un texte français sont superposées de manière semblable par rapport à la clé, le comptage des coïncidences fera apparaître une fréquence empirique d'environ 0,0778. À l'opposé, si elles sont décalées par rapport à la clé, les coïncidences du message clair ne correspondent pas à des coïncidences du cryptogramme, et elles n'apparaîtront qu'avec une fréquence d'environ 0,0385 qui est le taux de coïncidences d'une distribution aléatoire. Cette méthode permet de retrouver la taille de la clé d'un chiffre de Vigenère avec une très grande efficacité. Considérons le cryptogramme très court suivant :

```
seflh yjtqc rekah eosle ekush
eyeks lccmy eeryq zvvwi ucexr
```

Nous allons déterminer la longueur de la clé en cherchant la plus probable parmi toutes les longueurs possibles. Supposons pour simplifier que le cryptanalyste sait qu'elle se situe entre 9 et 15. Pour une longueur de clé supposée égale à ℓ, le cryptogramme est écrit sur ℓ colonnes. Si la supposition est juste, toutes les

lettres situées sur une même colonne sont chiffrées avec la même lettre clé. Voici la disposition obtenue pour $\ell = 9$:

s	e	f	l	h	y	j	t	q
c	r	e	k	a	h	e	o	s
l	e	e	k	u	s	h	e	y
e	k	s	l	c	c	m	y	e
e	r	y	q	z	v	v	w	i
u	c	e	x	r				
coïncidences :	1	2	3	2	0	0	0	0	0

Les lettres d'une colonne sont chiffrées avec la même lettre clé. Comptons les coïncidences par colonne. On observe un total de huit coïncidences. Répéter ce comptage pour toutes les valeurs possibles de la longueur supposée de la clé aboutit aux nombres N de coïncidences suivants :

ℓ	9	10	11	12	13	14	15
N	8	8	5	9	2	4	3

L'indice de coïncidence empirique est le quotient du nombre de coïncidences observées par le nombre de coïncidences possibles. Cette dernière quantité correspond au cas où les lettres sont identiques par colonne. Pour neuf colonnes, les cinquante lettres du cryptogramme sont réparties en cinq colonnes de six lettres, suivies de quatre colonnes de cinq lettres. Le nombre maximal de coïncidences est atteint si chaque colonne est remplie avec la même lettre. Pour les colonnes de six lettres, cela fait autant de coïncidences que de choix de deux lettres parmi six, soit $(6 \times 5)/2 = 15$. Pour les colonnes de cinq lettres, ce nombre est égal à $(5 \times 4)/2 = 10$. Ainsi le nombre maximal de coïncidences possibles vaut-il $5 \times 15 + 4 \times 10 = 115$. L'indice de coïncidence empirique I_c, pour une taille de clé supposée égale à 9, vaut $8/115 = 0{,}069565$. Le tableau 4.2 indique le résultat du calcul pour les autres longueurs de clé.

ℓ	9	10	11	12	13	14	15
							
I_c	0,069565	0,080000	0,055556	0,112500	0,027778	0,060606	0,050000

Tableau 4.2.

La valeur pour $\ell = 12$ est notablement supérieure aux autres, et c'est cette taille de clé qui est la plus vraisemblable. La poursuite du décryptement se fait comme décrit dans le précédent paragraphe, en cherchant la lettre clé la plus probable pour chaque colonne. La détermination de cette lettre peut commencer en cherchant le décalage qui conduira au plus grand nombre d'*esantirulo* et se poursuivre avec une étude plus fine se fondant sur l'analyse de la langue, les combinaisons possibles et impossibles de lettres, etc. (solution page 175).

4.4 La cryptanalyse anglaise pendant la deuxième guerre mondiale

Pendant la deuxième guerre mondiale, l'Angleterre a attaché une grande importance au renseignement et en particulier au décryptement des messages ennemis. Les services cryptologiques du gouvernement anglais étaient situés à Bletchley Park, lieu maintenu secret, situé au nord de Londres à la bifurcation des routes vers les universités d'Oxford et de Cambridge. Ce centre était peu important avant la seconde guerre mondiale, mais pendant celle-ci, il a employé jusqu'à dix mille personnes, surtout des femmes, les hommes étant mobilisés sur le front. Bletchley Park était au cœur d'un réseau d'interceptions, les *Y stations*, réparties sur l'ensemble du territoire anglais. Les messages en provenance des sous-marins allemands étaient interceptés par les stations côtières, puis acheminés à Bletchley Park par tous les moyens, parfois même à bicyclette.

Avant le conflit, les mathématiciens polonais Marian Rejewski (1905-1980), Jezny Rozycki (1909-1942) et Henryck Zygalski (1907-1978) ont réalisé une cryptanalyse de la machine Enigma. Des mots probables du message en clair permettent de retrouver la clé par recherche exhaustive. Le nombre gigantesque de combinaisons possibles a nécessité d'investir dans des moyens techniques et théoriques à la hauteur du défi à relever. Cela débouchera sur les premiers dispositifs électromécaniques de résolution, les *bombes*, appelées ainsi en raison du tic-tac sonore qu'elles émettaient pendant leur fonctionnement. La réduction de la complexité de la recherche s'est aussi appuyée sur des considérations théoriques quant à la décomposition des permutations en cycles. Comme la substitution réalisée par la machine Enigma résulte de la composition des permutations des trois rotors, du réflecteur, puis des trois mêmes rotors dans l'ordre inverse, la structure cyclique de la permutation résultante se trouve être identique à celle du seul réflecteur, constituée seulement d'échanges de deux lettres. Toutefois, la recherche exhaustive restait encore trop complexe pour aboutir avec les moyens de calcul disponibles en Pologne.

Lorsque la Pologne a été envahie par l'Allemagne, les informations sur la cryptanalyse d'Enigma ont été transmises en France, puis en Angleterre lorsqu'à

son tour la France a été envahie. Selon David Kahn, c'est l'écrivain Sacha Guitry qui aurait transporté les documents jusqu'à Londres.

Les cryptanalystes de Bletchley Park, avec en particulier la participation du mathématicien Alan Turing (1912-1954), ont alors affiné la méthode de recherche, l'ont adaptée aux différentes évolutions de la machine Enigma, et ont amélioré les bombes électromécaniques, permettant ainsi le décryptement effectif des messages transmis par les forces allemandes. Il est reconnu que cette activité a conféré aux Alliés un avantage notable.

Un second résultat essentiel des équipes de Bletchley Park a été la construction du premier calculateur électronique, le *Colossus*, qui comprenait 2 500 tubes à vide, et qui était destiné à décrypter les messages chiffrés avec la machine de Lorenz. Cette machine est un chiffrement à flot pour téléscripteur, utilisant le principe du chiffrement de Vernam (voir paragraphe 4.1 page 26), dans lequel la bande aléatoire est remplacée par un générateur pseudo-aléatoire électromécanique utilisant douze roues codeuses. La cryptanalyse de cette machine a été rendue possible par des erreurs d'utilisation. En effet, les messages devaient être répétés en raison des interférences radio qui troublaient la réception correcte du signal et, lors de la deuxième émission, les roues codeuses étaient réinitialisées dans la même position, avec pour conséquence l'utilisation d'une séquence pseudo-aléatoire identique. Mais l'utilisation d'abréviations pour le second message faisait que celui-ci n'était pas identique au premier, et certaines informations du message en clair se trouvaient alors connues. Le cryptanalyste pouvait s'appuyer sur cette connaissance pour mener à bien le décryptement.

La machine Colossus constitue un progrès considérable par rapport aux bombes électromécaniques. L'innovation essentielle réside dans l'utilisation pour la première fois de l'électronique, avec des tubes à vide, qui réalisent les opérations bien plus rapidement que les relais des machines électromécaniques. Cette machine a été conçue par l'ingénieur des téléphones Tommy H. Flowers (1905-1998). La technologie des tubes à vide était auparavant utilisée pour accélérer le fonctionnement des commutateurs téléphoniques. Après le conflit, le Premier ministre britannique Wiston Churchill ordonnera la destruction de Colossus, comme matériel de guerre touché par le secret-défense. La réalisation de Colossus inspirera les premiers ordinateurs anglais, grâce au savoir-faire des mathématiciens et des ingénieurs toujours mobilisés sur la question. Il faudra attendre quelques années pour voir réapparaître en Grande Bretagne le calculateur EDSAC (*Electronic Delay Storage Automatic Calculator*), dont la construction est achevée à Cambridge en mai 1949, et qui est considéré comme le premier ordinateur avec des programmes en mémoire interne. Le premier ordinateur américain, l'EDVAC (*Electronic Discrete VAriable Computer*), date lui de

1952. Il a été conçu avec la participation du mathématicien d'origine hongroise John Von Neuman.

5 Les cryptanalyses des chiffrements modernes

Les chiffrements modernes ne traitent plus la langue naturelle écrite avec un alphabet, mais une information plus générale décrite avec les symboles 0 et 1. La cryptanalyse quitte le domaine de l'analyse du langage pour entrer dans celui du calcul.

5.1 Les modèles d'attaque

Les attaques contre les procédés cryptographiques se classent selon l'information à laquelle l'adversaire a accès. Le principe de Kerckoffs est maintenant universellement admis. La description du mécanisme mis en œuvre, en particulier l'algorithme et le protocole, est une donnée qui est supposée connue de l'adversaire. La confidentialité repose exclusivement sur le secret de la clé. Cependant, l'adversaire peut disposer d'informations complémentaires pour mener son attaque.

Attaque « *à clair connu* »

Dans une attaque à clair connu, l'adversaire dispose d'un certain nombre de messages en clair et du cryptogramme correspondant. C'est le cas par exemple lorsque d'anciens messages chiffrés sont dévoilés, ou lorsqu'ils commencent tous par le même en-tête, comme la date ou le lieu d'origine, ou encore lorsqu'ils finissent par une formule convenue.

Attaque « *à clair choisi* »

Lorsque l'adversaire peut choisir les messages en clair dont il peut connaître le cryptogramme, on parle d'attaque *à clair choisi*. Dans un système à clé publique, la clé de chiffrement étant publique, l'adversaire peut chiffrer les messages de son choix, l'attaque à clair choisi est toujours applicable.

Attaque « *à cryptogramme choisi* »

Dans une attaque à cryptogramme choisi, l'adversaire dispose temporairement d'un dispositif qui peut lui fournir le message clair qui correspond à un cryptogramme de son choix. Un tel dispositif s'appelle un *oracle de déchiffrement*. Il s'agit surtout d'un modèle théorique destiné à établir la solidité d'un procédé

de chiffrement, mais on peut imaginer par exemple qu'un espion s'introduise discrètement dans le local du directeur, accède à son poste de travail pour déchiffrer certains cryptogrammes de son choix, et ainsi accumule assez d'information pour mener un futur décryptement.

5.2 La cryptanalyse différentielle

La cryptanalyse différentielle a été introduite par les cryptologues israéliens, Eli Biham et Adi Shamir, en 1990, contre l'algorithme DES. Elle s'applique à tous les procédés qui procèdent par itération sur plusieurs tours et appartient à ce qu'on appelle aujourd'hui la famille des attaques sur le dernier tour. Il s'agit d'une attaque à clair choisi. Il faut connaître les chiffrés de 2^{47} messages choisis pour réussir cette attaque, ce qui est en pratique très difficile. Cependant, elle met en évidence des faiblesses de conception et conduit à des critères pour lui résister.

Cette attaque cherche à reconstituer la sous-clé du dernier tour, ou du moins une partie de cette sous-clé. Celle-ci étant connue, les autres sous-clés sont retrouvées de la même façon. Lorsque suffisamment de sous-clés sont connues, le reliquat peut être retrouvé par recherche exhaustive. L'efficacité de la méthode repose sur la notion de *différentielle*. Il s'agit par définition d'un couple de données (α,β) telle que si l'on applique des données qui diffèrent de α à l'entrée du calcul, on trouve, à la sortie de l'avant-dernier tour, des données qui diffèrent de β avec une probabilité notablement supérieure à la probabilité uniforme attendue qui, dans le cas du DES, vaut $1/2^{64}$.

Rechercher une différentielle est un travail délicat et demande un examen attentif et détaillé de tous les éléments de la fonction de chiffrement. Le principe pour les trouver est de composer les meilleures différentielles pour chaque tour. Toutefois, ce travail laborieux n'est à effectuer qu'une fois pour toutes.

Comme il s'agit d'une attaque à clair choisi, le cryptanalyste dispose de la connaissance des cryptogrammes pour des messages en clair de son choix. Pour mener son attaque, il examine les cryptogrammes de deux messages dont la différence vaut α, puis il estime les valeurs au niveau de l'avant-dernier tour. Pour cela, il compose les cryptogrammes avec la fonction inverse du dernier tour en essayant toutes les sous-clés possibles. Si la sous-clé essayée est égale à la sous-clé inconnue réellement utilisée sur le dernier tour, le résultat observé correspondra exactement à la sortie de l'avant-dernier tour. Il suffit alors de répéter cette expérience un grand nombre de fois, et de compter les sous-clés pour lesquelles l'estimation de la différence à l'avant-dernier tour vaut β. Cette opération est répétée jusqu'à ce qu'une sous-clé soit significativement comptée plus que les autres. En raison des propriétés de la différentielle (α,β), cette sous-clé sera très probablement la bonne. Cette attaque est schématisée sur la figure 4.3.

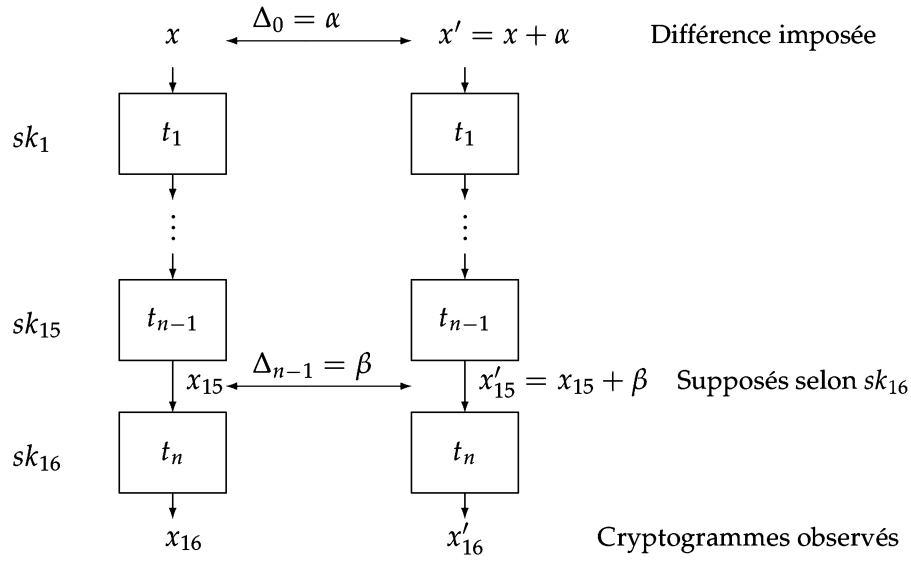


FIGURE 4.3. Cryptanalyse différentielle d'un chiffrement itéré. On chiffre deux messages x et x' qui diffèrent de α. On observe les cryptogrammes $y = x_{16}$ et $y' = x'_{16}$. Par une recherche exhaustive, on compte les sous-clés qui sont compatibles avec une différence estimée égale à β au niveau de la sortie de l'avant-dernier tour. La sous-clé qui est la plus souvent comptée est vraisemblablement la bonne.

5.3 La cryptanalyse linéaire

La cryptanalyse linéaire est une attaque à clair connu, découverte contre le DES en 1993 par Mitsuru Matsui, chercheur à Mitsubishi. Elle nécessite, pour réussir 2^{43} couples clair-cryptogramme connus, ce qui fait d'elle l'attaque théorique la plus efficace contre le DES. Mais le grand nombre de données nécessaires pour la mener à bien la rend très difficile à réaliser et on lui préfère la recherche exhaustive de la clé parmi les 2^{56} possibles. Comme la cryptanalyse différentielle, elle est applicable aux chiffrements par bloc itérés et permet de dégager des critères de conception pour lui résister.

Cette attaque consiste à retrouver des équations linéaires satisfaites par la clé. Pour trouver la clé dans son ensemble, il faut l'appliquer plusieurs fois, par exemple vingt fois, pour trouver vingt équations linéaires satisfaites par la clé. Cela permet d'exprimer vingt composantes binaires de la clé en fonction des autres qui sont ensuite trouvées par recherche exhaustive.

On utilise pour cela une approximation linéaire de l'algorithme, c'est-à-dire un triplet d'ensembles de positions de symboles binaires sur le message clair, la clé et le cryptogramme tels qu'en additionnant modulo 2 les valeurs des symboles binaires du message clair, de la clé et du cryptogramme à ces positions, on trouve une valeur biaisée, qui vaut significativement plus souvent 1 que 0 ou le contraire.

Lorsqu'on dispose d'un grand nombre de couples clair-cryptogramme connus, non nécessairement choisis, en appliquant l'approximation linéaire, on trouvera

une estimation d'une équation linéaire satisfaite par la clé. La confiance sera d'autant plus grande que le calcul aura été fait avec un grand nombre de données. On réitère alors ce processus avec d'autres approximations linéaires jusqu'à accumuler assez de résultats pour reconstituer la clé.

5.4 Les attaques algébriques

Les attaques algébriques ont été d'abord appliquées contre le chiffrement à flot. Elles consistent à écrire les équations de la fonction de chiffrement, où l'inconnue est la clé secrète. Chaque observation d'un terme de la suite chiffrante conduit à une équation. Avec un nombre suffisant d'observations, on obtient un système dont la solution est unique. Les équations ainsi obtenues sont des équations non linéaires complexes qui font intervenir les symboles binaires de la clé k_i, mais aussi des produits de deux symboles binaires $k_i k_j$, de trois symboles binaires $k_i k_j k_\ell$, etc. Pour résoudre un tel système, une technique est de le rendre linéaire en remplaçant chaque produit de symboles binaire $k_i k_j$, ou $k_i k_j k_\ell$ par une nouvelle inconnue k_{ij} ou $k_{ij\ell}$, et pareillement pour tout autre produit. On obtient alors un système d'équations linéaires, qu'il est possible de résoudre, à condition toutefois qu'il ne comprenne pas trop d'inconnues.

Ce qui va bloquer le cryptanalyste, c'est le degré des équations, c'est-à-dire le nombre maximal de facteurs dans un produit de symboles binaires de la clé $k_{i_1} k_{i_2} \cdots k_{i_d}$ constituant une équation. Plus ce degré est élevé, et plus le nombre d'inconnues à ajouter sera important, plus le nombre d'équations nécessaires pour le résoudre le sera aussi. Si le concepteur a bien fait son travail, suivant le principe de confusion, la résolution n'est pas envisageable en raison du trop grand nombre d'inconnues et donc du trop grand nombre d'équations requises. Il est possible toutefois de remplacer certaines équations par d'autres, de degré moindre, par exemple en faisant intervenir plusieurs symboles consécutifs de la suite chiffrante et en éliminant des produits d'inconnues. Un examen attentif et soigneux de la fonction de chiffrement peut alors mettre en évidence des faiblesses et conduire à une possible résolution du système.

Ce type d'attaque a été menée avec succès contre un algorithme de chiffrement à flot, appelé E_0, utilisé dans les liaisons sans fil à courte distance *bluetooth*.

6 La factorisation des entiers

La cryptographie à clé publique s'appuie sur des problèmes réputés difficiles à résoudre. L'approche immédiate pour attaquer cette cryptographie est de tenter de trouver une solution au problème sur lequel elle repose. Pour le RSA, il s'agit

de la factorisation des entiers. La première méthode de cryptanalyse du RSA est donc la recherche des facteurs du module public.

Factoriser les entiers a depuis longtemps fait l'objet de nombreux travaux. Dans une lettre à Mersenne datant de 1643, Fermat décrit une méthode pour factoriser l'entier 2 027 651 281. En 1926, le Français d'origine russe Maurice Kraitchik propose une amélioration de la méthode de Fermat. L'avènement du RSA donnera un nouvel élan aux recherches sur ce problème, conduisant à des progrès significatifs. Le tableau 4.3 indique l'année de factorisation et la taille des entiers qui ont pu être factorisés.

Année de factorisation	1994	1999	2003	2005	2010
Nombre de chiffres décimaux	129	155	174	200	232

Tableau 4.2.

Pour factoriser un entier on peut chercher à l'exprimer comme différence de deux carrés. Lorsque cela est établi, l'application d'une identité remarquable bien connue des collégiens fournit la factorisation cherchée :

$$n = a^2 - b^2 = (a + b)(a - b)$$

La méthode de Fermat consiste à prendre, comme point de départ de la recherche, un entier qui approche la racine carrée de l'entier à factoriser. Cherchons par exemple à factoriser l'entier $n = 2\,183$. Sa racine carrée entière approchée par excès est $a = 47$. Calculons $47^2 - 2\,183$ qui vaut 26. L'entier 26 n'est pas un carré. Essayons maintenant avec $a = 48$. Cela conduit à $48^2 - 2\,183 = 121$. Mais cette fois, 121 est le carré de 11 et nous pouvons fièrement annoncer :

$$2\,183 = 48^2 - 11^2 = (48 + 11)(48 - 11) = 59 \times 37$$

Cette méthode est efficace lorsque l'entier à factoriser est le produit de deux entiers proches l'un de l'autre. L'obtention d'un carré peut s'avérer plus longue que dans cet exemple. Il faudra onze étapes à Fermat pour arriver à factoriser :

$$2\,027\,651\,281 = (45\,041 + 1\,020)(45\,041 - 1\,020) = 46\,061 \times 44\,021$$

Kraitchik a amélioré la méthode en remarquant qu'il n'est pas nécessaire d'attendre l'obtention d'un carré, mais qu'on peut combiner certains résultats pour obtenir un carré. Cherchons par exemple à factoriser l'entier 1 649. Sa racine carrée entière par excès vaut 41, et les premières étapes de la méthode exposée

ci-dessus sont :

$$41^2 - 1\,649 = 32$$
$$42^2 - 1\,649 = 115$$
$$43^2 - 1\,649 = 200$$

Nous n'avons pas obtenu de carré, mais notons que $32 = 2^5$ et que $200 = 2^3 \times 5^2$. Le produit 32×200 vaut donc $2^8 \times 5^2$. Les exposants dans cette décomposition sont tous pairs, montrant qu'il s'agit d'un carré : $32 \times 200 = (2^4 \times 5)^2 = 80^2$. Nous tenons notre décomposition en différence de deux carrés en multipliant la première et la troisième relation. En effet, 41×43 est congru à 114 modulo 1 649, ce qui permet de conclure que 114^2 et 80^2 diffèrent d'un multiple de 1 649. Le produit $(114 - 80)(114 + 80)$ est donc multiple de 1 649 sans qu'aucun des facteurs ne le soit. Cela signifie que les facteurs de 1 649 sont répartis entre $114 - 80$ et $114 + 80$. Le calcul du plus grand diviseur commun à $114 - 80$ et 1 649 avec l'algorithme d'Euclide permettra de conclure. Effectivement, $\text{pgcd}(114 - 80, 1\,649) = 17$, d'où $1\,469 = 17 \times 97$.

Au cours de cette méthode, on a construit des relations, comme par exemple $41^2 - 1\,649 = 2^5$, où l'on a su factoriser le second membre. Pour obtenir cette factorisation, on se contente d'essayer de diviser par des nombres premiers choisis dans un ensemble fixé, tous inférieurs à une certaine borne. Si un résultat n'est pas factorisable, on abandonne la relation et on passe à la suivante. Les nombres que l'on arrive à factoriser par cette méthode ont tous leurs facteurs premiers inférieurs à la borne choisie. Un tel nombre est appelé un *nombre lisse*. Pour réussir la factorisation, on doit accumuler un nombre suffisant de relations avec second membre lisse afin de les combiner et obtenir qu'un produit de certaines d'entre elles n'aient que des exposants pairs dans leur décomposition en facteurs premiers. Cette combinaison s'obtient en résolvant un système d'équations linéaires modulo 2.

Le choix du plus grand nombre premier pour la factorisation des seconds membres est un élément critique pour déterminer l'efficacité de la méthode. S'il est choisi trop petit, il y a moins de chance de pouvoir factoriser, et beaucoup de relations seront rejetées. S'il est choisi trop grand, il faudra davantage de relations pour espérer pouvoir les combiner et obtenir un produit où tous les exposants sont pairs. Un examen attentif de la distribution des nombres lisses permet un choix optimal de la borne.

De nombreux raffinements ont été portés à cet algorithme, pour aboutir à l'algorithme appelé *crible quadratique multi-polynômes* qui est le premier algorithme de factorisation dont la complexité a été prouvée sous-exponentielle en la taille

de l'entier à factoriser. C'est cet algorithme qui a permis de factoriser le module RSA de 129 chiffres de la revue *Scientific American* en 1994.

Depuis, d'autres progrès ont encore été accomplis. À partir de résultats très fins de la théorie algébrique des nombres, un algorithme appelé *crible général du corps de nombres* a été mis au point. Il est à ce jour asymptotiquement le plus efficace et c'est lui qui est utilisé pour battre les records de factorisation.

Encadré 4.4. Entre complexité polynomiale et exponentielle.

Pour évaluer la complexité de factorisation d'un entier N, on utilise la fonction suivante, qui comprend deux paramètres : un degré k et un curseur α compris entre 0 et 1 :

$$L_{k,\alpha}(N) = \exp\left(k(\ln N)^{\alpha}(\ln \ln N)^{1-\alpha}\right).$$

Pour un entier N de n chiffres décimaux, $\ln N$ vaut sensiblement an, où $a = \ln 10$.

- Lorsque $\alpha = 0$, cette fonction est le monôme de degré k en n égal à $(an)^k$.
- Lorsque $\alpha = 1$, elle est exponentielle en n, égale à e^{kan}.

Entre ces deux bornes, la fonction domine asymptotiquement tout polynôme, mais est dominée par toute fonction exponentielle. On parle de complexité *sous-exponentielle*. Le paramètre α agit comme un curseur qui fera approcher la fonction $L_{k,\alpha}(N)$ d'un polynôme ou d'une exponentielle en $n = \ln N / \ln 10$ selon que α s'approchera de 0 ou de 1.

Pour l'algorithme du crible quadratique multipolynôme, le curseur α vaut $\frac{1}{2}$ et le degré k vaut 2. Pour le crible général du corps de nombres, le curseur est $\frac{1}{3}$ et le degré $\left(\frac{64}{9}\right)^{1/3}$. Le gain obtenu entre ces deux algorithmes est considérable. Il est près de deux millions de fois plus long de factoriser un nombre de 100 chiffres qu'un nombre de 50 chiffres avec le premier algorithme, alors qu'avec le second, ce n'est qu'un peu plus de mille fois plus long. Avec l'algorithme de Fermat dont la complexité est exponentielle de degré 1/2, le quotient des complexités est de l'ordre de 10^{41}.

7 Les attaques physiques

Les attaques décrites jusqu'à présent cherchent à mettre en défaut la logique du procédé dans son aspect mathématique. À l'opposé, les attaques physiques cherchent à extraire matériellement les secrets enfouis dans des dispositifs qui exécutent les calculs cryptographiques. Les attaques sur les canaux secondaires (*Side-channel attacks*) exploitent la fuite d'information par des voies inattendues, non prévues par le concepteur, comme la consommation électrique ou le temps de calcul. Les attaques par faute tentent d'extraire des données en faisant fonctionner le composant dans des conditions anormales, jusqu'à provoquer des erreurs de calcul.

Ces attaques sont connues depuis assez longtemps. Dans ses mémoires, *Spycatcher*, l'officier du contre-espionnage britannique Peter Wright en raconte un exemple contre l'ambassade de France à Londres :

> *Dans les années cinquante, le plus sophistiqué des procédés de chiffrage consistait à taper le texte en clair sur une sorte de télégraphe relié à une machine à chiffrer. À l'autre bout, la machine délivrait en cliquetant un message chiffré. La sécurité de tout ce procédé dépendait uniquement d'une bonne isolation. Si l'isolation électromagnétique entre le télégraphe et la machine à chiffrer était insuffisante, il était possible que l'écho du texte en clair subsiste le long des câbles parallèlement au texte codé.*
> *(...) Une courbe régulière palpitait sur mon écran de contrôle. On voyait très clairement la superposition de deux courbes : celle de la machine à chiffrer et celle du « fantôme » du texte clair qui l'accompagnait.*

Empêcher le rayonnement compromettant des machines qui manipulent des données sensibles, comme par exemple les écrans d'ordinateur, a été une préoccupation constante des milieux militaires qui ont pour cela élaboré des normes appelées normes TEMPEST (*Telecommunications Electronic Material Protected from Emanating Spurious Transmissions*).

Plus récemment, en 1996, le cryptologue américain Paul Kocher a développé des méthodes pour retrouver des secrets enfouis dans des dispositifs comme les cartes à puce en exploitant la mesure du temps de calcul ou de la consommation électrique. Cela permettrait par exemple à un terminal de paiement, lors d'une transaction et par simple mesure du courant pendant le travail de la carte bancaire, d'en extraire les clés privées. Les fabricants de cartes ont heureusement pris des mesures pour parer ces attaques.

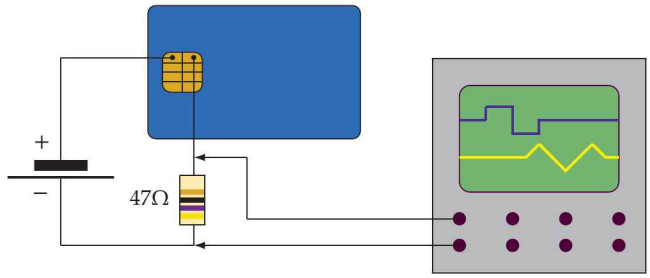


FIGURE 4.4. Banc de mesure pour analyser la consommation d'une carte à puce pendant la réalisation d'un calcul cryptographique : la consommation du dispositif est mesurée et mémorisée en vue d'une analyse statistique. Un banc similaire permet de mesurer avec précision le temps d'exécution.

Prenons une carte à puce qui calcule la signature RSA d'un document. Pour cela, on lui fournit son condensé qu'elle élève à la puissance exposant privé. Cet exposant est une donnée secrète qui ne doit pas sortir de la carte. Une méthode efficace pour réaliser ce calcul consiste à effectuer une succession de multiplications et d'élévations au carré selon la décomposition binaire de l'exposant. Pour élever une donnée à une certaine puissance, les chiffres binaires de l'exposant sont parcourus de gauche à droite. À chaque étape, on élève au carré le résultat courant et, lorsque le chiffre de l'exposant est 1, on le multiplie par la donnée. Avec cette façon de procéder, savoir quand le dispositif réalise une multiplication fournit directement la décomposition binaire de l'exposant.

Or les dispositifs électroniques ont une consommation électrique qui dépend de leur activité. Cette consommation est directement proportionnelle au nombre d'éléments du circuit qui changent d'état, de 0 à 1, ou de 1 à 0. Le calcul en cours laisse des traces sur la consommation qui fait donc fuir des informations qui devraient rester confinées à l'intérieur du circuit. Il est également possible d'exploiter le rayonnement électromagnétique du composant. Avec une sonde très précise, il est même possible de cibler la partie du composant qui traite les données sensibles, comme l'unité centrale de calcul. Les circuits sont aujourd'hui dotés de contre-mesures pour limiter l'observabilité de l'activité du composant de l'extérieur.

La figure 4.5 montre comment l'observation fine de la consommation dévoile directement l'exposant privé d'une clé RSA. On parle d'analyse simple de consommation (SPA, *Simple Power Analysis*). D'autres méthodes plus élaborées ont été développées pour tenter de dévoiler la clé secrète d'un chiffrement symétrique. Pour cela, on fait réaliser au circuit visé un grand nombre d'opérations de chiffrement. Les courbes de consommation sont alors mémorisées et réparties en deux classes selon la valeur d'une fonction de sélection qui fait une hypothèse sur la valeur d'un symbole binaire de la clé. Si l'hypothèse est bonne, les deux classes présenteront des différences notables qui valideront l'hypothèse, et dans le cas contraire, rien ne distinguera particulièrement une classe de l'autre et l'hypothèse pourra être rejetée. La méthode sera d'autant plus efficace que la fonction de sélection choisie pour classer les courbes de consommation montrera une forte dépendance entre le secret supposé et l'activité du composant. Cette technique, appelée *analyse différentielle de consommation* (DPA, *Differential Power Analysis*) a été utilisée avec succès sur les premières générations de cartes à puce pour extraire la clé secrète lors d'un chiffrement avec l'algorithme DES.

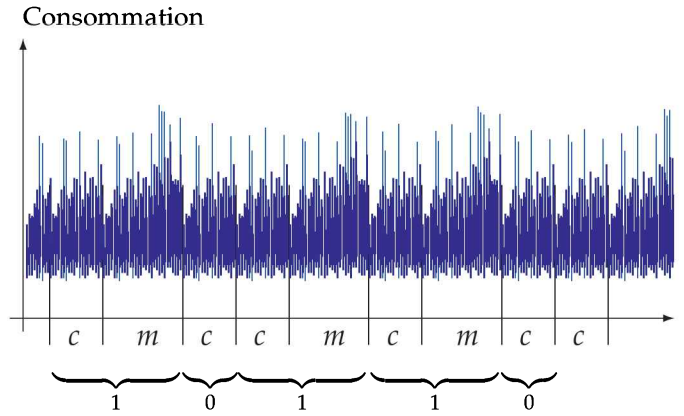


FIGURE 4.4. Analyse de consommation sur un dispositif réalisant un calcul RSA. La courbe de consommation permet de discerner assez clairement les multiplications m des élévations au carré c. Cela dévoile directement les chiffres binaires de l'exposant privé.

7.2 Le temps de calcul

Un autre paramètre qui fait fuir de l'information est le temps de calcul. Sur certaines données, le calcul est plus lent que d'autres, par exemple lorsque l'algorithme comprend un test et qu'une opération supplémentaire doit être effectuée en cas de test positif. Dans ce cas, le calcul est plus lent et le cryptanalyste ne manquera pas d'exploiter cette information. Prenons par exemple l'algorithme de réduction de Montgomery, très populaire pour les calculs RSA, présenté dans l'encadré 3.13 page 73. Il se termine par une soustraction lorsque le résultat est supérieur au module et, dans ce cas, cela ralentit légèrement le calcul.

Dans un calcul RSA, la cible de l'adversaire est la valeur de l'exposant privé. Selon les valeurs de ses chiffres binaires, l'algorithme d'exponentiation effectuera ou non une multiplication. Si l'adversaire peut déterminer si une multiplication a ou n'a pas lieu, il saura en déduire si un symbole binaire de l'exposant privé vaut 1 ou vaut 0. Il peut procéder itérativement en déterminant leurs valeurs les unes après les autres. Le premier chiffre binaire de l'exposant est toujours 1. Ensuite, l'algorithme effectue un carré, suivi ou non d'une multiplication selon la valeur du deuxième chiffre binaire de l'exposant. L'adversaire peut alors imposer le calcul sur un ensemble de valeurs qui provoqueront une soustraction finale dans la réduction de Montgomery après élévation au carré, et également sur un autre ensemble de valeurs qui n'en provoqueront pas. Si le deuxième chiffre de l'exposant privé vaut 1, la multiplication attendue est exécutée, et cela se traduira par une différence notable dans les temps de calcul moyens entre ces deux familles de nombres. Les opérations effectuées ensuite nécessiteront ou

non la soustraction finale, mais cela sera réparti selon le hasard et ne conduira pas à une différence statistiquement notable. À l'opposé, si le deuxième chiffre de l'exposant vaut 0, alors la première multiplication n'a pas lieu et aucune différence notable de temps de calcul ne sera observable entre ces deux familles. Une fois connus ces deux premiers chiffres binaires, on peut contrôler la valeur qui sera multipliée ou non à la troisième étape. On procède de même pour évaluer le troisième chiffre de l'exposant privé, et ainsi de suite pour tous les autres.

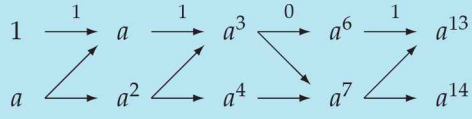
Cet exemple simple montre à quel point il est crucial de programmer les fonctions cryptographiques pour que le temps d'exécution soit indépendant des données traitées. Cela peut conduire à des traitements moins rapides, car il faut s'aligner sur le pire des cas. Pour éliminer cette fuite, il faut agir sur deux fronts. Le premier concerne la réduction de Montgomery qu'il faut programmer de telle sorte qu'elle prenne toujours le même temps à s'exécuter. Une solution est de ne pas effectuer la soustraction finale à chaque étape, de se contenter d'un résultat partiellement réduit, la réduction n'étant effectuée qu'au dernier moment. Le deuxième concerne l'algorithme d'exponentiation. Il existe une méthode, appelée *échelle de Montgomery*, qui exécute systématiquement une élévation au carré et une multiplication à chaque itération, de telle sorte qu'on ne puisse pas différentier les chiffres binaires de l'exposant.

Encadré 4.5. L'échelle de Montgomery.

Pour calculer a^n en temps constant, Peter Montgomery a proposé cette méthode appelée en son honneur *échelle de Montgomery*. Elle consiste à calculer à chaque étape deux puissances successives a^k et a^{k+1}. Le point de départ de l'algorithme, pour $k = 0$, est le couple $(1,a)$. Ensuite, les chiffres binaires de l'exposant sont parcourus de gauche à droite :

- lorsque ce chiffre est un 0, étant donné qu'en numération binaire, ajouter un zéro revient à multiplier par deux, il faut passer de l'exposant k à l'exposant $2k$. Ceci est réalisé par la transition $(a^k,a^{k+1}) \rightarrow (a^{2k},a^{2k+1})$. Le premier terme est le carré de a^k et le second terme est le produit $a^k \times a^{k+1}$;
- lorsque ce chiffre est un 1, il faut cette fois passer de l'exposant k à l'exposant $2k+1$, par la transition $(a^k,a^{k+1}) \rightarrow (a^{2k+1},a^{2k+2})$. Le premier terme est le produit $a^k \times a^{k+1}$ et le second terme est le carré de a^{k+1}.

Une fois tous les chiffres de l'exposant parcourus, le résultat est (a^n,a^{n+1}). La méthode est illustrée ci-après pour le calcul de a^{13}. Rappelons que l'écriture de 13 en numération binaire est 1101.



Chaque itération exécute systématiquement une élévation au carré et une multiplication. Ce n'est pas optimal, mais le temps de calcul est constant, empêchant ainsi le temps de calcul de faire fuir de l'information sur les données traitées.

Pour accélérer les calculs du système RSA, on a imaginé d'utiliser le théorème chinois des restes : plutôt que d'effectuer une élévation à la puissance modulo n, comme n est connu pour être le produit de deux nombres premiers p et q, on effectue deux exponentiations, l'une modulo p et l'autre modulo q. Le résultat modulo n est alors reconstitué à partir de ces deux résultats partiels. Bien sûr cela ne tient que du côté de la clé privée, car il n'est pas envisageable de révéler la factorisation. Mais l'exposant public est en général assez court et le calcul est rapide. Comme ces deux exponentiations opèrent sur des entiers dont la taille est la moitié de celle de n, leur exécution est huit fois plus rapide et, comme il y en a deux à calculer, le gain escompté dans cette opération est un facteur 4, ce qui est loin d'être négligeable.

Malheureusement, si aucune précaution n'est prise, cette façon de faire est sensible à une attaque par faute. Choisissons un message m, qui est un entier compris entre 0 et $n-1$, élevons-le à la puissance exposant public et demandons à un dispositif de déchiffrer. Si tout se passe bien, la réponse qu'il fournira sera m. Mais si, pendant le calcul, nous avons réussi à stresser le composant pour lui faire commettre une erreur pendant l'une des deux exponentiations seulement, par exemple pendant l'exponentiation modulo q, alors le résultat fourni sera correct modulo p, mais incorrect modulo q. Cela a pour conséquence que la différence avec le message attendu sera bien un multiple de p mais ne sera pas un multiple de q. Une simple application de l'algorithme d'Euclide pour déterminer le plus grand diviseur commun à cette différence et à n fera apparaître le facteur p, dévoilant ainsi la factorisation de n.

Pour forcer un composant à exécuter un calcul faux, on peut par exemple jouer sur sa tension d'alimentation en l'abaissant temporairement en dessous de la limite normale de fonctionnement. Il est également possible de le cadencer plus rapidement qu'il ne peut le supporter.

Cela ne signifie pas qu'il faille abandonner cette optimisation du calcul RSA avec l'exposant privé, mais il est essentiel que le dispositif ne fournisse pas de réponse erronée. Pour cela, on procède au chiffrement du résultat avec l'exposant public pour vérifier que le résultat est bien égal à la donnée qui vient d'être déchiffrée. En cas d'incohérence, le composant ne doit pas retourner le résultat, mais seulement signaler un échec de l'opération. Comme l'exposant public est souvent un entier assez court, cette vérification ne grève pas trop le temps de calcul.

5

La cryptographie au quotidien

À travers quelques exemples, de l'internet à la carte bancaire et du téléphone mobile à la télévision à péage, ce chapitre décrit quelques applications familières de la cryptographie.

1 Les infrastructures de gestion des clés publiques

La cryptologie à clé publique résout le problème de l'échange discret entre deux correspondants qui communiquent sans s'être préalablement entendus sur une clé secrète commune. Pour transmettre un message confidentiel, il me suffit de connaître la clé publique du destinataire. Et pour cela, je la lui demande et il peut me la communiquer publiquement. Mais cette clé publique que je reçois, est-ce bien la sienne ? N'est-ce pas un intrus qui se fait passer pour mon correspondant ? Comment puis-je m'en assurer ? Les infrastructures de gestion des clés publiques (*Public Key Infrastructure*, PKI) sont une réponse à ce problème.

Les clés publiques sont transmises accompagnées d'un certificat qui consiste en une signature de celle-ci, assurée par un organisme officiel appelé *autorité de certification*. Les clés publiques des autorités de certifications suffisent pour vérifier la signature présente dans le certificat et ainsi assurer la confiance dans l'identité et la clé publique de mon destinataire.

Il existe des entreprises internationales reconnues comme autorité de certification, comme VeriSign, GlobalSign ou Entrust. La plupart des banques agissent également comme autorité de certification pour délivrer les certificats des clés publiques présentes dans les cartes de paiement. Au sein d'une entreprise, le service informatique peut également agir comme autorité de certification locale en charge de produire les certificats des clés publiques des membres (Fig. 5.1).

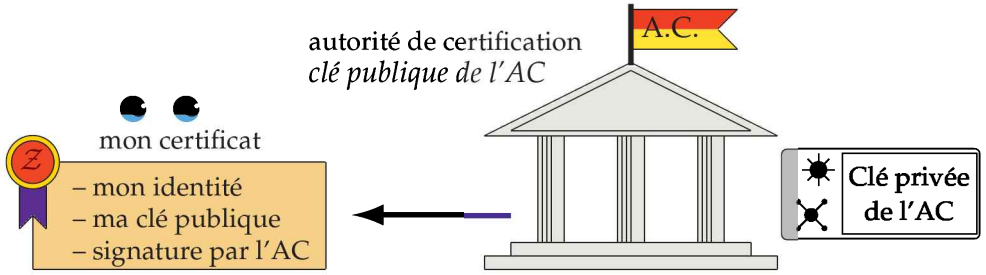


FIGURE 5.1. Délivrance d'un certificat par une autorité de certification. La signature présente dans mon certificat est vérifiable par tous mes correspondants avec la clé publique de l'autorité de certification. Ils peuvent alors utiliser en toute confiance la clé publique de mon certificat pour chiffrer des messages à mon intention.

Lorsque l'autorité de certification n'est pas suffisamment reconnue, sa clé publique de vérification des certificats est elle-même munie d'un certificat émanant d'une autorité supérieure. Un certificat peut contenir une liste de signatures et les clés publiques certifiées requises pour vérifier ces signatures, jusqu'à un certificat émanant d'une autorité assez connue pour que sa clé publique soit assurément authentique. On parle alors de *chaîne de certification*.

Il existe une alternative à la certification des clés publiques délivrée par une autorité centralisée telle qu'elle est décrite ci-dessus, il s'agit de la propagation de la confiance de proche en proche, de pair à pair. Si je dispose de la clé publique d'un correspondant en qui j'ai toute confiance, et si ce dernier a confiance dans la clé publique d'un autre correspondant, alors il peut me transférer sa confiance en signant cette clé publique. Cela permettra à mon tour d'avoir confiance en cette clé, mais également de transmettre cette confiance à d'autres. Ce mécanisme fonctionne aussi dans l'autre sens ; il peut propager la révocation d'une clé publique douteuse. Cette philosophie décentralisée est celle mise en œuvre dans le logiciel de protection des courriers électroniques PGP (*Pretty Good Privacy*).

2 La carte bancaire

2.1 Historique

La première carte de paiement en matière plastique date de 1950. Il faudra attendre 1970 pour que les premières cartes à piste magnétique apparaissent. Une telle carte constitue seulement une manière commode d'emmagasiner les informations bancaires du porteur de la carte, et c'est le terminal de paiement qui effectue toutes les opérations pour valider la transaction.

Le 15 mars 1974, Roland Moreno (1945-2012) dépose son premier brevet de carte à mémoire et crée la société Innovatron pour le promouvoir.

Michel Ugon, ingénieur à la compagnie CII Honeywell Bull jette les bases de la carte à microprocesseur en 1977, dont la première sera commercialisée dès 1979. Elle est adoptée par les banques françaises à partir de 1984, faisant considérablement baisser le taux de paiement frauduleux par carte.

Les premières spécifications des cartes bancaires sont établies dès 1983 par le CCETT (Centre Commun d'Études de Télévision et Télécommunications). Il s'agissait d'une signature RSA statique des données de personnalisation de la carte, empêchant un fraudeur de créer une carte avec des informations factices. Trois modules RSA de 321 symboles binaires, soit 97 chiffres décimaux, ont été créés en 1983 : un module opérationnel, un module de test et un module de secours en cas de besoin. La petite histoire raconte que les facteurs du module de secours ont été perdus lors d'un déménagement, le rendant totalement inopérant. Ces modules avaient été donnés pour une durée de vie de cinq à dix ans. Au-delà, la sécurité n'était plus assurée, en raison des progrès prévisibles pour factoriser les entiers.

Dès 1992, des entiers de 97 chiffres étaient factorisés par des équipes de chercheurs. Pourtant, le module opérationnel de 97 chiffres produit en 1983 était toujours en service en 1998, date où a éclaté l'affaire Serge Humpich. Cet informaticien de 35 ans a reconstitué les mécanismes de la carte bancaire en examinant le code de terminaux de paiement d'occasion qu'il s'était procuré. Il a pu factoriser le module RSA à l'aide d'un logiciel japonais disponible sur internet, dévoilant ainsi un secret de fabrication des cartes. Certains paiements frauduleux effectués hors ligne, comme ceux des tickets de métro ou des billets de train, pouvaient être acceptés.

Aujourd'hui, les cartes de paiement sont définies par le consortium EMV (Europay, Mastercard and Visa), auquel se sont joints le japonais JBC en 2004, et l'américain American Express en 2009. Les documents définissant les caractéristiques des cartes sont désormais publics et librement accessibles sur internet.

Les objectifs de sécurité de la carte bancaire sont ainsi définis :

- le terminal de paiement doit s'assurer que la carte bancaire est une véritable carte et non pas une contrefaçon. Cela permet de protéger le vendeur contre les paiements frauduleux ;
- la carte traite les paramètres de gestion des risques, comme le montant maximal des retraits et des paiements ;
- les paiements ont une signature numérique afin d'en assurer l'intégrité ;
- une vérification du porteur de la carte est réalisée pour empêcher les paiements avec des cartes perdues ou volées.

Lors d'une transaction, le terminal de paiement authentifie les données présentes dans la carte et effectue une vérification du porteur. Les terminaux acceptent aujourd'hui deux types d'authentification : l'une dite statique et l'autre dite dynamique.

2.2 L'authentification statique

L'authentification statique consiste en une signature numérique des données présentes dans la carte : numéro de carte, nom et prénom du porteur, date de validité et autres données bancaires. Au moment de la mise en place de cette norme en 1983, les capacités de calcul des cartes ne permettaient pas un calcul des signatures. Les données ainsi que la signature sont introduites dans la carte au moment de sa personnalisation pour un client et, lors de la transaction, les données signées sont transmises au terminal qui peut ainsi vérifier leur authenticité.

L'objectif de cette authentification est seulement de s'assurer que les données de personnalisation de la carte n'ont pas été altérées. Rien n'interdit toutefois de cloner une carte en copiant les mêmes données dans une autre carte.

La signature est assurée par l'algorithme RSA. La taille du module, initialement fixée à 321 symboles binaires, est maintenant augmentée pour tenir compte des progrès effectués en matière de factorisation. Actuellement, la taille dépend de la banque émettrice sans toutefois pouvoir dépasser 1 984 symboles binaires. L'exposant public du RSA peut être égal à 3 ou bien à 65 537 qui vaut $2^{16} + 1$ et conduit à une exponentiation assez rapide.

Les données transmises par la carte au terminal ont préalablement été signées par la banque émettrice de la carte. La signature est accompagnée d'un certificat, élaboré par une autorité de certification, et qui contient :

- la clé publique qui permet la vérification de la signature ;
- la signature de la clé publique réalisée par l'autorité de certification.

Le terminal contient la clé publique de l'autorité de certification. Avec cette clé publique, il commence par vérifier la validité du certificat, puis vérifie la signature des données de la carte à l'aide de la clé publique présente dans le certificat. Ainsi, le terminal ne contient aucune donnée spécifique à une banque émettrice particulière.

Lors d'une authentification statique, le terminal s'assure de l'identité du porteur de la carte en lui demandant son code PIN (*Personal Identification Number*) qu'il transmet en clair à la carte. La carte répond sur la validité de ce code PIN.

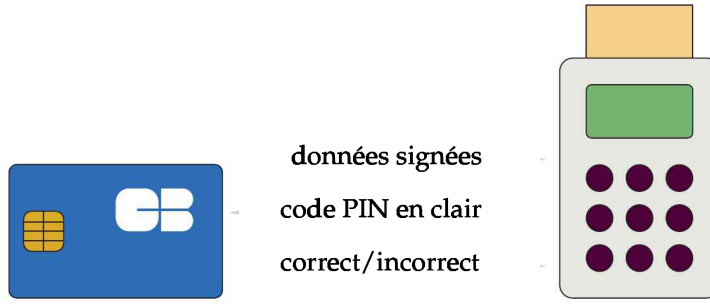


FIGURE 5.2. Carte bancaire : authentification statique. La carte à puce ne fait que transmettre au terminal les données bancaires signées par la banque émettrice. Cela garantit qu'elles sont authentiques et proviennent d'un compte existant.

2.3 L'authentification dynamique

L'authentification statique est considérée comme insuffisamment sûre. La carte n'effectue aucun calcul. Elle ne fait que transmettre des données qui ont été signées une fois pour toutes par la banque au moment de la personnalisation de la carte. Un adversaire peut copier les données bancaires à l'insu du porteur, réaliser une copie qui transmettra au terminal les données bancaires signées et qui répondra systématiquement « *oui* » à toute demande de vérification du code PIN. Il s'agit des *Yes-Cards*.

Les premières générations de carte à puce étaient incapables de réaliser des calculs de signature assez rapidement sans rendre prohibitive la durée de la transaction. Les progrès techniques ont permis d'envisager que les calculs soient effectués dans la carte elle-même et de se prémunir ainsi des contrefaçons.

La carte à puce signe elle-même, avec une clé privée qui lui est propre, les données bancaires à transmettre au terminal. De plus, cette signature dépend d'un nombre imprévisible, différent à chaque transaction et fourni par le terminal, empêchant qu'une fausse carte ne puisse rejouer des données qui auraient été interceptées lors d'une précédente transaction. Afin que le terminal puisse vérifier la signature, les données signées sont accompagnées d'une chaîne de certificats impliquant la banque émettrice et l'autorité de certification.

- Les données bancaires et l'aléa sont signés avec la clé privée individuelle de la carte.
- Ces données sont accompagnées d'un certificat qui contient la clé publique pour vérifier la signature. La clé publique de la carte est signée par la banque émettrice.
- La clé publique de la banque émettrice qui permet de vérifier ce certificat est elle-même signée par l'autorité de certification.
- La clé publique de l'autorité de certification qui sert à vérifier la clé publique de la banque émettrice est présente dans tous les terminaux.

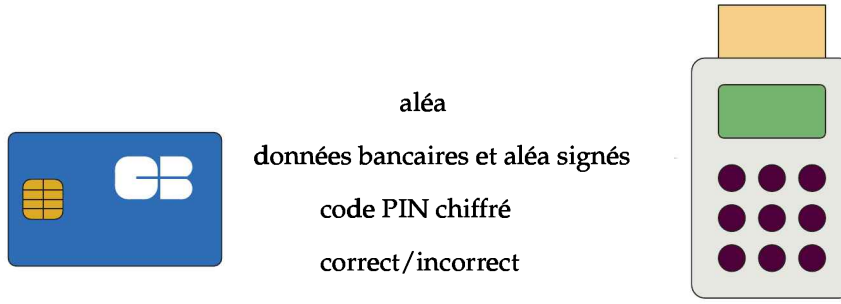


FIGURE 5.2. Carte bancaire : authentification dynamique. La carte à puce signe elle-même les données bancaires en fonction d'une valeur aléatoire fournie par le terminal. Cela authentifie la puce elle-même et empêche le clonage des cartes.

Lors d'une authentification dynamique, le code PIN du porteur de la carte est transmis chiffré à la carte, empêchant qu'un intercepteur ne puisse en prendre connaissance.

2.4 Validation de la transaction

La transaction est validée par la génération d'un *cryptogramme de transaction* produit par la carte, qui consiste à chiffrer les données de la transaction (date, montant, bénéficiaire, etc.) avec une clé unique par transaction, qui est déduite d'une clé secrète présente dans la carte et d'un compteur de transaction incrémenté par la carte à chaque transaction.

La validation de la transaction peut se faire *en ligne* ou *hors ligne*. Lors d'une validation en ligne, le terminal demande à la carte un *cryptogramme de requête de transaction* produit par la carte à partir d'un bloc de données qui inclut le cryptogramme de transaction. Ce cryptogramme de requête de transaction est alors transmis à la banque émettrice qui le vérifie et envoie en retour au terminal un *cryptogramme de réponse d'autorisation*. Le terminal demande à la carte un certificat de transaction qui inclut l'autorisation de la banque. Cet échange avec la banque émettrice est effectué en direct au moment du paiement. Cela peut prendre du temps et provoquer une attente prohibitive dans des lieux de paiements engorgés comme les caisses de supermarché le samedi après-midi. La transaction peut aussi s'effectuer hors ligne. Dans ce cas, le terminal, qui ne peut vérifier le cryptogramme de transaction, le mémorise pour une vérification ultérieure auprès de la banque émettrice.

Le choix d'une validation de la transaction en ligne ou hors ligne est défini par une politique de sécurité qui repose sur :

- une sélection aléatoire du type de validation, la transaction pouvant s'effectuer en ligne quel que soit le montant ou la longueur de la file d'attente à la caisse ;

- une proportion minimale de validations en ligne sur l'ensemble des transactions ;
- une validation en ligne systématique à partir d'un certain montant, ou à partir du montant cumulé des validations hors ligne précédemment effectuées.

2 La sécurité de l'internet

3.1 Contexte historique

Le 4 octobre 1957, l'Union soviétique procède au lancement du premier satellite artificiel, le *Spoutnik*. Cet événement d'une grande portée scientifique et technique provoque un traumatisme aux États-Unis qui créent en réaction, dès 1958, l'*Advanced Research Project Agency* (ARPA) dont le but est de combler le retard américain dans le domaine de la conquête spatiale. Une des missions de l'ARPA sera de développer à partir de 1962 un réseau pour relier les ordinateurs du ministère de la Défense américain entre :

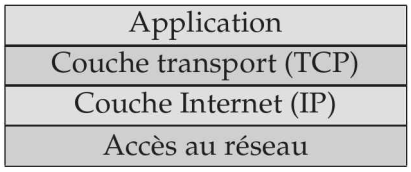
- le Pentagone, quartier général du département de la défense situé en Virginie près de la ville de Washington ;
- Cheyenne Mountain, siège du Crystal Palace situé près des sources du Colorado chargé de la collecte des données provenant de la constellation des satellites américains d'interception ;
- le Strategic Air Command Headquarter (SACHQ), établissement situé à Washington qui contrôle les bombardiers et les missiles de l'arsenal nucléaire américain.

Ce réseau constitue l'embryon de l'ARPANet, réseau de l'ARPA qui connectera également les universités et les fournisseurs de l'armée. Il deviendra internet à partir du milieu des années 1980.

Depuis janvier 1986, les techniques et les standards d'internet sont développés et mis au point par un groupe de travail appelé l'*Internet Engineering Task Force* (IETF), sur la base d'une collaboration volontaire d'ingénieurs du monde entier. L'IETF produit des documents appelés RFC, ce qui signifie *Request For Comments*, c'est-à-dire un appel à commentaires sur des propositions de conception, mais qui sont aujourd'hui davantage les descriptions abouties des standards, publiées après un long processus de révisions et de validations. Tous ces documents décrivent le détail des protocoles en vigueur sur internet, ils sont publics et librement accessibles sur le site www.ietf.org. Depuis 1992, l'*Internet Society*, association internationale à but non lucratif, fournit un cadre légal au processus de standardisation.

3.2 Une organisation du réseau en couches

Un réseau d'ordinateurs est constitué d'un grand nombre de machines connectées entre elles par des liens directs ou indirects. L'information transite d'un ordinateur à l'autre selon un protocole qui doit assurer l'acheminement des données au destinataire. En raison de la diversité des ordinateurs et des liaisons, une architecture en couches s'est imposée. Chaque couche fait appel aux services de la couche de niveau inférieur et fournit des services à la couche de niveau supérieur. La communication entre les couches est définie par un *protocole* qui rassemble les règles et les procédures permettant le dialogue. Les quatre couches standards du réseau internet sont les suivantes :



Sur son ordinateur connecté sur internet, l'utilisateur trouve l'application avec laquelle il interagit. S'il s'agit d'un navigateur, l'application a pour nom http (*HyperText Transfert Protocol*). Il peut aussi s'agir de transfert de fichier (ftp, *File Transfert Protocol*) ou de courrier électronique, pour lequel il existe les applications smtp (*Simple Mail Transfert Protocol*) et pop (*Post Office Protocol*). À l'autre bout, se trouve la couche d'accès au réseau qui est chargée du transfert physique des données par un signal électrique, soit par un câble connecté à l'ordinateur, soit par des ondes électromagnétiques dans le cas d'une liaison sans fil de type WiFi. Entre les deux, se trouvent les couches *Transport* et *Inter-Réseaux*.

La couche Transport

L'application dialogue avec la couche inférieure, appelée *couche Transport* (TCP, *Transport Control Protocol*), et chargée de tous les mécanismes nécessaires à un transport fiable des données. Elle assure la fragmentation et le réassemblage des données en paquets, la réinsertion des paquets manquants, la détection et la correction des erreurs.

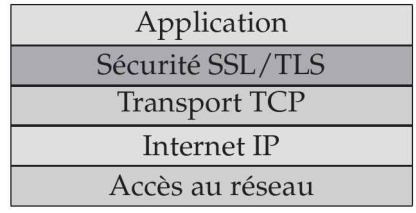
La couche Inter-Réseaux

Le protocole Inter-Réseaux (IP, *Internet Protocol*) a été défini pour pouvoir interconnecter de nombreux réseaux entre eux. Cette couche traite chaque paquet indépendamment des autres. Sa fonction principale est l'acheminement correct au destinataire. Cela repose sur la notion d'adresse, qui est une donnée, à l'origine

de 32 symboles binaires et portée aujourd'hui à 128 symboles binaires. Elle définit la machine destinatrice du paquet. Une partie de l'adresse identifie le réseau et une autre partie de l'adresse identifie la machine dans le réseau.

3.3 La sécurité du réseau

La sécurité dans le réseau internet est assurée par une couche supplémentaire, insérée entre la couche TCP et la couche application. Elle a pour nom SSL qui signifie *Secure Sockets Layer*, couche de sécurisation des connecteurs. Elle a été définie par l'entreprise Netscape à partir de 1994 et reprise par l'IETF sous le nom TLS (*Transport Layer Security*) suite au rachat des brevets à Netscape en 2001. L'utilisateur peut observer que cette couche est active en remarquant un « *s* » supplémentaire dans la dénomination son application : https, ftps, pops, etc. Certains navigateurs affichent également une clé ou un cadenas pour indiquer que la transaction en cours est protégée. Cette sécurité contribue en particulier à la confiance dans le commerce sur internet, par la protection des transactions, la dissimulation des données bancaires et la confidentialité des achats.



La couche SSL/TLS authentifie le correspondant et assure la confidentialité et l'intégrité des données échangées. Ces services de sécurité reposent sur des algorithmes cryptographiques symétriques et asymétriques qui sont négociés à partir d'un protocole appelé *Handshake Protocol* (poignée de main). Une fois que sont convenus entre les correspondants les processus et les algorithmes, les données sont encapsulées dans des enregistrements selon un protocole appelé *Record Protocol*, chargé du chiffrement et de la signature des données.

La poignée de main

Ce protocole initie une transaction sécurisée entre la machine de l'utilisateur, appelée « client », et le serveur auquel il se connecte. Il s'agit de se mettre d'accord sur la version SSL/TLS applicable et sur les algorithmes cryptographiques qui sont acceptés de part et d'autre. Pour cela, le client transfère la plus haute version SSL/TLS avec laquelle il peut travailler ainsi qu'une liste d'algorithmes cryptographiques utilisables dans un ordre préférentiel. Le serveur répond avec

le même type d'informations le concernant. Après cet échange, une procédure commune aux deux machines leur permet de convenir des algorithmes cryptographiques qui seront utilisés. Le serveur transmet alors sa clé publique incluse dans un certificat, ce qui permet au client de l'authentifier.

Ensuite, le client génère une clé initiale appelée *Master Secret*, qu'il transmet au serveur, protégée par sa clé publique. Cette clé sert à déduire quatre autres clés symétriques qui seront produites à l'identique, mais séparément par le client et le serveur selon un algorithme convenu. Ces quatre clés ne circulent pas sur le réseau. Leur fonction est d'assurer la confidentialité et l'authentification des données échangées :

- *Server write mac secret* : utilisée pour authentifier les messages du serveur ;
- *Client write mac secret* : utilisée pour authentifier les messages du client ;
- *Server write key* : utilisée pour chiffrer les données émises par le serveur ;
- *Client write key* : utilisée pour chiffrer les données émises par le client.

Échange des données protégées

Les données transmises sont encapsulées de manière à pouvoir être traitées de manière homogène. Si un bloc de données est trop long, il est fragmenté en plusieurs blocs de taille inférieure à 16 384 (2^{14}) symboles binaires. Puis, un algorithme de compression est appliqué pour éventuellement réduire la taille des données transmises. De manière optionnelle, un code d'authentification est adjoint afin de les authentifier et d'assurer leur intégrité. Elles sont ensuite chiffrées. Selon l'algorithme de chiffrement choisi, il peut être nécessaire d'adjoindre des données supplémentaires, par exemple pour que la taille soit multiple de 64 ou 128 symboles binaires. Le tout est encapsulé avec des données d'en-tête comprenant des informations sur le traitement subi afin de permettre au destinataire de reconstituer l'ensemble des données en clair.

4 La cryptologie dans la téléphonie mobile

4.1 Historique et principes généraux

En 1982, la CEPT (Conférence Européenne des Postes et Télécommunications) constitue le *groupe spécial mobile* pour définir un standard européen pour la téléphonie mobile. Le travail de ce groupe aboutira à la définition de la norme GSM (*Global System for Mobile communications*), dite de deuxième génération. La mobilité introduite dans le réseau téléphonique nécessite de nouvelles fonctionnalités, comme la localisation de l'abonné. De plus, comme la communication sans fil utilise les ondes électromagnétiques, la question de la sécurité devient

cruciale pour assurer la discrétion des échanges et empêcher un accès frauduleux au réseau au détriment d'un abonné régulier.

La norme GSM a évolué une première fois vers le système GPRS (*General Packet Radio Service*), caractérisé par un débit plus élevé et la communication par paquets, mieux adaptée à la transmission de données. Le débit a encore augmenté et la sécurité a été renforcée dans la téléphonie mobile dite *de troisième génération* UMTS (*Universal Mobile Telecommunication System*), spécifiée dès 1999 et déployée pour la première fois en 2002 par l'opérateur norvégien Telenor, tout en assurant la cohabitation et la compatibilité avec la norme GSM. Le réseau UMTS utilise des algorithmes cryptographiques qui lui sont propres.

Dans le téléphone mobile, la sécurité est assurée par une carte à puce, appelée carte SIM (*Suscriber Identity Module*) dans le réseau GSM et carte USIM dans le réseau UMTS. Cette carte contient les données de l'abonné et en particulier une clé secrète appelée K_i, partagée avec le centre d'authentification. Toute la sécurité repose sur cette clé K_i.

Les mesures de sécurité mises en place assurent l'authentification de chaque abonné avant de lui autoriser l'accès au service. L'abonné est identifié par une identité temporaire protégée, destinée à empêcher le piratage et à dissimuler sa véritable identité. La communication est également chiffrée pour interdire son écoute. Dans le réseau UMTS, l'authentification est mutuelle, ce qui signifie que l'abonné authentifie aussi le réseau, afin de ne pas répondre aux fausses stations de bases qui tenteraient d'obtenir frauduleusement des informations de connexion.

4.2 Fonctions cryptographiques de l'UMTS

La sécurité du réseau UMTS repose sur huit fonctions cryptographiques, que la norme UMTS note f_0 à f_5, f_8 et f_9.

- f_0 est une fonction de génération de données aléatoires. Elle est utilisée par le centre d'authentification pour produire un nombre aléatoire RAND (*random*) qui servira de paramètre aux autres fonctions d'authentification.
- f_1 est la fonction d'authentification du réseau. Elle produit un code MAC-A qui permettra à la carte USIM de l'abonné d'authentifier le réseau.
- f_2 est la fonction d'authentification de l'utilisateur. Elle produit un code XRES (*eXpected RESponse*) qui permettra au réseau d'authentifier l'abonné.
- f_3 est la fonction qui va calculer la clé de chiffrement CK (*Cipher Key*) avec laquelle les communications seront chiffrées.
- f_4 est la fonction qui calcule la clé d'intégrité IK (*Integrity Key*) avec laquelle sera contrôlée l'intégrité des communications.

f_5 est la fonction qui calcule une clé d'anonymat AK (*Anonymity Key*). Cette clé est utilisée lorsque la séquence identifiant la communication doit être dissimulée afin d'en assurer l'anonymat.

f_8 est la fonction de chiffrement des communications entre le téléphone et la station de base pour en assurer la confidentialité. Cette fonction est paramétrée par la clé de chiffrement CK.

f_9 est la fonction de contrôle d'intégrité des communications entre le téléphone et la station de base. Cette fonction est paramétrée par la clé d'intégrité IK.

Les fonctions f_1 à f_5 sont présentes à la fois dans la carte USIM et dans le centre d'authentification. Ces fonctions cryptographiques sont paramétrées par la clé secrète K_i partagée par le centre d'authentification et la carte USIM de l'abonné. Elles servent à l'authentification de l'abonné et du réseau, ainsi qu'à l'établissement de clés de session CK et IK qui serviront à chiffrer et à contrôler l'intégrité des communications à l'aide des fonctions f_8 et f_9. Ces fonctions sont présentes dans le téléphone de l'abonné et dans la station de base. Elles sont construites à partir d'un algorithme cryptographique par blocs de 128 symboles binaires appelé *Kasumi*, qui est un schéma de Feistel comprenant huit itérations.

Lors d'une demande d'accès au réseau, le centre d'authentification envoie au mobile un nombre aléatoire RAND issu de la fonction f_0, une séquence SQN qui identifiera temporairement la communication et un code d'authentification donné par la formule MAC-A $= f_1(\text{RAND,SQN})$. La carte USIM vérifie alors la validité de MAC-A, ce qui lui assure que la borne avec laquelle elle communique n'est pas une borne pirate. Optionnellement, la séquence SQN peut être dissimulée. Pour cela, elle est masquée avec la clé d'anonymat AK calculée par la formule AK $= f_5(\text{RAND})$. La donnée échangée est alors SQN $\oplus$ AK. L'authentification de l'abonné est réalisée par le calcul, avec la fonction f_2, de la réponse attendue XRES, par la formule XRES $= f_2(\text{RAND})$. Cette réponse est transmise au centre d'authentification qui la contrôle avant d'autoriser l'accès au réseau.

Les clés de chiffrement et d'intégrité sont calculées par les formules CK $= f_3(\text{RAND})$ et IK $= f_4(\text{RAND})$. Les communications sont alors chiffrées et authentifiées avec les fonctions f_8 et f_9, dont les paramètres sont les clés de sessions CK et IK.

5 La télévision à péage

5.1 Historique

En France, le monopole de l'État sur les radiodiffusions a été aménagé par un décret en date du 20 mars 1978, ouvrant la voie à la télévision à péage et à ce qu'on appelle « *l'accès conditionnel aux services audiovisuels* ». Ces nouveaux

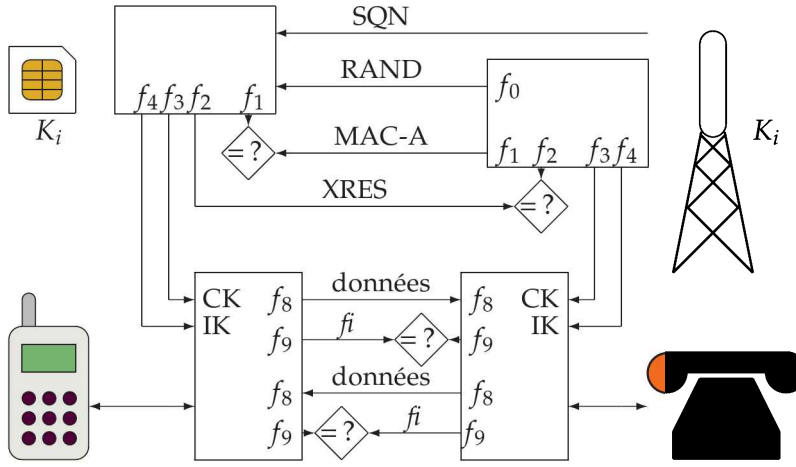


FIGURE 5.4. La carte USIM et le centre de diffusion partagent une clé K_i propre à chaque abonné. Lors d'une demande de connexion au réseau, le centre d'authentification transmet des données RAND, SQN et MAC-A que la carte USIM peut contrôler avec K_i. En retour, elle renvoie une réponse XRES assurant l'authenticité de l'abonné. Ces données permettent également de calculer une clé de chiffrement CK et une clé d'intégrité IK qui serviront à la protection des données émises et reçues par voie radio. Ces clés sont transmises au téléphone par la carte USIM afin que celui-ci puisse chiffrer le signal et l'assortir d'une figure *fi* de contrôle d'intégrité à l'émission, ainsi que déchiffrer le signal et en contrôler l'intégrité à la réception.

services sont liés au développement du cinéma, de la télévision et de la musique sur des supports, magnétiques ou numériques, devenus accessibles à tous.

À l'origine, en France, la télévision était entièrement financée par la redevance. Les premières publicités sont apparues en 1968. Le 4 novembre 1984, a eu lieu la première émission de Canal Plus, cryptée avec le système d'embrouillage *Syster* (Système Terrestre), inaugurant un nouveau modèle économique pour la télévision. Elle devient un service payé par l'utilisateur.

Encadré 5.1. Embrouillage, désembrouillage, cryptage, décryptage.

Les termes *embrouillage* et *désembrouillage* sont les traductions légales des mots anglais *scrambling* et *descrambling*, données dans le *journal officiel* numéro 290 du 14 décembre 2004. Selon ce texte, l'embrouillage désigne la « *transformation réversible d'un signal numérique, en vue d'en faciliter la transmission ou l'enregistrement, en un signal numérique de même signification et de même débit binaire* ».

Les termes *cryptage* et *décryptage* sont des anglicismes incorrects qui viennent de *encrypt* et de *decrypt*. Ils sont en usage principalement dans le contexte de la télévision numérique à péage à la place des termes corrects *chiffrement* et *déchiffrement*.

En 1989, est lancé en France un projet de télévision à haute définition qui ne sera jamais déployé, mais qui sera l'occasion de la conception par le CCETT

du système *Eurocrypt* d'accès conditionnel, utilisé par la suite par Canal Satellite et TPS (*Télévision Par Satellite*). Outre l'abonnement, ce système introduit de nouveaux services, comme le paiement à la séance par commande ou achat impulsif. Ces nouveaux services ont convergé vers la *Télévision Numérique Interactive*, autour d'une norme promue par le consortium européen DVB (*Digital Video Broadcasting*), rendue possible par l'amélioration considérable des techniques de compression numérique d'image en mouvement avec les normes MPEG2, puis MPEG4. Aujourd'hui, la télévision est diffusée sur une grande variété de support : le satellite, le câble, la télévision numérique terrestre (TNT), l'ADSL, les téléphones mobiles, etc.

5.2 L'accès conditionnel

Un système d'accès conditionnel a pour objectif de réserver l'accès aux *contenus multimédia* à ceux qui ont acquitté un droit d'accès et d'en exclure les autres. Les clients ne doivent pas subir les contraintes de la sécurisation qui protège les intérêts des fournisseurs de contenu. L'accès conditionnel doit être transparent. Le client paye et ne doit pas être gêné par la protection mise en place. La première priorité d'un système d'accès conditionnel est le bon fonctionnement pour l'abonné régulier. Ensuite vient la sécurité pour en interdire l'accès à ceux qui n'ont pas payé.

En raison des montants mis en jeu dans ce domaine – rappelons que les droits de retransmission des rencontres de football s'élèvent à plusieurs centaines de millions d'euros par an –, ces systèmes sont très attaqués. Dans le courant des années 1990, ils ont subi de nombreux assauts de pirates, qui ont réussi à réaliser de fausses cartes donnant accès aux programmes.

La problématique de l'accès conditionnel est très spécifique et sort du cadre usuel de la communication confidentielle. Il est conçu pour protéger le modèle économique de l'opérateur de télévision. Et pour cela, la clé est diffusée à tous les abonnés qui sont autant de pirates potentiels. Le même cryptogramme est diffusé à tous (*broadcast*) de manière unidirectionnelle, depuis le centre de diffusion vers les abonnés. Le message transmis n'est pas secret, qu'il s'agisse d'un film ou d'une rencontre sportive, et dans ce dernier cas, sa valeur marchande est de très courte durée, limitée à la diffusion en direct.

5.3 Description

Le système d'accès conditionnel est l'ensemble des moyens techniques mis en œuvre pour satisfaire les objectifs énoncés dans le paragraphe précédent, depuis la prise d'abonnement jusqu'à sa résiliation. Il régit la relation entre deux acteurs :

l'abonné, à qui on fournit un décodeur et une carte à puce, et le fournisseur de contenu, qui embrouille le programme au niveau du centre de diffusion. Il est défini en Europe, en Asie et en Océanie par une norme DVB sur la diffusion de la télévision numérique.

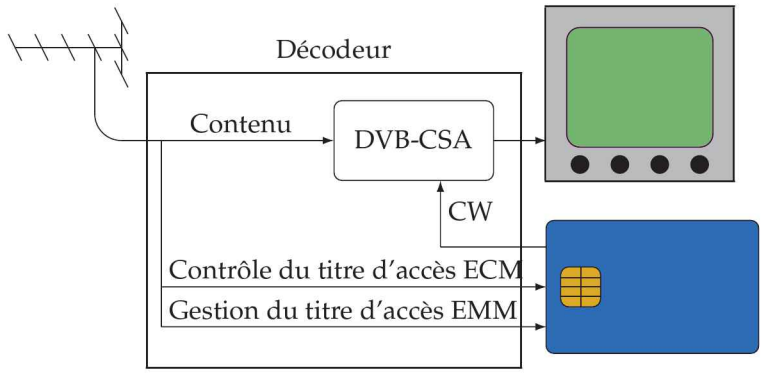


FIGURE 5.4. Le décodeur reçoit un signal composé de plusieurs informations : le contenu audiovisuel chiffré avec le mot de contrôle CW (*Control Word*), les messages de contrôle des titres d'accès ECM et les messages de gestion des titres d'accès EMM. Les messages ECM et EMM sont transmis à la carte à puce qui, en fonction des titres d'accès, renvoie ou non le mot de contrôle en clair au décodeur en vue du déchiffrement du contenu par l'algorithme DVB-CSA.

Le signal radiodiffusé comprend trois composantes : la vidéo, une ou plusieurs pistes audio pour la version originale, la version française ou en une autre langue, et le texte pour les sous-titres. Le décodeur est chargé de la transformation du signal depuis la réception par l'antenne jusqu'aux signaux vidéo et audio destinés au téléviseur.

En plus du signal porteur de contenu, il existe deux types de messages uniquement destinés au traitement des droits d'accès :

- les *messages de contrôle du titre d'accès* (*Entitlement Control Message*, ECM) contiennent la clé d'embrouillage du signal ;
- les *messages de gestion du titre d'accès* (*Entitlement Management Message*, EMM) permettent de communiquer les informations d'accès au service et toute autre information utile au système, comme la mise à jour du logiciel du décodeur ou des applications interactives.

5.4 L'embrouillage du signal

L'embrouillage utilise un algorithme standard appelé DVB-CSA (*Common Scrambling Algorithm*), composé d'un chiffrement par bloc opérant sur des blocs de 64 symboles binaires et d'un chiffrement à flot. Il est commun à tous les opérateurs

et à tous les programmes qu'ils soient cryptés ou en clair. La clé d'embrouillage, appelée le *mot de contrôle*, est constante pour les programmes en clair. Pour les programmes cryptés, elle change toutes les dix secondes, et est transportée chiffrée dans les ECM. Ces ECM sont transmis quatre fois par seconde afin de limiter le temps d'attente de la réception en clair lorsqu'un abonné vient de brancher son téléviseur. L'algorithme de chiffrement des ECM et des EMM est propre au fournisseur du système d'accès conditionnel.

5.5 Hiérarchie des clés

L'architecture d'un système d'accès conditionnel se décrit par une structure hiérarchique de clés et des messages qui véhiculent ces clés ou des informations de service.

- Le *mot de contrôle* embrouille le signal vidéo. Il est renouvelé toutes les dix secondes environ et est transmis chiffré dans un ECM.
- La *clé d'exploitation* permet de déchiffrer les ECM. Elle est transmise chiffrée dans un EMM tous les mois.
- Les clés qui permettent de déchiffrer les EMM contenant les clés d'exploitation sont les clés de gestion des abonnés.

Pour limiter la bande passante requise pour le renouvellement mensuel des clés d'exploitation, les abonnés sont rassemblés par groupes ayant une clé de gestion commune. La clé d'un groupe n'est renouvelée qu'en cas de compromission. Dans ce cas, chaque abonné reçoit individuellement une nouvelle clé de groupe dans un EMM, chiffrée avec sa clé individuelle. La clé individuelle est unique et propre à un abonné. Elle est inscrite dans sa carte au moment de la souscription de l'abonnement. Elle dure jusqu'à sa résiliation.

Lors de la souscription, l'abonné reçoit une carte à puce qui lui permettra l'accès aux services. Elle permet de stocker les secrets que sont la clé individuelle, la clé de gestion de groupe et la clé d'exploitation du mois. La sécurité du système d'accès conditionnel repose pour une grande part sur l'impossibilité d'extraire ces éléments secrets de ces cartes à puce par une attaque physique ou logique.

5.6 Le décryptage du programme

Le décodeur reçoit un signal vidéo et audio crypté. Il contient l'algorithme DVB-CSA, mais il a besoin du mot de contrôle pour réaliser le déchiffrement et fournir un signal visible et audible au téléviseur. Le mot de contrôle est transmis dans les ECM qui sont eux-même chiffrés. Pour procéder au décryptage du programme, le décodeur transmet l'ECM à la carte à puce présente dans le décodeur.

À la réception de cet ECM, le logiciel de la carte à puce vérifie s'il dispose des critères qui lui permettent de le déchiffrer afin de transmettre le mot de

contrôle au décodeur. Ces critères comprennent la clé d'exploitation avec laquelle l'ECM est chiffré ainsi que le titre d'accès, constitué des informations sur le type d'abonnement souscrit par l'abonné, sa date de validité, etc. Si tous les critères sont remplis, alors la carte à puce procède au déchiffrement de l'ECM et transmet le mot de contrôle au décodeur qui peut ainsi rendre clair le programme crypté. Dans le cas contraire, par exemple si le programme ne fait pas partie de l'abonnement souscrit, la carte ne renvoie rien. Si le décodeur ne reçoit pas le mot de contrôle, il fournit au téléviseur le signal d'un écran noir et muet.

6

La théorie cryptologique

On voit se développer aujourd'hui une théorie cryptologique dont l'approche scientifique contraste avec l'art exercé par les premiers artisans du chiffre. Ce chapitre, dont la première lecture peut sembler rude, présente les éléments de cette nouvelle discipline.

1 Motivation

Dans l'approche classique du développement cryptographique, le concepteur propose un procédé. Le cryptanalyste cherche à le pénétrer. Lorsqu'une faille est décelée, elle est corrigée, un nouveau procédé est proposé qui est à nouveau soumis à l'attaque, et ainsi de suite. Finalement, la communauté cryptologique accepte la sécurité du schéma lorsqu'aucune faiblesse n'a été mise en évidence par les travaux des cryptanalystes pendant un certain temps. Cette approche a montré ses limites. Au bout de combien de temps la solidité doit-elle être acceptée ? trois ans ? cinq ans ? dix ans ?

Le chiffre de César a, selon Aulu Gelle, été résolu par le grammairien Valerius Probus dès le I^{er} siècle. Le polyalphabétisme a résisté longtemps avant de succomber aux attaques de Babbage et Kasisky à la fin du XIX^{e} siècle. La machine ENIGMA est tombée sous les coups des cryptanalystes polonais et anglais pendant le second conflit mondial. Le mode d'utilisation ECB des chiffreurs par blocs, pourtant normalisé, n'est plus considéré comme sûr. Plus récemment, un système à clé publique reposant sur le problème du sac à dos (voir encadré 6.1), proposé par Benny Chor et Ronald Rivest en 1988, a été cassé dix ans plus tard par Serge Vaudenay. En novembre 1993, le standard de chiffrement PKCS#1 est publié par l'entreprise RSA laboratories. Cinq ans plus tard, Daniel Bleichenbacher décrit une façon de décrypter un message de son choix par une attaque à cryptogramme choisi.

Ces exemples ont poussé les cryptographes à rechercher des arguments pour attester de la sécurité des schémas de chiffrement qu'ils construisent dès leur conception. Aujourd'hui, toute nouvelle proposition doit être accompagnée de tels arguments. Cette approche a elle-même ses limites, car des schémas prouvés sûrs ont eux aussi été cassés. Le système cryptographique de Ajtai-Dwork, publié en 1987, bien qu'il soit assorti d'une preuve de sécurité, a été cryptanalysé un an plus tard par Phong Nguyen et Jacques Stern. L'adversaire d'un système réel n'obéit pas forcément à la modélisation que les théoriciens de la cryptologie ont faite de lui.

Encadré 6.1. Le problème du sac à dos.

Le problème du sac à dos s'énonce ainsi : « *Étant donné une liste d'entiers, comment choisir certains d'entre eux pour que leur somme soit égale à une valeur ciblée à l'avance ?* » C'est aussi le problème du marchand de pommes qui doit choisir les fruits sur son étalage pour satisfaire le client qui lui en demande un poids précis. Dans certains cas, ce problème est facile à résoudre, par exemple si la liste d'entiers est constituée des puissances de deux : 1, 2, 4, 8, 16, 32. Pour atteindre la valeur 43, sa numération binaire, ici égale à 101011, fournit directement la solution :

$$43 = 32 + 8 + 2 + 1$$

Le choix des unités de pièces et des billets de banque est aussi conçu pour pouvoir sans trop de difficulté résoudre ce problème et *faire la monnaie* pour n'importe quelle somme. Mais si la liste d'entiers est quelconque, le problème du sac à dos est difficile et nécessite pratiquement d'explorer tous les choix possibles pour trouver une solution. Cette difficulté a inspiré les cryptologues pour tenter de concevoir des mécanismes de chiffrement. La clé publique est une liste d'entiers apparemment quelconque, par exemple :

$$(17, 34, 21, 42, 37, 27)$$

Si le message à chiffrer est 101011, le cryptogramme sera la somme des entiers de la liste qui correspondent à un 1. Ici, ce sera :

$$17 + 21 + 37 + 27 = 102$$

Si le destinataire sait résoudre le problème du sac à dos, il saura comment choisir les entiers de la liste pour atteindre la valeur 102, et ainsi reconstituer le message en clair correspondant.

Les méthodes proposées pour que la résolution du problème du sac à dos soit conditionnée par la connaissance d'une trappe secrète prennent comme point de départ une instance facile à résoudre, comme la liste des puissances de deux, et en masquent la structure par une fonction secrète compatible avec l'addition, comme une fonction linéaire. Malheureusement, les propositions pour exploiter ce problème en cryptographie ont toutes été cryptanalysées les unes après les autres et les chercheurs ont finalement abandonné toute recherche dans cette direction.

La théorie de la cryptologie est une discipline nouvelle en pleine expansion aujourd'hui, qui a dépassé l'art de concevoir des mécanismes de dissimulation et d'authentification sûrs, mais qui s'attache à construire un édifice permettant d'élaborer, à partir de briques de base, des mécanismes assortis de preuves de sécurité. La preuve s'appuie sur une modélisation de l'adversaire. Cette théorie définit des jeux où l'objectif de l'adversaire est de trouver une information cachée qui attesterait de la faiblesse de la construction, et où son avantage est exprimé en fonction de sa probabilité de gagner la partie.

Les premiers éléments d'une théorie cryptographique ont été posés par Claude Shannon dès 1949, dans un article *Théorie de la communication des systèmes de confidentialité*, à partir de travaux menés pendant la deuxième guerre mondiale. Shannon a en effet participé à l'un des projets les plus secrets, qui n'a été révélé qu'en 1975, le « *projet X* ». Il s'agissait de créer et de développer un système de transmission chiffrée de la voix, destiné à une liaison téléphonique entre le président américain et le Premier ministre britannique qui ne puisse être décrypté à aucun prix. Ce projet a aussi été connu sous le nom de « *Sigsaly* » ou « *Ciphony One* ». Comme aucun traitement analogique de la voix ne pouvait satisfaire le niveau de sécurité requis, ce projet a élaboré un traitement numérique du signal vocal, initiant ce mouvement de numérisation qui s'est prolongé jusqu'à aujourd'hui, et qui est connu sous le nom de *révolution numérique*.

Les travaux de Shannon ont tout d'abord donné naissance à des résultats sur la sécurité inconditionnelle, qui est celle du système idéal où le cryptogramme ne dévoile absolument rien du texte dont il est issu. L'avènement des clés publiques a obligé les cryptologues à affaiblir l'exigence de sécurité idéale attendue de ce type de cryptographie, et à se contenter d'une sécurité calculatoire, mesurée à l'aune des moyens de calcul de l'adversaire.

2 La sécurité inconditionnelle

Shannon est le premier à avoir fait un usage systématique et général de la théorie des probabilités dans le domaine de la communication, jusqu'à la définir comme un processus relevant du hasard.

> *Les aspects sémantiques de la communication ne sont pas pertinents pour l'ingénieur. L'important est que le message réel est choisi dans un ensemble de messages possibles.*

La langue écrite elle-même est modélisée comme une succession plus ou moins imprévisible de caractères.

Jusqu'à ce qu'il prenne connaissance du message, le destinataire est dans l'ignorance de son contenu. L'ouverture et la lecture d'une lettre peuvent être vues

comme la réalisation d'une expérience au résultat aléatoire, au même titre que le lancé d'un dé ou d'une pièce. La lecture révèle le texte élaboré par l'émetteur du message. La théorie de l'information, développée dans le cadre des systèmes de communication, vise à étudier et à quantifier l'incertitude dans laquelle se trouve le correspondant sur le point de recevoir une nouvelle. Quelle quantité d'information sa lecture a-t-elle révélée ?

Appliqué aux systèmes de confidentialité, cela conduit à postuler que le cryptogramme ne doit rien révéler du message en clair, et doit laisser l'adversaire face à son incertitude sur son contenu. En d'autres termes, le cryptogramme ne doit présenter aucune signification pour qui ne dispose pas du droit d'accéder au contenu, c'est-à-dire à qui n'est pas en possession de la clé de déchiffrement.

2.1 Information

Au cœur de la théorie de l'information, se trouve la notion de probabilité qui évalue la vraisemblance d'un événement incertain par une quantité graduée entre 0 et 1. Cette évaluation repose sur une connaissance *a priori* des phénomènes observés. Par exemple, en lançant un dé à six faces, on dispose de trop peu de données pour déterminer la face qui sera visible lorsqu'il se stabilisera. La symétrie du problème interdit de privilégier un résultat particulier, ce qui conduit à attribuer la probabilité $1/6$ à chaque possibilité.

Au contraire de la probabilité, l'information est une quantité associée à un événement une fois qu'il a eu lieu. Anton, 1 800 points Elo, a plus de chances de gagner contre Alexandre, 1 500 points Elo, lors de la partie qu'ils disputeront au prochain tournoi d'échecs. Si Anton gagne, ce qui était prévisible, il est peu probable qu'on en parle. Si, au contraire, Alexandre réussit à s'imposer, sa performance surprenante fera sans doute la une des gazettes. Moins un événement est probable, et plus sa survenue apporte d'information. Appliqué aux communications, cela revient à exprimer qu'un message inattendu sera porteur de plus d'informations qu'une nouvelle anodine.

Il convient aussi de postuler que l'information apportée par deux événements indépendants est la somme des informations apportées par chacun d'eux pris seul. Une fois cela posé, la définition de l'information s'impose. Comme la probabilité conjointe de deux événements indépendants est le produit de leurs probabilités et qu'on veut que, dans ce cas, les informations s'ajoutent, la définition de l'information reposera sur une fonction qui transforme les produits en sommes. Cette fonction est bien connue des mathématiciens, il s'agit de la fonction logarithme. L'information apportée par un événement doit être l'opposé du logarithme de sa probabilité :

$$I(E) = -\log\big(p(E)\big)$$

Le logarithme en base 2 est d'un usage courant. Il définit l'unité d'information, le *bit* qui vient de l'anglais *BInary digiT*.

2.2 L'entropie de Shannon

L'entropie est une quantité associée à une expérience qui quantifie l'incertitude de son résultat. Prenons par exemple l'expérience qui consiste à observer si la première personne rencontrée est un droitier ou un gaucher. Étant donné la proportion de gauchers dans la population, environ égale à 13 %, le résultat est prévisible avec finalement assez peu de chance de se tromper. Il n'en est pas de même s'il s'agit de déterminer si la première personne rencontrée est une femme ou un homme. Dans ce cas, l'incertitude est complète et l'observation de la personne apportera une information maximale.

L'entropie de Shannon d'une expérience, définie par une distribution de probabilités, est par définition la moyenne des informations de tous les résultats possibles, pondérée par leur probabilité. Si une expérience a deux résultats possibles, l'un avec la probabilité p et donc l'autre avec la probabilité $1 - p$, alors son entropie est :

$$H = -p\log(p) - (1-p)\log(1-p)$$

Cette valeur est nulle lorsque p vaut 0 ou 1 et est maximale pour $p = 1/2$.

2.3 Les autres entropies

L'entropie de Shannon représente le nombre minimal de questions avec réponses binaires *oui/non* qu'il faut poser pour avoir la connaissance d'un résultat aléatoire. Elle donne le nombre minimal de symboles binaires requis pour coder une information. En ce sens, elle est adaptée au contexte de la compression des données. Elle n'est cependant pas la seule mesure possible pour évaluer l'incertitude d'une expérience.

L'entropie de Rényi

En 1961, le mathématicien hongrois Alfréd Rényi a proposé d'autres entropies, dont l'une repose sur la probabilité de collision, qui est par définition la probabilité de répétition d'un même résultat lorsque l'on réalise deux fois une même expérience au résultat aléatoire. Ce qui est connu sous le nom d'*entropie de Rényi* est l'opposé du logarithme de la probabilité de collision P_c de l'expérience :

$$R = -\log_2(P_c)$$

L'importance de la probabilité de collision avait déjà été soulignée par William Friedman avec l'introduction de l'indice de coïncidence (voir encadré 4.3 page 92). La théorie de l'information a confirmé son importance en mettant en avant la façon dont cette notion est représentative de l'information portée par les lettres d'un texte.

La min-entropie

L'entropie éclaire aussi sur la qualité d'une source de symboles aléatoires. La qualité attendue d'une telle source est l'incertitude totale dans laquelle se trouve un observateur face à elle. Tout symbole produit doit être inattendu, et tout joueur pariant sur le prochain symbole ne doit espérer aucun gain. La production d'aléa est essentielle en cryptographie pour générer des clés, des nombres premiers aléatoires ou les défis des protocoles d'authentification. Une source aléatoire binaire parfaite produit des séquences de n symboles binaires dont l'entropie est égale à n bits. En pratique, les sources d'aléa sont loin d'atteindre cet idéal. Elles sont imparfaites, en partie prévisibles, les symboles produits ne sont pas indépendants et finalement réalisent des variables aléatoires dont l'entropie de Shannon est inférieure au nombre de symboles binaires générés. Un dispositif convenable de production d'aléa est constitué d'une source malheureusement imparfaite, complétée d'un extracteur, c'est-à-dire une fonction de compression, réduisant le nombre de symboles, de telle sorte que les symboles sortant de l'extracteur ont une entropie qui approche l'idéal de un bit d'entropie par symbole binaire produit.

L'entropie, appelée *min-entropie*, représente le nombre maximum de symboles binaires, porteurs d'une incertitude maximale, qu'il est possible d'extraire d'une source imparfaite. Elle est l'information minimale portée par un symbole produit par la source :

$$H_\infty = \min_x\left(-\log_2(p_x)\right),$$

où p_x désigne la probabilité que la source produise le symbole x.

Les différentes notions d'entropie décrivent des situations différentes d'incertitude. Elles sont toutefois reliées par les inégalités suivantes :

$$\frac{R}{2} \leq H_\infty \leq R \leq H \tag{6.1}$$

2.4 L'entropie conditionnelle

L'entropie conditionnelle, appelée aussi *incertitude résiduelle*, quantifie l'incertitude sur la réalisation d'une expérience aléatoire qui est levée par la connaissance

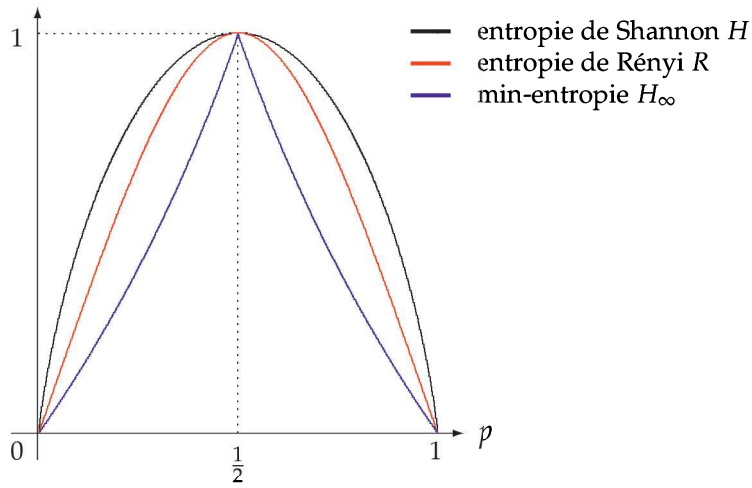


FIGURE 5.4. Sur ce graphique, sont représentées les trois courbes d'entropie binaire, pour une expérience à deux résultats, l'un avec une probabilité p et l'autre avec une probabilité $1-p$. En noir, figure l'entropie binaire de Shannon, en rouge, l'entropie binaire de Rényi, et en bleu la min-entropie. Dans les trois cas, l'entropie est maximale lorsque le résultat est le plus incertain. La position relative des courbes illustre l'inégalité 6.1.

du résultat d'une autre expérience. Supposons par exemple qu'une rencontre sportive oppose ce soir deux équipes A et B. Le résultat de la rencontre est incertain, et l'entropie de la rencontre évalue cette incertitude. Mais si on sait que la veille, l'équipe A a battu l'équipe C, qui elle-même a battu l'équipe B, alors cela apporte une certaine information sur le résultat probable de la rencontre de ce soir, information qui est plutôt en défaveur de l'équipe B.

Plus généralement, l'incertitude du résultat de la rencontre des équipes A et B sera partiellement levée si on connaît les résultats de A contre C et de C contre B. C'est ce qu'on appelle l'incertitude résiduelle.

Encadré 6.2. Expression de l'incertitude résiduelle.

Considérons deux expériences X et Y. Les résultats de la première expérience sont des éléments d'un ensemble E et p_x est la probabilité d'avoir x comme résultat. De même les résultats de la deuxième expérience sont des éléments d'un ensemble F et p_y est la probabilité d'avoir y pour résultat. Lorsqu'on observe les deux résultats, p_{xy} désigne la probabilité conjointe que le résultat de l'expérience X soit x et que celui de l'expérience Y soit y. La quantité $p_{x|y}$ désigne la probabilité que l'expérience X ait pour résultat x sachant que l'expérience Y a pour résultat y. Le calcul des probabilités nous enseigne que $p_{x|y} = p_{xy}/p_y$. L'incertitude résiduelle de Shannon sur l'expérience X lorsque le résultat de l'expérience Y est noté $H(X \mid Y)$ est donnée par :

$$H(X \mid Y) = -\sum_{\substack{x\in E\\ y\in F}} p_{xy} \log_2(p_{x|y})$$

L'incertitude résiduelle évalue l'information révélée par la réalisation d'une expérience aléatoire lorsque le résultat d'une autre expérience est connue. Si les deux expériences sont indépendantes, la connaissance du résultat de l'une ne changera rien sur l'incertitude de l'autre mais, souvent, une information supplémentaire réduira l'incertitude.

2.5 Le chiffrement parfait

Le schéma du système de confidentialité de la figure 2.1 page 24 montre trois acteurs : l'émetteur, le destinataire et l'adversaire, ainsi que trois données : le message, le cryptogramme et la clé. Les trois acteurs sont dans des situations d'incertitudes différentes relativement à ces données. Pour l'émetteur, tout est connu. Pour le destinataire, la clé est connue, le cryptogramme aussi et l'incertitude sur le message sera levée dès qu'il aura procédé au déchiffrement. L'adversaire, lui, n'a connaissance que du cryptogramme. La clé et le contenu du message sont pour lui des données inconnues et donc incertaines.

On dit qu'un système cryptographique est parfait si la connaissance du cryptogramme n'apporte à l'adversaire aucune information sur le message, autre que celle qu'il connaît *a priori*, comme par exemple les statistiques de la langue utilisée pour écrire le message. En d'autres termes, l'incertitude résiduelle de l'adversaire sur le message connaissant le cryptogramme est exactement égale à l'incertitude initiale sur le message. La théorie de l'information montre que :

- le système de Vernam, avec une bande aléatoire utilisée une seule fois, est parfait ;
- l'incertitude résiduelle sur le message connaissant le cryptogramme est toujours inférieure à l'incertitude sur la clé ;
- si le système cryptographique est parfait, alors l'entropie de Shannon du message est toujours inférieure à l'entropie de Shannon de la clé.

Par conséquent, tous les systèmes conventionnels qui utilisent une clé réduite, par exemple à 128 symboles binaires, et qui ne peuvent donc avoir une entropie supérieure à 128 bits, sont forcément imparfaits dès qu'ils sont utilisés pour chiffrer de longs messages dont l'entropie dépasse cette valeur. Un adversaire est en théorie capable de résoudre de tels systèmes. Il dispose d'assez de données pour extraire de l'information du cryptogramme. C'est l'effort à fournir pour retrouver cette information qui rend cette opération pratiquement impossible.

2 La sécurité calculatoire

Dans un système réel à clé de taille fixe, lorsque l'adversaire observe un nombre suffisant de cryptogrammes, il dispose d'assez d'information pour reconstituer

la clé. Prenons par exemple un système à substitution simple. Si la substitution secrète est tirée au hasard parmi les 26! substitutions possibles, l'incertitude de l'adversaire sur celle qui a été choisie vaut $\log_2(26!)$, soit environ 88 bits. Si le message clair est écrit en français, les statistiques des lettres dans cette langue nous apprennent que l'entropie moyenne d'une lettre est d'environ 3,96 bits. En première approximation, le cryptogramme peut être considéré comme aléatoire. Ce n'est en toute rigueur pas le cas, puisqu'en raison de fréquentes collisions, l'entropie de Rényi sera inférieure à celle d'une distribution aléatoire des lettres. Mais sous cette hypothèse simplificatrice, chaque observation d'une lettre de cryptogramme révélera en moyenne une information de $\log_2(26) \approx 4{,}7$ bits. Cela signifie que l'information apportée par l'observation d'au moins 119 lettres du cryptogramme dépasse de 88 bits l'information manquante sur le message clair, soit justement l'information nécessaire pour révéler la substitution utilisée. Cette valeur, ici de 119 lettres, s'appelle la *distance d'unicité* du système. Elle est par définition la quantité de cryptogramme qui apporte assez d'information pour déterminer la clé de manière unique. Il résulte de ces considérations que la sécurité des systèmes réels ne peut se fonder sur le manque d'information de l'adversaire, mais seulement sur son incapacité à mener à bien les calculs en raison de leur trop grande complexité. On parle alors de *sécurité calculatoire*. La résolution du cryptogramme est théoriquement accessible, mais elle est en pratique un problème insurmontable. La sécurité ne repose que sur le coût prohibitif du calcul requis pour le décryptement. Le dernier quart du XX$^{\text{e}}$ siècle a vu le développement de la théorie de la complexité qui s'est précisément attachée à étudier les coûts de calcul.

3.1 Classes de complexité

On dit qu'un problème appartient à la classe $\mathcal{P}$, pour *polynomial*, s'il peut être résolu par une machine de Turing déterministe en un nombre d'étapes borné par un polynôme de la taille des données d'entrée, exprimée par exemple en nombre de symboles binaires requis pour les coder. Il s'agit des problèmes dont la solution peut être trouvée par un algorithme efficace. Additionner, soustraire, multiplier ou diviser des entiers sont des problèmes polynomiaux. Un problème qui n'appartient pas à cette classe sera tenu pour difficile à résoudre, du moins pour certaines entrées.

Hormis pour les systèmes parfaits, la cryptographie exploite notre difficulté à résoudre certains problèmes pour rendre les services de confidentialité et d'authentification. Sans problème difficile, il n'y a pas de cryptographie. Certains de ces problèmes sont propres à la cryptographie, comme par exemple retrouver la clé d'un procédé de chiffrement symétrique à partir de couples

clair-cryptogramme. D'autres problèmes ont intéressé des générations de mathématiciens depuis plusieurs siècles, comme celui de la factorisation des entiers. Dans les deux cas, les solutions sont difficiles à trouver, mais une fois la solution connue, il est immédiat de vérifier qu'elle convient, en chiffrant les messages clairs avec la clé dans le premier cas, ou en multipliant les facteurs dans le second.

Encadré 6.3. La machine de Turing.

La machine de Turing formalise la notion de calculabilité. Par définition, les fonctions calculables sont celles qui le sont par une machine de Turing. La thèse de Church-Turing (1936) énonce que cette notion recouvre ce que le sens commun entend par calculabilité. Une machine de Turing comporte :

1. Une unité centrale de calcul qui peut se trouver dans un certain nombre fini d'états internes.
2. Un ruban illimité où figurent au départ des données à traiter, et où s'écrivent les résultats. Ces données s'expriment à l'aide d'un alphabet fini.
3. Une tête de lecture/écriture qui peut :
   – remplacer un caractère par un autre sur le ruban ;
   – déplacer le ruban à gauche ou à droite.

Le programme d'une telle machine est une séquence d'instructions constituées de quatre données (q,s,r,a), telles que si la machine se trouve dans l'état q et lit le symbole s sur le ruban, alors elle se positionne dans le nouvel état r et exécute l'action a qui peut être :

– un symbole à écrire sur le ruban à la place de s ;
– le déplacement du ruban vers la gauche ou vers la droite.

Le « *programme* » suivant, qui comprend quatre états numérotés de 1 à 5, calcule la somme de deux entiers représentés par une liste de bâtons inscrits sur le ruban et séparés par le symbole de séparation #, montrant, certes de façon très inefficace, que l'addition des entiers est une fonction calculable.

(**1**, |, **2**, #)	efface le premier bâton de x
(**2**, #, **2**, G)	se positionne sur le bâton suivant x
(**2**, |, **3**, #)	efface le second bâton de x
(**3**, #, **4**, G)	déplace le ruban jusqu'au premier bâton de y
(**4**, |, **4**, G)	
(**4**, #, **5**, |)	et remplace le caractère séparateur par un bâton

On ne connaît aujourd'hui aucune méthode pour résoudre ces problèmes en temps polynomial. Il est cependant possible d'imaginer pouvoir factoriser un entier en temps polynomial en utilisant un nombre de machines égal à la racine carrée de l'entier à factoriser, chacune d'entre elles le divisant par un candidat diviseur. Immanquablement, l'une des machines trouvera un reste nul et pourra afficher un facteur. Les problèmes qui peuvent être résolus en temps polynomial sur des machines qui présentent un parallélisme illimité appartiennent à la classe $\mathcal{NP}$, pour *Nondeterministic Polynomial*. Ce sont les problèmes qui peuvent être

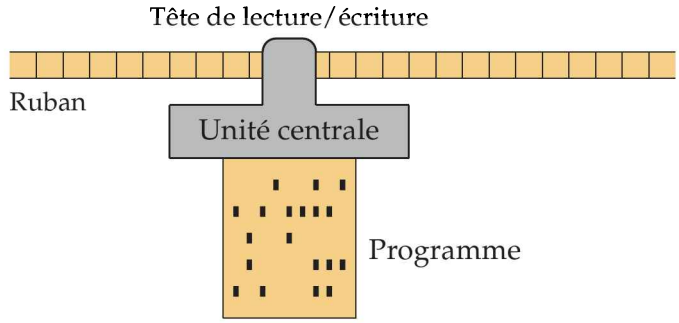


FIGURE 6.2. Schéma d'une machine de Turing. Elle est constituée d'un ruban illimité pouvant se déplacer à droite ou à gauche, d'une tête de lecture/écriture et d'une unité centrale contrôlant les actions de la machine.

résolus en temps polynomial sur une machine de Turing non déterministe. Une autre formulation décrit ces problèmes comme ceux qui sont vérifiables en temps polynomial. La classe $\mathcal{NP}$ contient bien sûr la classe $\mathcal{P}$, et une des grandes questions ouvertes de la théorie de la complexité est de savoir si la classe $\mathcal{NP}$ est ou non strictement plus grande que la classe $\mathcal{P}$. Savoir que $\mathcal{P} = \mathcal{NP}$ réjouirait nombre de scientifiques. Des solutions à des problèmes aujourd'hui pratiquement insolubles deviendraient accessibles. Mais, en contrepartie, les cryptographes perdraient des pans entiers de leur activité. L'avenir ne sera pas radieux pour tous.

Encadré 6.4. Machine de Turing déterministe, non déterministe.

Une machine de Turing est dite *déterministe* si à tout état interne et tout symbole lu sur le ruban, correspond au plus une instruction applicable. L'exécution du programme est alors parfaitement déterminée par la liste des instructions. Dans le cas contraire, la machine est dite *non déterministe*.

Dans une machine non déterministe, à une étape du calcul, il peut exister plusieurs actions applicables. On dit que le problème est résolu s'il existe au moins une suite d'instructions valides qui mène à la solution. On peut imaginer par exemple qu'un groupe illimité de machines effectue tous les calculs possibles en parallèle. Au moins l'une d'entre elles aboutira au résultat. Une autre image est celle d'un oracle qui indique la bonne instruction à exécuter lorsque plusieurs possibilités se présentent. Un tel oracle se comporte comme un moyen de vérifier un résultat déjà connu.

3.2 Construction de l'édifice cryptographique

La conception des schémas cryptographiques suit aujourd'hui une méthode qui consiste à s'appuyer sur des fonctions élémentaires ayant des propriétés admises, comme la multiplication des entiers et la difficulté de factoriser leur produit.

La sécurité est définie par l'objectif assigné à un adversaire, comme retrouver une information sur le message en clair, s'authentifier, forger une fausse signature, etc.

Il est admis que les seules attaques envisageables contre des systèmes cryptographiques ne peuvent reposer que sur des algorithmes de complexité polynomiale. Au-delà, l'attaque n'est pas réaliste. Il suffit d'augmenter légèrement la taille de la clé pour la rendre impraticable. Les fonctions utilisées pour bâtir l'édifice cryptographique sont des familles de fonctions, définies à partir d'un paramètre de sécurité. Ce paramètre n est par exemple le nombre de symboles binaires d'une clé. L'objectif de sécurité sera une résistance face à un adversaire doté d'une puissance de calcul bornée par un polynôme en n.

La sécurité sera avérée si une démonstration existe du fait que l'adversaire ne peut atteindre le but qui lui est assigné qu'avec une probabilité négligeable, malgré les moyens et les informations qui sont mis à sa disposition. La probabilité de réussite de l'adversaire est fonction de la valeur n du paramètre de sécurité. On dit qu'elle est *négligeable* si elle décroît plus vite que tout polynôme en $1/n$. Par exemple, une probabilité de succès inférieure à $1/2^n$ est négligeable et sera considérée comme satisfaisante pour la sécurité. Il ne reste alors qu'à choisir la valeur convenable du paramètre de sécurité n.

Encadré 6.5. Les preuves de sécurité.

Un schéma cryptographique est construit à partir d'une fonction élémentaire supposée associée à un problème difficile à résoudre, comme le système de Rabin qui repose sur la factorisation des entiers. L'objet de la preuve de sécurité est de prouver l'implication suivante :

Si le problème sous-jacent est difficile, alors le schéma est sûr.

La logique nous apprend que cette implication est équivalente à sa contraposée :

Si le schéma est faible, alors le problème sous-jacent est efficacement résoluble.

Le concepteur s'attachera à prouver cette dernière implication. Supposer la faiblesse du schéma revient à supposer l'existence d'un adversaire contre lui. Si l'on arrive à résoudre efficacement le problème difficile en exploitant cet adversaire, on aura montré l'implication. C'est exactement la démarche suivie pour prouver la sécurité du système cryptographique à clé publique de Rabin page 53. La conclusion s'impose : tant que le problème sous-jacent est difficile, le schéma cryptographique qui repose sur lui restera sûr.

Fonctions à sens unique

Dans un procédé de chiffrement symétrique sûr, on ne peut pas retrouver la clé, même si un couple clair-cryptogramme est connu. La fonction qui à la clé

associe, par exemple, le cryptogramme du message « *bonjour !* » est facilement calculable, mais doit être pratiquement impossible à inverser. Le cryptogramme d'un message connu ne doit pas révéler la clé. Ceci montre que s'il existe un chiffrement sûr, alors il s'en déduit une fonction à sens unique. Autrement dit, sans l'existence de fonctions à sens unique, la cryptographie symétrique ne peut exister. Ces fonctions constituent le socle sur lequel s'appuie l'édifice cryptographique.

L'objectif d'un adversaire contre une fonction à sens unique est de trouver l'antécédent d'une valeur y qu'on lui lance comme défi. Un adversaire ne doit réussir à exhiber un antécédent qu'avec une probabilité négligeable. Si une fonction est à sens unique, c'est qu'elle cache de l'information sur l'antécédent. Un prédicat difficile (*hard core predicate*) d'une fonction à sens unique est par définition une information binaire sur l'antécédent qu'il est difficile de retrouver à partir de la valeur de la fonction. Un adversaire polynomial ne doit pouvoir trouver la valeur d'un prédicat difficile qu'avec un avantage négligeable. Le théorème de Goldreich-Levin énonce que l'existence d'un prédicat difficile caractérise les fonctions à sens unique. Ce résultat est au cœur de la théorie cryptologique.

Encadré 6.6. Avantage d'un adversaire.

Lorsque l'adversaire doit choisir entre deux réponses également probables pour gagner, sa performance est évaluée par son *avantage* qui est par définition la quantité $2p - 1$, où p est sa probabilité de gagner. Ainsi, s'il répond au hasard, il a une chance sur deux de gagner et son avantage est nul. S'il gagne toujours, son avantage vaut 1 et s'il perd toujours, son avantage vaut -1. Un adversaire qui gagne toujours se déduit de ce dernier en inversant ses réponses.

Générateur pseudo-aléatoire

Un générateur pseudo-aléatoire agit comme un amplificateur d'aléa. Il s'agit d'un algorithme déterministe, qui opère sur les données d'une source aléatoire pour produire une séquence plus longue de données constituée de symboles qu'on ne peut discerner d'une véritable source aléatoire. Le degré d'expansion d'un générateur pseudo-aléatoire est le rapport entre la taille de données produites et la taille du germe qu'il utilise. Un résultat de la théorie de l'information énonce qu'il n'est pas possible de créer d'incertitude par la seule manipulation des données, montrant qu'un générateur pseudo-aléatoire ne génère pas réellement d'aléa, et que sa qualité est une notion calculatoire. Un adversaire polynomial ne doit pouvoir distinguer la sortie d'un générateur pseudo-aléatoire d'un véritable aléa qu'avec un avantage négligeable. La figure 6.3 montre comment construire

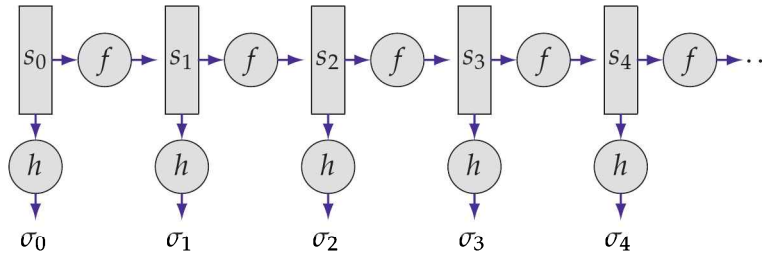


FIGURE 6.2. Construction d'un générateur pseudo-aléatoire avec une fonction à sens unique. À partir d'une graine s_0, la fonction à sens unique f est itérée pour calculer les états successifs $s_1, s_2 \ldots$ Les valeurs pseudo-aléatoires sont issues d'un prédicat difficile h de la fonction à sens unique. La suite $\sigma_0, \sigma_1 \cdots$ est indiscernable d'une suite parfaitement aléatoire.

un générateur pseudo-aléatoire à partir d'une fonction à sens unique f et d'un prédicat difficile h de cette fonction.

Un générateur pseudo-aléatoire permet de réaliser le chiffrement de messages plus longs que la clé tout en assurant un bon niveau de sécurité. Deux correspondants produisent la même séquence à partir d'une graine qu'ils partagent et, comme dans le chiffrement de Vernam, chaque symbole binaire du message est additionné modulo 2 au symbole correspondant de la chaîne pseudo-aléatoire produite avec la graine. Il est primordial de n'utiliser la séquence aléatoire qu'une seule fois (*one time pad*). Pour chiffrer un autre message, il faut changer de graine.

Famille de fonctions pseudo-aléatoires

Une famille de fonctions pseudo-aléatoires est une famille de fonctions, indicées par un paramètre et qui sont indiscernables d'une fonction aléatoire. Un adversaire contre une telle famille est un algorithme que l'on place en face d'un oracle qu'il peut interroger librement en lui soumettant un paramètre x. L'oracle lui répond avec une valeur $f(x)$. L'objectif de l'adversaire est de reconnaître si l'oracle réalise une fonction de la famille, ou une fonction aléatoire. La qualité de la famille pseudo-aléatoire se mesure à l'avantage de tout adversaire polynomial qui doit rester négligeable. La figure 6.4 montre comment construire une famille de fonctions pseudo-aléatoires à partir d'un générateur pseudo-aléatoire.

Pour réaliser un chiffrement par bloc, il faut que la fonction soit inversible et les fonctions construites selon le schéma de la figure 6.4 ne le sont pas nécessairement. Mais en utilisant cette construction dans un schéma de Feistel à trois tours, on obtient alors une permutation pseudo-aléatoire qu'un adversaire polynomial ne peut discerner d'une permutation aléatoire quelconque qu'avec un avantage négligeable. Une famille de permutations pseudo-aléatoires réalise un

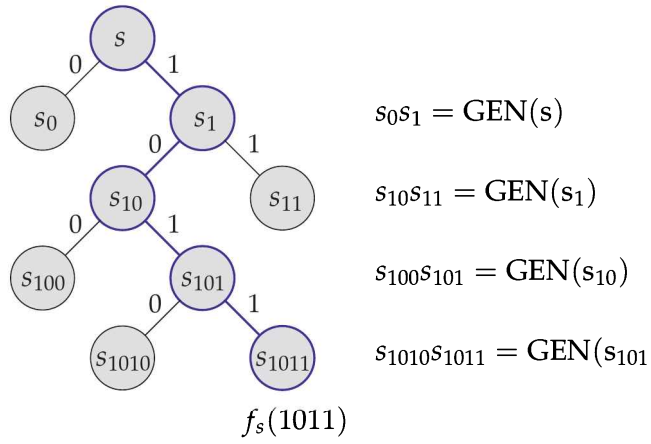


FIGURE 6.4. Construction d'une famille de fonctions pseudo-aléatoires à partir d'un générateur pseudo-aléatoire GEN de degré d'expansion 2. Une graine s est l'indice qui définit l'élément de la famille. Le paramètre x de la fonction est une séquence de symboles binaires $x_1 \cdots x_n$. À partir de la graine s, le générateur pseudo-aléatoire produit deux données s_0 et s_1, chacune composée d'autant de symboles binaires que s. Si x_0 vaut 0, c'est s_0 qui est retenu, sinon, c'est s_1. Le procédé est itéré avec chaque symbole binaire x_i du paramètre pour aboutir à la valeur $y = f_s(x)$.

chiffrement par bloc et peut servir de cœur à un procédé de chiffrement utilisant par exemple le mode opératoire CBC (voir paragraphe 5.6 page 40).

Fonctions de hachage

Les fonctions de hachage servent à produire l'empreinte d'un message en vue de sa signature numérique. Le modèle idéal est une fonction aléatoire, où les valeurs seraient comme tirées au hasard, sans lien entre elles. Mais comme il s'agit d'une fonction déterministe, deux appels avec le même paramètre doivent fournir le même résultat. Une telle fonction s'appelle un *oracle aléatoire*. Lorsqu'un schéma cryptographique a besoin de l'empreinte d'un message, il s'adresse à l'oracle qui se charge de la lui fournir. L'hypothèse d'indépendance entre les différentes valeurs est très forte et n'est pas réaliste. Un oracle aléatoire reste un modèle idéal sans existence réelle. Les deux critères réalistes pour une fonction de hachage sont la résistance aux collisions et la résistance au second antécédent. Si un adversaire ne peut trouver deux messages qui ont la même empreinte qu'avec une probabilité négligeable, on dit que la fonction de hachage est résistante aux collisions. La résistance au second antécédent est une notion plus faible, et donc plus facile à satisfaire en pratique. Il s'agit de s'assurer que, pour tout message, il est difficile d'en trouver un autre, différent du premier, qui a la même empreinte. Il faut noter que l'analyse de la sécurité est une notion asymptotique qui ne peut

être pertinente que pour des familles de fonctions de hachage, dépendantes d'un paramètre de sécurité. On ne s'intéresse qu'aux adversaires dont les moyens de calcul sont bornés par un polynôme de ce paramètre.

Pour formaliser la résistance au second antécédent, on a aussi besoin, pour un paramètre de sécurité donné, de considérer une famille de fonctions de hachage. Un adversaire contre le second antécédent présente tout d'abord un message au maître du jeu. Celui-ci ne lui indique qu'ensuite l'indice s de la fonction qu'il aura à attaquer. L'objectif de l'adversaire est alors de trouver un second message produisant la même empreinte avec cette fonction d'indice s. La famille sera dite sûre si l'adversaire ne réussit qu'avec une probabilité négligeable relativement au paramètre de sécurité. On montre que, lorsqu'on compose une permutation pseudo-aléatoire avec un élément d'une famille universelle (voir encadré 6.7 ci-après), on obtient un élément d'une famille de fonctions de hachage qui satisfait cette propriété. Il en résulte en particulier que l'existence de ce type de famille ne repose que sur l'existence de permutations pseudo-aléatoires, qui elles-même ne découlent que de l'existence de fonctions à sens unique.

Encadré 6.7. Famille universelle.

Considérons un ensemble E de n éléments et un ensemble F de m éléments, avec $m < n$. Une famille d'applications de E vers F est dite *universelle* si pour tout couple (x_1,x_2) d'éléments de E, la proportion d'éléments de la famille qui ont la même image en x_1 et en x_2 ne dépasse pas $1/m$. Cela signifie que l'application d'un élément quelconque de la famille à un élément quelconque de l'ensemble E ne va pas créer un nombre anormalement élevé de collisions. Cette notion a initialement été introduite pour limiter les collisions dans les tables de recherche par clé de hachage. Elle est aussi au cœur de la théorie cryptographique et est un outil central pour la cryptographie quantique.

Un exemple : si E et F sont des espaces vectoriels, l'ensemble des applications linéaires de E vers F constitue une famille universelle.

Signature à clé publique

La signature numérique à clé publique assure le service d'authentification asymétrique. Seul celui qui dispose de la clé privée peut signer, et tous peuvent vérifier la signature avec la clé publique correspondante. Le premier schéma de signature asymétrique réaliste est le RSA, publié en 1978. Le chiffrement RSA se transforme naturellement en signature en échangeant les rôles des clés publiques et privées. Finalement, la création de la signature apparaît comme une sorte de *chiffrement avec la clé privée*, la vérification de cette signature étant un déchiffrement avec la clé publique. Cette symétrie suggère l'idée que la signature est duale

du chiffrement à clé publique et que cela nécessite, comme pour le chiffrement, une fonction à sens unique avec trappe secrète comme cela est présenté page 50.

Cette approche est fausse ! D'une part, il n'est pas toujours possible de transformer un chiffrement asymétrique en signature. Par exemple, si le chiffrement n'est pas déterministe, comme dans le schéma de McEliece présenté au paragraphe 2.6 page 56, il n'est possible d'appliquer la fonction de déchiffrement, invoquant la clé privée, qu'à un message qui est le résultat d'un chiffrement, et pas toujours à un message arbitraire. D'autre part, le chiffrement à clé publique nécessite une fonction à sens unique avec trappe secrète pour l'inverser, alors que pour construire une signature asymétrique, il suffit seulement de disposer d'une simple fonction à sens unique. La signature, même asymétrique, ne requiert pas de trappe secrète !

Dès 1976, Leslie Lamport de la firme Massachusetts Company Associates a suggéré un schéma de signature, certes assez inefficace et incomplet, utilisable une fois seulement pour signer un unique document, mais dont l'intérêt est qu'il ne nécessite rien d'autre qu'une fonction à sens unique. Il s'agit d'une signature asymétrique à clé jetable. La clé privée est constituée de deux suites de données $x_1, \ldots x_n$ et $y_1, \ldots y_n$. La clé publique est l'image de ces données par une fonction à sens unique. Pour tout indice i, posons $a_i = f(x_i)$ et $b_i = f(y_i)$. Pour signer un message m constitué de n symboles binaires m_i, on révèle l'élément secret x_i si $m_i = 0$ et l'élément y_i si $m_i = 1$. Pour vérifier la signature, il suffit d'appliquer la fonction à sens unique aux termes de la signature et de vérifier qu'ils sont égaux aux éléments correspondants dans la clé publique.

Ce schéma a été par la suite amélioré pour pouvoir utiliser la clé plusieurs fois, en utilisant une famille de fonctions de hachage résistantes au second antécédent. La construction qui en résulte reste encore assez inefficace, puisque la signature d'un document reste beaucoup plus longue que le document lui-même, mais le résultat théorique est fondamental :

> *la signature asymétrique ne repose sur rien d'autre que sur l'existence des fonctions à sens unique, une trappe secrète n'est pas nécessaire !*

Sécurité du chiffrement

Le niveau de sécurité requis aujourd'hui pour le chiffrement s'appelle la *sécurité sémantique*. Il s'agit d'une version calculatoire de la sécurité parfaite. Elle énonce que le cryptogramme n'est d'aucune utilité à l'adversaire. Tout ce qu'il peut calculer avec le cryptogramme, il peut le calculer tout aussi bien sans lui. La sécurité sémantique énonce qu'évaluer le moindre bit d'information sur le message clair à partir du cryptogramme ne peut se faire qu'avec un avantage négligeable.

La propriété d'*indistinguabilité*, notée IND, est une autre façon d'attester la sécurité du chiffrement. Elle est décrite comme un jeu proposé à l'adversaire. Dans un premier temps, l'adversaire choisit deux messages clairs distincts m_0 et m_1. Ensuite, le maître du jeu fournit à l'adversaire un cryptogramme qui est celui de l'un des deux messages, et l'adversaire doit alors déterminer de quel message est ce cryptogramme : m_0 ou m_1 ? Un schéma de chiffrement est sûr si l'adversaire ne peut apporter la réponse correcte qu'avec un avantage négligeable. Un résultat de la théorie cryptographique est l'équivalence de la propriété d'indistinguabilité et de la sécurité sémantique.

Pour aider l'adversaire, on peut l'autoriser à avoir recours à ce qui est appelé un *oracle de chiffrement* qui chiffre les messages de son choix. On parle alors d'attaque à clairs choisis (CPA, *Chosen Plaintext Attack*). S'il s'agit d'un mécanisme de chiffrement à clé publique, la clé de chiffrement étant publique, cet oracle est toujours accessible. L'adversaire peut aussi avoir recours à un *oracle de déchiffrement* à qui il peut demander le déchiffrement d'un message de son choix, à l'exception bien sûr du cryptogramme qui lui est demandé de résoudre, sans quoi le jeu serait sans intérêt. Dans ce cas, on parle d'attaque à cryptogramme choisi (CCA, *Chosen Cyphertext Attack*). Un chiffrement sera toujours évalué dans des conditions qui sont les plus favorables à l'adversaire. Certains mécanismes ne résistent pas à ces conditions d'évaluation. Le chiffrement RSA dans sa version élémentaire n'atteint pas la sécurité sémantique, comme tout autre mode de chiffrement déterministe. Pour résister à une attaque à clair choisi, il est nécessaire d'introduire une certaine quantité d'aléa dans le chiffrement. Imaginons par exemple qu'on cherche à sécuriser le vote à un référendum par le chiffrement des réponses « *oui* » ou « *non* » à une question posée. Si le chiffrement est déterministe et à clé publique, chiffrer ces deux réponses permet de connaître tous les votes. Le chiffrement ElGamal, dont le processus inclut le choix d'une donnée aléatoire, est sûr dans une attaque à clair choisi, mais ne l'est pas dans une attaque à cryptogramme choisi.

Pour résister aux attaques à cryptogramme choisi, il faut empêcher l'adversaire de modeler des cryptogrammes à sa guise. Cela est fait en adjoignant des données qui attestent de la validité du cryptogramme et permettent de détecter ceux qui sont contrefaits. Pour résister à une attaque à cryptogramme choisi, le chiffrement ne doit pas être *malléable*. Les chercheurs Mihir Bellare et Philip Rogaway ont proposé un mécanisme qui atteint la sécurité sémantique dans ces conditions très sévères, appelé OAEP, comme *Optimal Asymetric Encryption Padding*, où une quantité aléatoire est adjointe au message, puis brouillée par un schéma de Feistel à deux tours comme schématisé sur la figure 6.5. La preuve de sécurité du schéma OAEP repose de façon cruciale sur l'hypothèse que les fonctions utilisées dans le schéma de Feistel sont des oracles aléatoires. Cette

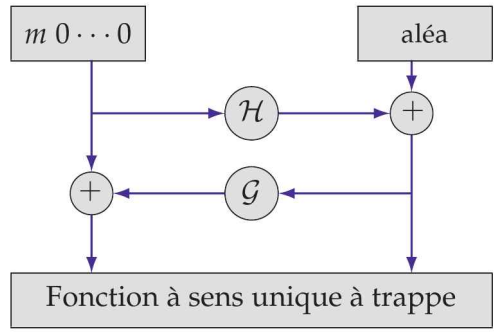


FIGURE 6.4. *Optimal Asymetric Encryption Padding* – OAEP. Il s'agit d'un mécanisme qui traite le message à chiffrer avant de le soumettre à une fonction à sens unique avec trappe pour le rendre sémantiquement sûr dans une attaque à cryptogramme choisi. Le traitement consiste à le compléter avec des zéros et de l'aléa, puis de faire passer le tout dans un schéma de Feistel à deux tours. Pour atteindre ce niveau de sécurité, les fonctions $\mathcal{G}$ et $\mathcal{H}$ sont supposées être des oracles aléatoires et la fonction à sens unique à trappe secrète doit avoir une propriété supplémentaire : il ne doit pas être possible de calculer la partie gauche de l'entrée à partir de sa sortie, ce qui est le cas pour la fonction RSA.

hypothèse est considérée comme irréaliste, l'oracle aléatoire restant un idéal trop fort pour modéliser les fonctions réelles.

En 1998, les chercheurs Ronald Cramer et Victor Shoup ont défini un schéma de chiffrement qui porte leur nom et qui a la propriété d'être prouvé sémantiquement sûr dans une attaque à cryptogramme choisi sous l'hypothèse que le problème décisionnel Diffie-Hellman est difficile, c'est-à-dire connaissant les entiers g, g^x et g^y modulo un nombre premier, il est difficile de décider si une valeur s donnée est ou non égale à g^{xy}. Ce schéma n'utilise par ailleurs que des fonctions de hachage résistantes au second antécédent, ce qui constitue une hypothèse de loin plus plausible en pratique.

3.3 Les cinq mondes d'Impagliazzo

L'existence même des fonctions à sens unique n'est pas assurée. Il s'agit pour l'instant d'une situation à la fois empirique et virtuelle. Il existe certaines fonctions qu'on ne sait pas inverser efficacement, mais peut-être les progrès algorithmiques les rendront-elles inversibles, et on ne sait pas non plus prouver la difficulté de leur inversion. Il est possible que l'on prouve que certaines fonctions sont vraiment à sens unique, c'est-à-dire que le problème de leur inversion est un problème prouvé difficile. Le monde dans lequel les fonctions à sens unique ont une existence autre qu'empirique n'est pour l'instant qu'un monde à la fois imaginaire et possible. Le chercheur américain Russel Impagliazzo a imaginé

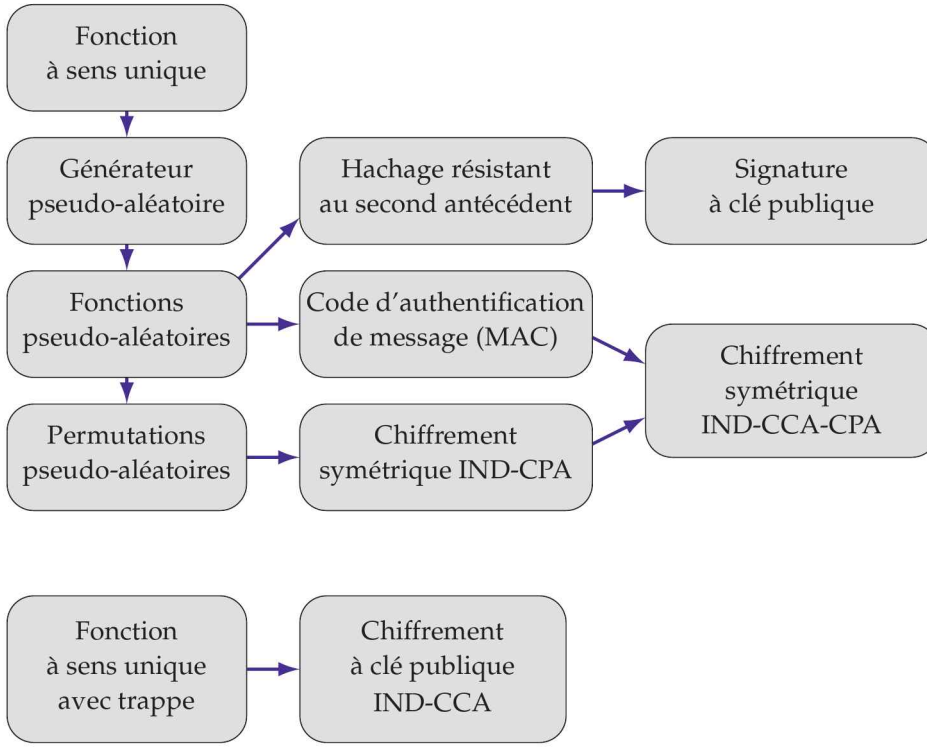


FIGURE 6.6. L'édifice cryptographique. Les constructions du chiffrement symétrique et de la signature, tant symétrique qu'asymétrique, reposent sur les fonctions à sens unique, facilement calculables, mais dont un antécédent d'une valeur est dans la plupart des cas difficile à trouver. Le chiffrement à clé publique, lui, nécessite une propriété plus forte : la fonction à sens unique doit avoir une trappe, c'est-à-dire une information supplémentaire secrète qui permet à ses détenteurs et à eux seuls de pouvoir inverser la fonction. Cet édifice montre sur quelles briques s'appuie la construction des procédés de chiffrement qui atteignent la propriété d'indistinguabilité (IND), qui est le niveau standard exigible aujourd'hui en matière de sécurité, lors d'attaques à clairs choisis (CPA, *Chosen Plaintext Attack*) ou à cryptogrammes choisis (CCA, *Chosen Cyphertext Attack*).

cinq mondes selon la hiérarchie des problèmes que l'on sait résoudre ou non, avec les conséquences sur le type de cryptographie qu'on peut y pratiquer.

Cryptomania

Cryptomania est le monde conforme à celui dans lequel nous évoluons aujourd'hui. Il y existe des fonctions à sens unique avec trappe, comme la fonction RSA, qu'on ne sait inverser que grâce à la connaissance de la factorisation du module, mais son inversion reste un problème difficile pour tous ceux qui ignorent

cette factorisation. Dans ce monde, il est possible de faire du chiffrement asymétrique : le chiffrement est accessible à tous avec une clé publique, mais le déchiffrement reste réservé aux détenteurs d'une information supplémentaire privée et maintenue secrète.

Mais nous n'avons aucune preuve de l'existence de telles fonctions. Leur existence n'est qu'empirique, une observation de l'état actuel de nos savoir-faire. Aujourd'hui, nous ne savons pas inverser la fonction RSA sans connaître la factorisation du module. Et factoriser le module reste aujourd'hui un problème en pratique insoluble pour les très grandes valeurs. Mais ce problème est-il intrinsèquement difficile ? Selon la réponse que la recherche future apportera à cette question, il se peut que le monde Cryptomania disparaisse.

Minicrypt

Minicrypt est un deuxième monde imaginaire possible. Dans Minicrypt, une fonction est à sens unique ou ne l'est pas. Les fonctions à sens unique existent, mais pas les fonctions à sens unique avec trappe. Il n'est pas possible de restreindre la possibilité d'inverser une fonction à la seule connaissance d'une trappe cachée. Si elle est inversible pour certains, alors elle est inversible pour tous. Il est possible dans Minicrypt de poser aux autres des problèmes difficiles dont on connaît la solution. Je choisis par exemple une fonction à sens unique f. Une telle fonction existe dans le monde Minicrypt. Je choisis un élément x au hasard. Je calcule $y = f(x)$. Il sera difficile à tout autre que moi de trouver un antécédent à l'élément y. Mais contrairement à Cryptomania, je ne pourrai pas donner d'indication à certains pour qu'il sachent inverser. Le problème difficile d'inversion des fonctions à sens unique reste toutefois utilisable pour faire une cryptographie minimale. On peut y faire de la cryptographie symétrique, mais également de la signature à clé publique et de l'authentification sans divulgation. Par contre, le chiffrement à clé publique n'existe plus dans Minicrypt. Contrairement à une idée répandue, la signature asymétrique et le chiffrement asymétrique n'appartiennent pas au même monde.

Pessiland

Un troisième monde imaginaire possible est Pessiland. Il ne peut exister que si Minicrypt n'existe pas. Dans Pessiland, il existe des problèmes faciles à vérifier mais difficiles à résoudre, y compris en moyenne, mais il n'existe pas de fonction à sens unique. Toutes les fonctions efficacement calculables sont aussi efficacement inversibles. Ce monde est considéré par Impagliazzo comme le pire qui puisse exister. Les scientifiques ne seront pas satisfaits, car la solution à certains problèmes restera inaccessible, mais cette difficulté de résolution ne

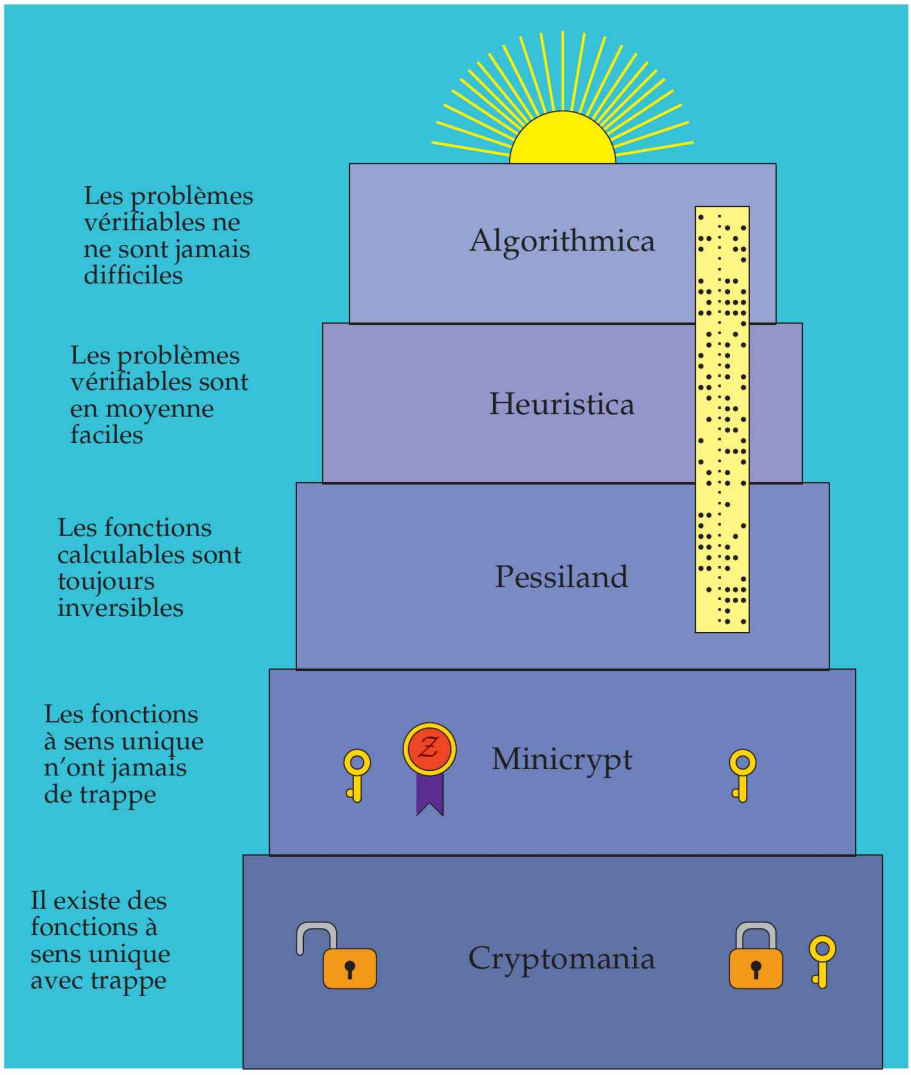


FIGURE 6.2. Les cinq mondes d'Impagliazzo. Ces mondes sont des mondes imaginaires possibles en l'état actuel de nos connaissances. Le développement de la théorie pourrait, soit les rendre réels et non plus imaginaires, soit les faire disparaître. Toute la cryptographie, et en particulier le chiffrement à clé publique, appartient au monde Cryptomania qui est notre monde empirique actuel. Le chiffrement symétrique et la signature à clé publique appartiennent au monde Minicrypt. La seule cryptographie utilisable dans les autres mondes est la cryptographie inconditionnellement sûre comme le chiffre de Vernam avec bande aléatoire. Il est étonnant de noter que la signature à clé publique appartient au monde Minicrypt, alors que le chiffrement à clé publique appartient, lui, au monde Cryptomania.

s'accompagne d'aucun avantage pour réaliser de la cryptographie. À partir de Pessiland, la seule cryptographie possible est celle, inconditionnellement sûre, définie par Vernam et Shannon avec des clés jetables, aléatoires, très longues et utilisables une fois seulement.

Heuristica

Ce monde commence à satisfaire les scientifiques. Dans Heuristica, les problèmes faciles à vérifier sont en moyenne faciles à résoudre. Certes la classe $\mathcal{P}$ n'y est pas égale à la classe $\mathcal{NP}$, ce qui signifie qu'il y reste des problèmes facilement vérifiables et difficiles à résoudre, mais dans le pire des cas seulement. On trouvera cependant toujours une instance d'un problème qui soit difficile à résoudre.

Algorithmica

Ce monde est celui, toujours hypothétique aujourd'hui, où $\mathcal{P} = \mathcal{NP}$. Tout les problèmes qui sont faciles à vérifier y sont aussi faciles à résoudre. Il est impossible d'y poser le moindre problème difficile dont certains seulement connaîtraient une solution, et que tous peuvent vérifier.

Pour l'instant, ces cinq mondes sont imaginaires. Nous ne savons toujours pas prouver leur existence, ni leur impossibilité. Cryptomania est, peut-être provisoirement seulement, celui dans lequel nous vivons virtuellement. Mais le développement de nos connaissances dans la théorie de la complexité rendra peut-être réel l'un de ces mondes, faisant disparaître les mondes précédents. Le premier à subir un tel sort serait Cryptomania. Si l'on arrivait à démontrer que s'il existe un moyen pour inverser une fonction alors ce moyen ne peut pas être maintenu secret, cela ferait disparaître Cryptomania et nous ferait entrer dans Minicrypt. Toutefois, tant que Minicrypt existe, du moins empiriquement, le chiffrement symétrique et la signature à clé publique resteront possibles. Et si l'on arrivait à démontrer que toute fonction calculable efficacement est aussi efficacement inversible, faisant disparaître Minicrypt, la seule cryptographie applicable serait alors la cryptographie parfaite de type Vernam avec une bande aléatoire.

7

Apport de la physique quantique

La physique quantique offre un cadre théorique qui rend compte de phénomènes en contradiction avec nos idées intuitives sur la matière. L'observation du monde microscopique a obligé les physiciens à repenser la matière et à en développer une description abstraite.

La théorie quantique a un impact aussi bien sur la cryptographie que sur la cryptanalyse. Du côté de l'attaque, cela a conduit à l'élaboration d'un nouveau modèle de calcul qui pourrait être mis en œuvre sur un *calculateur quantique*, dont l'existence est pour le moment théorique. Dans ce modèle, le chercheur Peter Shor de l'entreprise américaine AT&T a publié en 1994 un algorithme qui porte son nom et qui permettrait, si les calculateurs quantiques voyaient le jour, de factoriser les entiers avec une complexité polynomiale en temps et en espace. Un algorithme similaire, lui aussi efficace, permettrait de calculer le logarithme dans l'anneau des entiers modulo un nombre premier.

L'exécution de ces algorithmes sur un calculateur quantique rendrait caduque la cryptographie reposant sur l'arithmétique, comme le RSA ou la signature DSA. En revanche, la physique quantique propose un protocole cryptographique de remplacement pour que deux correspondants puissent partager un secret à distance en échangeant des données sur un canal supposé publiquement accessible.

Après une présentation des fondements sur lesquels repose la notion d'information quantique, ce chapitre explique comment un calculateur quantique permettrait de factoriser les entiers, et comment certains dispositifs réalisent déjà un échange de clé secrète entre deux protagonistes.

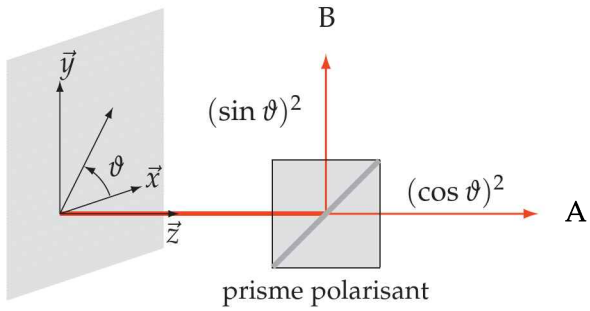


FIGURE 7.1. La polarisation linéaire du photon. Le flux lumineux dont la polarisation fait un angle ϑ avec la direction horizontale est partagé par le prisme polarisant en deux faisceaux. L'un suit la direction A en étant polarisé horizontalement avec une intensité proportionnelle à $(\cos \vartheta)^2$, l'autre suit la direction B en étant polarisé verticalement avec une intensité proportionnelle à $(\sin \vartheta)^2$. Si c'est un photon unique qui traverse le prisme, il suivra la direction A et sera polarisé horizontalement avec une probabilité égale à $(\cos \vartheta)^2$ et suivra la direction B et sera polarisé verticalement avec une probabilité égale à $(\sin \vartheta)^2$. Si ce photon traverse à nouveau un prisme polarisant identique, sa trajectoire ne sera plus incertaine, mais parfaitement déterminée. Il suivra avec certitude la direction définie par sa polarisation initiale.

1 Information et calcul quantique

1.1 L'unité binaire d'information quantique : le qubit

Une information quantique binaire est portée par une donnée sur l'état d'un système physique pouvant être décrit par un vecteur de deux composantes complexes. Par exemple, la polarisation linéaire d'un photon est représentée par un vecteur réel dont la direction est située dans le plan perpendiculaire à sa trajectoire et porte une information binaire : polarisation horizontale ou verticale.

De façon abstraite, une *unité d'information quantique*, appelée qubit (*Quantum Bit*), est un espace vectoriel hermitien de dimension 2, c'est-à-dire l'ensemble des vecteurs du plan représentés par deux composantes complexes dans une base orthonormée. La valeur d'un qubit est un vecteur normalisé de cet espace, c'est-à-dire un vecteur de norme 1, la norme d'un vecteur étant la somme des carrés des modules de ses deux composantes. Ce vecteur représente l'information sur l'état de la particule.

1.2 L'observation d'un qubit

La valeur d'un qubit est une donnée qui n'est accessible que par l'intermédiaire d'une mesure qui permet d'observer l'état de la particule. Pour réaliser cette mesure, il faut choisir *a priori* la base orthonormée dans laquelle s'effectuera la mesure. L'usage est de noter les vecteurs de cette base $|0\rangle$ et $|1\rangle$, respectivement

associés aux valeurs d'information binaire 0 et 1. Dans cette base, l'état $|\alpha\rangle$ d'un qubit est un vecteur qui se décompose en $|\alpha\rangle = x\,|0\rangle + y\,|1\rangle$, où x et y sont des composantes complexes qui vérifient $|x|^2 + |y|^2 = 1$.

Encadré 7.1. La sphère de Bloch.

La valeur d'un qubit est définie par deux composantes complexes. Chacune est elle-même définie par une partie réelle et une partie imaginaire, ce qui fait en tout quatre composantes réelles. Mais on peut remarquer que, si λ est un nombre complexe de module 1, les valeurs $|\alpha\rangle$ et $\lambda|\alpha\rangle$ sont porteuses de la même information, puisque pour ces deux valeurs, quelle que soit la base d'observation, le résultat de la mesure obéira exactement à la même loi de probabilité. En écrivant les composantes sous forme exponentielle, cela signifie que les deux valeurs :

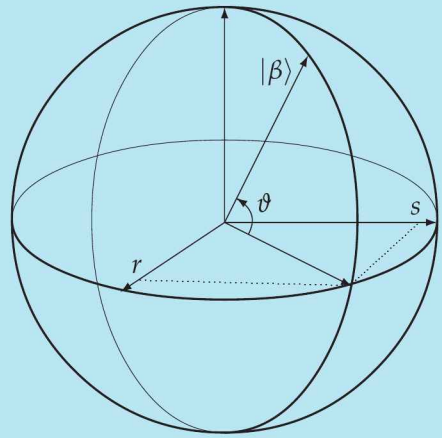
$$|\alpha\rangle = re^{i\varphi}|0\rangle + se^{i\psi}|1\rangle$$

$$\text{et} \quad e^{-i\varphi}|\alpha\rangle = r|0\rangle + se^{i(\psi-\varphi)}|1\rangle$$

sont équivalentes. On peut donc toujours supposer que la première composante de la valeur est un nombre réel. La valeur d'un qubit est donc entièrement définie par trois paramètres réels : deux composantes r et s vérifiant $r^2 + s^2 = 1$ et un angle ϑ compris entre $-\pi$ et π :

$$|\beta\rangle = r|0\rangle + se^{i\vartheta}|1\rangle$$

Ces trois paramètres définissent un point sur une sphère de rayon unité. Cette sphère, qui décrit les valeurs d'un qubit, s'appelle *la sphère de Bloch.*



L'observation de la valeur d'un qubit obéit aux lois suivantes :

– le résultat de la mesure est incertain. Il est donné par un vecteur aléatoire $|R\rangle$ qui ne peut être que l'un des vecteurs de la base dans laquelle est

effectuée la mesure. Si la mesure est effectuée dans la base $\{|0\rangle,|1\rangle\}$, le vecteur $|R\rangle$ vaut soit $|0\rangle$, soit $|1\rangle$;

- les composantes x et y de l'état dans la base d'observation $(|0\rangle,|1\rangle)$ définissent les probabilités respectives du résultat $|R\rangle$. Le résultat sera $|0\rangle$ avec une probabilité égale à $|x|^2$ et sera $|1\rangle$ avec une probabilité égale à $|y|^2$:

$$\text{Prob}(|R\rangle = |0\rangle) = |x|^2 \qquad \text{et} \qquad \text{Prob}(|R\rangle = |1\rangle) = |y|^2$$

  Dans le premier cas, on dit qu'on a observé la valeur 0, et dans le deuxième cas, qu'on a observé la valeur 1. Un vecteur d'état étant normalisé, la somme des probabilités d'observer l'une ou l'autre des valeurs vaut bien 1 ;
- après la mesure, si la valeur 0 a été observée, le qubit se trouve dans l'état $|0\rangle$, et si la valeur 1 a été observée, il se trouve dans l'état $|1\rangle$, conduisant à considérer que l'observation a créé l'état de la particule.

Il résulte des principes énoncés ci-dessus que l'observation change selon la base dans laquelle la mesure est effectuée. La mesure peut parfois conduire à un résultat certain. C'est le cas lorsque l'état, résultant d'une précédente observation, est l'un des vecteurs de la base dans lequel il est observé. Ainsi, dans la base $(|0\rangle,|1\rangle)$, le vecteur $|0\rangle$ s'exprime $|0\rangle = 1\,|0\rangle + 0\,|1\rangle$. L'observation de la valeur $|R\rangle = |0\rangle$ aura pour résultat $|0\rangle$ avec certitude.

Selon le troisième point, l'observation d'un qubit détermine son état. Une seconde mesure dans la même base donnera avec certitude la même valeur que la première. L'état initial avant la mesure, s'il existe, est perdu, rendant le processus d'observation irréversible. Cette propriété est exploitée en cryptographie quantique pour s'assurer que l'information transmise entre deux correspondants n'a pas été interceptée.

Tant qu'il n'est pas observé, l'état de la particule qui porte l'information du qubit n'est pas défini. Alors qu'une information classique ne peut prendre que l'une des deux valeurs 0 ou 1, un qubit peut se trouver dans une *superposition linéaire des états* $|0\rangle$ *et* $|1\rangle$ couvrant toutes les combinaisons linéaires de ces deux vecteurs. L'observation crée l'état dans l'une ou l'autre des valeurs définies par la base d'observation. Cette superposition des états est exploitée pour conduire un parallélisme virtuel dans les calculs, et obtenir ainsi une algorithmique plus efficace (Fig. 7.2).

1.3 Registres quantiques

Un registre quantique est par définition l'association de plusieurs qubits. La réalisation la plus simple d'un registre est l'association de deux qubits indépendants. L'indépendance est à comprendre au sens probabiliste. Elle signifie que l'observation de l'un n'a pas d'incidence sur l'observation de l'autre. Il y a quatre

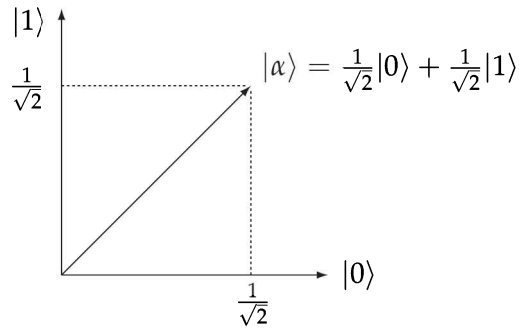


FIGURE 7.2. Exemple de superposition linéaire. La valeur $|\alpha\rangle$ du qubit se décompose dans la base orthonormée $(|0\rangle,|1\rangle)$ en $|\alpha\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$. Avant son observation, la particule se trouve dans une superposition équilibrée des états $|0\rangle$ et $|1\rangle$. L'observation projettera son état sur l'une ou l'autre des valeurs avec une probabilité égale à $1/2$.

valeurs possibles qui résultent de l'observation de deux qubits, et l'hypothèse d'indépendance signifie que la probabilité d'observer un couple donné de valeurs pour l'un et l'autre des qubits est égale au produit des probabilités pour chacun des qubits.

Pour deux qubits indépendants dont les états respectifs sont $|a\rangle = x|0\rangle + y|1\rangle$ et $|b\rangle = u|0\rangle + v|1\rangle$, il est donc naturel de considérer que l'état $(|a\rangle,|b\rangle)$ du couple de qubits est un vecteur de quatre composantes, appelé *produit tensoriel* des deux vecteurs $|a\rangle$ et $|b\rangle$. Ce vecteur de dimension quatre se décompose selon une base constituée de quatre vecteurs qui sont notés $|00\rangle$, $|01\rangle$, $|10\rangle$ et $|11\rangle$:

$$|a\rangle \otimes |b\rangle = xu|00\rangle + xv|01\rangle + yu|10\rangle + yv|11\rangle$$

Les quatre composantes de cet état, composé de deux qbits, définissent bien les probabilités d'observation des couples de valeurs possibles pour les qubits $|a\rangle$ et $|b\rangle$. On notera en particulier que la somme des carrés des modules de ces composantes vaut bien 1.

Un registre de taille deux peut aussi décrire une association de deux qubits qui ne sont pas indépendants. Ainsi, un registre de deux qubits est-il défini de façon très générale comme étant un espace hermitien de dimension quatre, et un état de ce registre comme un vecteur normalisé de cet espace. Dans ce cas, les résultats des observations des deux qubits peuvent ne pas être indépendants. Prenons par exemple un registre de deux qubits dont l'état est défini par le vecteur suivant :

$$\frac{1}{\sqrt{2}}|01\rangle - \frac{1}{\sqrt{2}}|10\rangle$$

Observer le premier qubit aura pour résultat 0 ou 1 de manière également probable. Mais si c'est la valeur 0 qui est observée, alors l'observation du second

qubit aura pour valeur 1 avec certitude. Cette propriété qui lie les valeurs observées de deux qubits différents est une manifestation de ce qu'on appelle *l'intrication quantique* ; on dit que les deux qubits sont *intriqués*.

Encadré 7.2. Observation d'un qubit d'un registre.

Considérons l'état suivant d'un registre de deux qubits :

$$|a\rangle = x|00\rangle + y|01\rangle + z|10\rangle + t|11\rangle.$$

L'observation du premier qubit projettera de façon aléatoire l'état du registre sur le plan engendré par les deux premiers vecteurs $|00\rangle$ et $|01\rangle$, ou par les deux derniers $|10\rangle$ et $|11\rangle$, selon que la valeur observée sera 0 ou 1. L'état du second qubit après cette observation sera le vecteur normalisé résultant de cette projection dans la base $(|0\rangle,|1\rangle)$ qui le définit :

- si la valeur observée du premier qubit est 0, l'état du second qubit devient $\frac{x}{\sqrt{x^2+y^2}}|0\rangle + \frac{y}{\sqrt{x^2+y^2}}|1\rangle$. Cela survient avec une probabilité égale à $|x|^2 + |y|^2$;
- si la valeur observée du premier qubit est 1, l'état du second qubit devient $\frac{z}{\sqrt{z^2+t^2}}|0\rangle + \frac{t}{\sqrt{z^2+t^2}}|1\rangle$. Cela survient avec une probabilité égale à $|z|^2 + |t|^2$.

1.4 Les portes quantiques

Les portes quantiques sont des dispositifs physiques qui opèrent sur une particule. Elles sont décrites par la transformation agissant sur la valeur du qubit porteur de l'information. Ce sont les analogues des portes logiques qui réalisent les calculs en électronique numérique. Les lois de la physique quantique nous apprennent que la transformation d'un état quantique est nécessairement le résultat d'une opération linéaire inversible, et comme le nouvel état est un vecteur normé, cette opération linéaire doit préserver la norme. Une telle transformation est appelée *unitaire*.

L'inverseur

L'inverseur quantique opère sur un qubit et transforme sa valeur $|\alpha\rangle = x|0\rangle + y|1\rangle$ en $\neg|\alpha\rangle = y|0\rangle + x|1\rangle$ où les composantes sont échangées. Ainsi, si la valeur initiale est l'un des vecteurs de la base d'observation, l'inverseur le transforme en l'autre vecteur de la base, $|0\rangle$ est transformé en $|1\rangle$ et vice versa, imitant exactement l'inverseur logique. Si l'état est une superposition, la probabilité d'observer une valeur avant inversion est la même que la probabilité d'observer l'autre valeur après l'inversion. L'état inversé s'interprète géométriquement, dans l'espace complexe de dimension 2, comme le symétrique de l'état initial relativement à la diagonale principale. En décrivant l'état dans la sphère de Bloch, l'état $r|0\rangle + se^{i\vartheta}|1\rangle$ est transformé en $s|0\rangle + re^{-i\vartheta}||1\rangle$, ce qui correspond

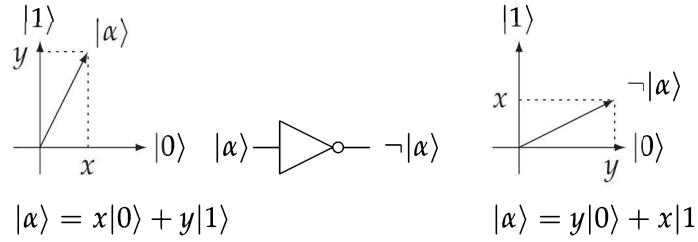


FIGURE 7.2. L'inverseur quantique échange les composantes de l'état du qubit, de telle sorte que la probabilité d'observer un 0 après l'inversion est égale à la probabilité d'observer un 1 avant l'inversion.

à une symétrie relativement à la diagonale du plan horizontal sur la figure de l'encadré 7.1 page 153.

La porte d'Hadamard

La porte d'Hadamard n'a pas d'équivalent en porte logique classique. La particularité du calcul quantique repose pour beaucoup sur elle. Elle transforme la valeur $|\alpha\rangle = x|0\rangle + y|1\rangle$ en $H(|\alpha\rangle) = \frac{x+y}{\sqrt{2}}|0\rangle + \frac{x-y}{\sqrt{2}}|1\rangle$.

L'état $|0\rangle$ est ainsi transformé en l'état $\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$ qui est la superposition équilibrée des deux états $|0\rangle$ et $|1\rangle$. Sur un registre comportant d qubits indépendants, tous dans l'état $|0\rangle$, l'application de la porte d'Hadamard à chacun d'entre eux conduit à un état qui est la superposition équilibrée des 2^d états possibles. Cette propriété remarquable revient à considérer que le registre se trouve virtuellement dans tous les états, autorisant la réalisation simultanée des calculs avec toutes les valeurs.

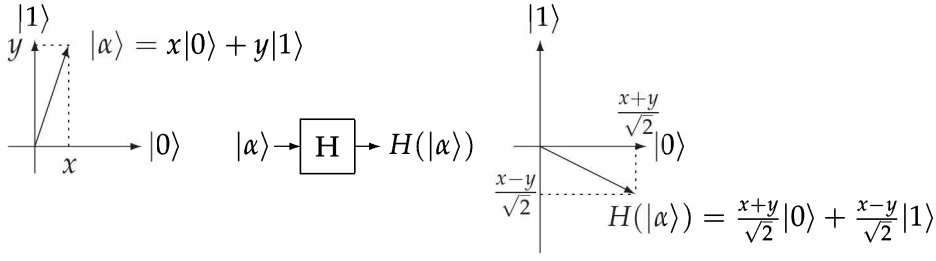


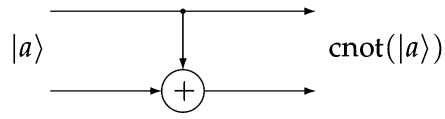
FIGURE 7.2. La porte d'Hadamard. Selon l'illustration simplifiée de son effet dans le plan réel, elle réalise une symétrie par rapport à la diagonale d'angle $\pi/8$, transformant l'état $|0\rangle$ en l'état $\frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$, qui est la superposition équilibrée des deux états $|0\rangle$ et $|1\rangle$.

L'inverseur contrôlé

Tout ce qui est calculable peut l'être en numération binaire, et il est possible de réaliser ces calculs à partir d'un assemblage de portes logiques de deux types seulement : la porte *ou exclusif* et la porte *et*. Ces portes logiques ont un équivalent quantique qui sont respectivement l'inversion contrôlée et la porte de Toffoli. Ainsi, toute fonction mathématique qui est efficacement calculable avec des portes logiques pourra l'être tout aussi efficacement avec des portes quantiques. Le calcul quantique fait au moins aussi bien que le calcul classique.

Une contrainte des portes quantiques est de réaliser des transformations inversibles qui ont par conséquent autant de sorties que d'entrées. L'inverseur contrôlé opère sur un registre de deux qubits et fournit un résultat sur deux qubits, produisant des états intriqués. Si le premier est observé avec la valeur $|0\rangle$, alors le second est inchangé, et si le premier est observé avec la valeur $|1\rangle$, alors le second est inversé. Ainsi, la valeur du deuxième qubit s'observe comme le *ou exclusif* des deux entrées.

La sortie de l'inverseur contrôlé cnot (*Controled Not*) est décrit par la transformation linéaire qui inverse les deux dernières composantes de l'état, comme indiqué sur la figure 7.5.



$$\begin{cases} |a\rangle & = & x|00\rangle + y|01\rangle + z|10\rangle + t|11\rangle \\ \text{cnot}(|a\rangle) & = & x|00\rangle + y|01\rangle + t|10\rangle + z|11\rangle \end{cases}$$

FIGURE 7.2. L'inverseur contrôlé. Les sorties sont intriquées. Si le premier qubit est dans l'état $|0\rangle$, la seconde sortie sera le même état que son entrée, et si le premier qubit est dans l'état $|1\rangle$, le second qubit sera inversé. Cette porte permet de réaliser le *ou exclusif* de la logique booléenne.

La porte de Toffoli

Cette porte quantique permet de réaliser la fonction logique *et*. Mais une porte quantique ne peut être modélisée que par des équations linéaires inversibles et, par ailleurs, la table de vérité de l'opérateur *et* n'est pas équilibrée, puisque la valeur 1 n'est atteinte que si les deux entrées sont égales à 1, soit une seule fois sur les quatre combinaisons possibles des entrées. En conséquence, cet opérateur ne peut être réalisé directement par une transformation inversible. La porte de Toffoli repose sur un principe similaire au schéma de Feistel et opère sur un registre

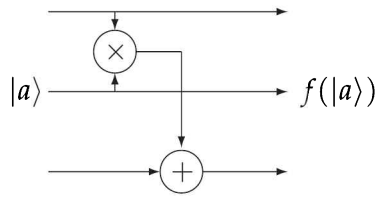


FIGURE 7.6. La porte de Toffoli. Cette réalisation d'une transformation unitaire d'état permet de réaliser le *et* de la logique booléenne. Si la troisième entrée est dans l'état déterminé $|0\rangle$, la troisième sortie sera toujours observée dans l'état $u \times v$, où u et v représentent les observations des deux premières sorties.

de trois qubits qui effectue le calcul de la fonction $f(a,b,c) = (a, b, a \oplus (b \cdot d))$. Cela revient à réaliser un inverseur contrôlé sur le troisième qubit du registre, l'inversion étant réalisée si les deux autres entrées sont observées à $|1\rangle$. Comme pour l'inverseur contrôlé, l'équation de la porte de Toffoli est obtenue en inversant les deux dernières composantes de l'état, mais cette fois, avec un état comportant trois qubits :

$$|a\rangle = s|000\rangle + t|001\rangle + u|010\rangle + v|011\rangle + w|100\rangle + x|101\rangle + y|110\rangle + z|000\rangle$$

$$f(|a\rangle) = s|000\rangle + t|001\rangle + u|010\rangle + v|011\rangle + w|100\rangle + x|101\rangle + z|110\rangle + y|000\rangle$$

2 L'algorithme de Shor pour la factorisation

Les portes quantiques présentées au paragraphe précédent permettent, en agençant les portes convenablement, de réaliser tous les calculs classiques sur un calculateur quantique, en particulier les calculs arithmétiques. Nous allons montrer maintenant comment la superposition des états et leur intrication permet de factoriser des entiers de manière efficace, c'est-à-dire avec une complexité polynomiale en leur taille.

2.1 Connaître l'ordre d'un entier modulo *n* permet souvent de factoriser *n*

Lorsqu'on calcule les puissances successives d'un entier modulo n qui est premier avec n, on revient immanquablement à 1. Prenons par exemple la suite des puissances successives de 2 modulo 21, chaque terme étant obtenu en multipliant le précédent par 2 modulo 21 :

$$1 \longrightarrow 2 \longrightarrow 4 \longrightarrow 8 \longrightarrow 16 \longrightarrow 11 \longrightarrow 1 \cdots$$

L'ordre d'un entier x modulo n est le premier exposant r non nul pour lequel x^r est congru à 1 modulo n. Ainsi, l'ordre de 2 modulo 21 vaut-il 6. Si l'exposant r

est pair, de l'égalité $x^r = 1$ on déduit que $(x^{r/2})^2 = 1$. Ainsi, l'entier y, égal à $x^{r/2}$, est-il une racine carrée de 1 modulo n. Si par chance cette racine carrée n'est pas -1, nous avons vu au paragraphe 2.4, page 51, qu'on pouvait en déduire un facteur de n en calculant $\text{pgcd}(n,y-1)$ ou $\text{pgcd}(n,y+1)$. Dans notre exemple, on a $\text{pgcd}(21,8-1) = 7$ et $\text{pgcd}(21,8+1) = 3$.

Factoriser l'entier n peut donc se faire en cherchant l'ordre d'un entier x modulo n. Si les conditions requises ne sont pas satisfaites pour obtenir la factorisation, par exemple si l'ordre r est impair, ou si $x^{r/2}$ vaut -1, alors il suffit de recommencer avec une autre valeur de x. En moyenne, une valeur sur deux permet de conclure, et la factorisation de l'entier n est rapidement atteinte.

2.2 Trouver l'ordre efficacement par un calcul quantique

Trouver l'ordre d'un entier modulo n est le cœur de l'algorithme quantique de Shor pour la factorisation. Il utilise deux registres quantiques dont les tailles sont sensiblement supérieures au nombre de chiffres binaires requis pour représenter l'entier n à factoriser. Nous illustrerons l'algorithme en factorisant $n = 21$, dont l'écriture binaire est 10101 avec deux registres quantiques de 7 qubits chacun.

Le premier registre, noté r, représentera des exposants. L'autre registre, noté y, contiendra les résultats du calcul. Notons k le nombre de qubits de chaque registre. Dans notre exemple, on a $k = 7$.

Les étapes du calcul sont les suivantes :

1. *Initialisation.* Le registre r reçoit toutes ses entrées à l'état $|0\rangle$, et le registre y reçoit une entrée qui représente l'entier 1 selon un codage convenu.
2. *Superposition des exposants.* Des portes d'Hadamard sont appliquées à chaque qubit du registre r, de telle sorte que son état devienne la superposition équilibrée de toutes les valeurs possibles comprises entre 0 et $2^k - 1$.
3. *Élévation simultanée à la puissance.* Le registre y est soumis à un assemblage de portes quantiques élaboré pour effectuer le calcul $y \times x^r \mod n$. Ce calcul étant efficacement réalisable en logique classique, il l'est également avec des portes quantiques. L'entier x est celui dont l'ordre sera déterminé modulo n. Il est codé, ainsi que l'entier n, dans l'assemblage des portes. Le registre y contient maintenant la superposition des valeurs de $y \times x^r$ pour toutes les valeurs de r. Si nous avons choisi la valeur $x = 2$ pour notre assemblage de portes, le registre y contient maintenant la superposition, sensiblement équilibrée des valeurs 1, 2, 4, 8, 16 et 11, qui sont les puissances successives de 2 modulo n.
4. *Observation d'un résultat.* La valeur du registre y est observée. Les deux registres r et y étant intriqués, l'état du registre r est maintenant projeté

sur l'ensemble des exposants qui conduisent à la valeur d'observation du registre y. Ainsi, le registre r contient maintenant la superposition des états correspondant aux exposants r tels que yx^r est congru à l'observation modulo n. Supposons avoir observé la valeur 8, le registre r contient alors la superposition sensiblement équilibrée des valeurs 3, 9, 15, 21, 27, 33, 39, 45, 51, 57, 63, 69, 75, 81, 87, 93, 99, 105 et 116. Ces valeurs sont régulièrement espacées avec une période égale précisément à l'ordre de 2 modulo 21.

5. *Transformation de Fourier.* Maintenant, le registre y ne sera plus utilisé. Pour déterminer la période de la distribution des états dans le registre r, on l'applique à un assemblage de portes quantiques qui réalise la transformation de Fourier. Il existe un algorithme classique qui réalise efficacement ce calcul, appelé *transformation de Fourier rapide*, et il se décline tout aussi efficacement en un calcul quantique. Le registre r contient maintenant le spectre de Fourier de la fonction périodique des exposants. Dans notre exemple, il contient 6 raies spectrales, centrées autour des valeurs 0, 21, 43, 64, 85 et 107. L'observation du registre r conduira avec une très forte probabilité à l'une de ces valeurs.

Encadré 7.3. Spectre de Fourier.

L'analyse de Fourier consiste à exprimer un signal périodique à l'aide de fonctions sinusoïdales. Cette analyse fait apparaître une somme de plusieurs termes, de différentes fréquences : la fondamentale dont la période est celle du signal, et certains multiples de cette fréquence fondamentale, qui sont appelés les harmoniques.

L'ensemble des fréquences qui entrent dans cette décomposition du signal s'appelle le *spectre de Fourier* du signal. Chaque fréquence du spectre est représentée par une *raie spectrale*. La transformation de Fourier est la fonction qui exprime le spectre à partir du signal. La découverte en 1965, par les Américains James Cooley et John Tukey, d'un algorithme rapide pour le calcul du spectre de Fourier a permis d'accomplir des progrès considérables dans le traitement des signaux.

La transformation de Fourier est une transformation linéaire unitaire qui est par conséquent réalisable par un assemblage de portes quantiques. Cet algorithme est utilisé ici pour trouver la période d'un signal en quelques observations seulement, sans avoir à explorer exhaustivement toutes les valeurs. L'intérêt du modèle quantique est qu'un registre superpose un nombre exponentiel de valeurs du spectre avec une complexité *quasi* linéaire.

6. *Détermination de l'ordre.* Les six valeurs des raies spectrales sont également réparties sur l'intervalle de 0 à 127 et correspondent à la valeur entière qui approche au mieux les fractions $\frac{0}{6}, \frac{1}{6}, \frac{2}{6}, \frac{3}{6}, \frac{4}{6}$ et $\frac{5}{6}$. Supposons par exemple que nous ayons observé la valeur 107, le dénominateur 6 apparaîtra par un

développement en fractions continues du quotient $\frac{107}{128}$. Le développement fait apparaître les approximations successives suivantes : $\frac{1}{1}, \frac{5}{6}, \frac{51}{61}$ et $\frac{107}{128}$. Le dénominateur 6 est la plus grande valeur inférieure à l'entier 21 qu'on cherche à factoriser et correspond à l'ordre cherché. Selon la raie spectrale observée, on peut tomber sur un diviseur de l'ordre. Il faut alors recommencer l'observation et retenir le plus petit multiple commun à tous les résultats observés.

Encadré 7.4. Les fractions continues.

Tout nombre réel se décompose en une partie entière et une partie décimale. Ainsi, le nombre $\pi \approx 3{,}141592653\cdots$ se décompose-t-il en :

$$\pi = \underbrace{3}_{\text{partie entière}} + \underbrace{0{,}141592653\cdots}_{\text{partie décimale}}$$

La partie décimale est par définition strictement inférieure à 1, elle est donc l'inverse d'un nombre réel strictement supérieur à 1 :

$$\pi = 3 + \frac{1}{7{,}0625132\cdots}$$

Le processus de décomposition en partie entière et partie décimale peut se poursuivre sur le dénominateur $7{,}0625132\cdots$ En continuant ainsi, on obtient une écriture du nombre π appelée *développement en fractions continues* :

$$\pi = 3 + \cfrac{1}{7 + \cfrac{1}{15 + \cfrac{1}{1 + \cfrac{1}{292 + \cdots}}}}$$

Lorsqu'on tronque ce développement pour ne conserver que les premiers termes, on obtient une suite de fractions. Pour le nombre π, cette suite de fractions est :

$$\frac{3}{1}, \frac{22}{7}, \frac{333}{106}, \frac{355}{113}\cdots$$

On y reconnaît les approximations bien connues du nombre π par des fractions. Une fraction de cette suite est la meilleure approximation possible du nombre duquel on est parti, ici le nombre π, en ce sens qu'il n'existe aucune fraction qui soit à la fois plus proche de ce nombre et plus simple. Toute fraction qui approche mieux le nombre π aura un dénominateur plus grand. Cette propriété des fractions continues est utilisée ici pour déterminer le nombre de raies dans un spectre dont on n'en observe qu'une seule.

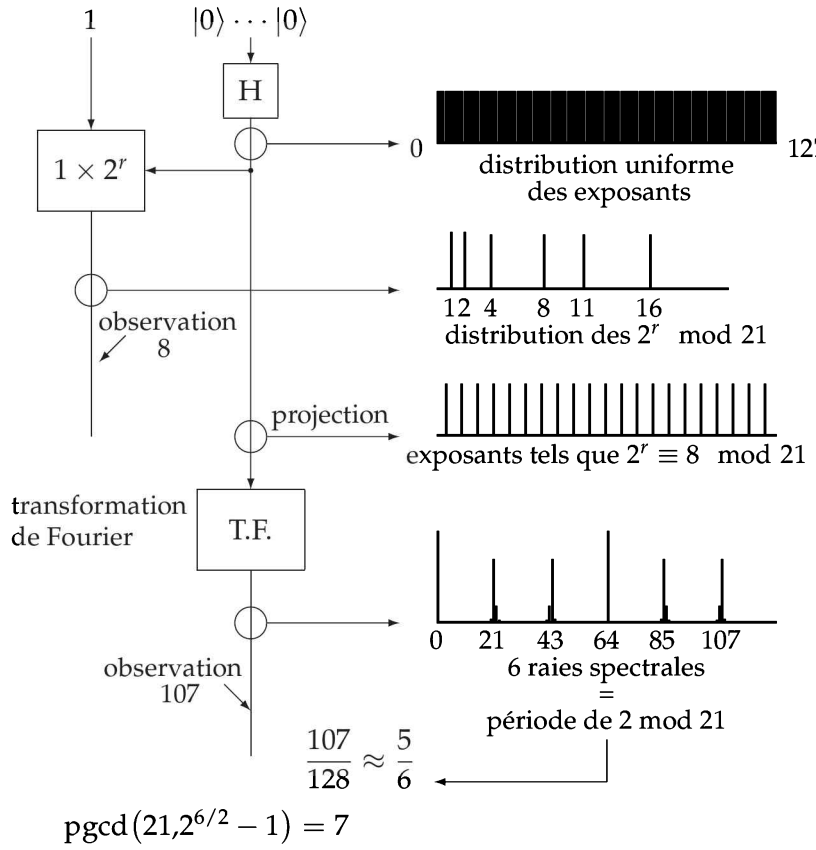


FIGURE 7.1. Algorithme de Shor pour la factorisation. La réalisation est schématisée pour factoriser l'entier 21 à partir du calcul de l'ordre de 2 modulo 21. Elle utilise deux registres quantiques de sept qubits chacun. Les histogrammes décrivent l'état de chaque registre. Ils représentent la probabilité d'observer une valeur d'un registre.

Une fois l'ordre de x modulo n connu, la factorisation s'en déduit comme exposé au paragraphe précédent. Les valeurs de x et n ne sont pas des paramètres de cet algorithme. Pour déterminer l'ordre d'un autre entier x modulo n, ou pour changer la valeur de n, il faudra, à l'étape 3, fabriquer un autre agencement des portes quantiques qui calculent $x \times y^r \mod n$. Pour cela, la dénomination *ordinateur quantique* est inappropriée, le terme *ordinateur* désignant un dispositif programmable. Le calcul quantique se compare davantage aux calculateurs analogiques qui, dans les années 1950 et 1960, permettaient de résoudre des équations différentielles par réalisation d'un circuit électronique obéissant aux mêmes équations et par observation du comportement de ce circuit.

2 La cryptographie quantique

La construction d'un calculateur quantique est actuellement hors de portée de nos moyens techniques. Mais si un tel calculateur voyait le jour, cela remettrait en cause la plupart des mécanismes à clé publique qui permettent aujourd'hui d'échanger des clés de chiffrement symétrique. La physique quantique propose une solution alternative, dénommée abusivement *cryptographie quantique*. Elle réalise la même fonction que l'échange Diffie-Hellman. Elle permet à deux correspondants de partager une clé secrète commune avec la certitude que personne d'autre n'aura accès à la moindre information sur ce secret partagé. La sécurité de cet échange repose sur les lois de la physique, et non pas sur la difficulté supposée de certains problèmes mathématiques comme dans la cryptographie à clé publique. Contrairement au calcul quantique, cette solution est aujourd'hui effective. Il existe des équipements commerciaux qui réalisent ce qu'il conviendrait plutôt d'appeler une *distribution quantique de clé*.

Rappelons que ce problème n'a pas de solution certaine. Un adversaire qui intercepte un échange peut toujours choisir au hasard la clé commune et compter sur sa chance. L'efficacité d'un tel protocole ne peut être évaluée que du point de vue de la théorie de l'information, et une formulation de la question posée est :

> « *Lors de la réalisation d'une expérience d'interception, quelle est la quantité d'information que peut espérer acquérir un adversaire ?* »

Le résultat quantitatif de cette question est une entropie mesurant l'état d'incertitude de l'adversaire.

En cryptographie classique, on cherche à rendre impossible le calcul de l'acquisition d'information par l'adversaire lorsqu'il observe les échanges entre deux protagonistes. En cryptographie quantique, on s'assure que l'adversaire ne dispose d'aucune information sur la donnée qui est parvenue au destinataire. Si un adversaire a pu intercepter certaines données, alors les lois de la physique quantique énoncent que cette interception peut être détectée, et les correspondants pourront distiller leur secret, en extraire une essence, afin de s'assurer qu'un éventuel adversaire ne dispose plus d'aucune information, que son incertitude est totale, sur la donnée qui résulte de la distillation.

L'échange de clé quantique requiert deux canaux de transmission :

- un canal quantique ;
- un canal public et intègre.

Le canal quantique est celui par lequel l'information est transmise. Chaque unité d'information est portée par une particule unique. On sait réaliser aujourd'hui des transmissions de photons uniques sur environ trente kilomètres à l'air libre, ou sur une centaine de kilomètres en utilisant des fibres optiques. En raison

de l'impossibilité de cloner les photons, qui est le principe sur lequel repose la sécurité, il est impossible d'inclure des répéteurs qui permettraient d'augmenter les distances de transmission.

Le canal public, lui, requiert absolument d'être intègre. L'information échangée sur ce canal peut sans problème être connue de tous, mais elle ne doit absolument pas pouvoir être changée aux dépens des correspondants. Ce canal peut consister par exemple en une liaison radio en ondes longues et de forte puissance. Il est pratiquement impossible de brouiller ou de modifier de tels signaux. Ce canal sert à échanger des informations publiques qui permettront aux deux correspondants de s'accorder sur leur secret partagé.

Le protocole présenté ci-après est une réalisation du protocole dit BB84, du nom de ses concepteurs, les chercheurs canadiens Gilles Brassard et Charles Bennet, et de l'année 1984 de son invention. Cet exemple de réalisation utilise des photons polarisés. L'ensemble du protocole comprend cinq étapes. Les deux acteurs de ce protocole sont un émetteur et un récepteur.

1. *Transmission de l'information*

   Pour transmettre son information, l'émetteur utilise deux bases. L'une notée $+$ est constituée d'un vecteur horizontal et d'un vecteur vertical, et l'autre, notée $\times$, est constituée de deux vecteurs diagonaux à 45 degrés des précédents. Ces deux bases sont conjuguées, ce qui signifie qu'une information représentée dans une base est détectée de façon aléatoire lorsqu'elle est observée dans l'autre base.

   Pour transmettre un 0 dans la base $+$, il transmet un photon polarisé ↔

   Pour transmettre un 1 dans la base $+$, il transmet un photon polarisé ↕

   Pour transmettre un 0 dans la base $\times$, il transmet un photon polarisé ⤢

   Pour transmettre un 1 dans la base $\times$, il transmet un photon polarisé ⤡

2. *Réception de l'information*

   Le récepteur va essayer de lire ce que l'émetteur lui a transmis. Il va lui aussi choisir au hasard la base $+$ ou $\times$ pour observer ce qu'il reçoit.

   Si le récepteur observe le photon dans la même base que celle utilisée à l'émission, alors il reçoit l'information sans erreur, ou du moins avec une erreur résiduelle très faible.

   Si le récepteur observe le photon dans l'autre base, alors l'information qu'il reçoit est aléatoire et indépendante de ce que l'émetteur a voulu lui transmettre.

Encadré 7.5. Stratégie de l'adversaire.

Si un adversaire cherche à intercepter un échange quantique, il ne peut qu'intercepter un photon dans sa totalité. Sa meilleure stratégie est d'observer l'information en choisissant une des bases de mesure au hasard, comme le fait le destinataire : soit $+$, soit $\times$, puis de renvoyer l'information qu'il a détectée. Une fois sur deux, il aura choisi la même base que l'émetteur et alors il aura retransmis correctement l'information. Mais dans l'autre moitié des cas, en choisissant une base différente, son observation sera indépendante de ce qu'il aura reçu. L'adversaire ne peut espérer connaître plus des trois quarts de l'information partagée par les correspondants.

3. *Publication des bases de codage*

   Sur le canal intègre, l'émetteur publie les bases dans lesquelles il a transmis l'information.

   De même, le récepteur publie sur le même canal intègre les bases qu'il a utilisées à la réception.

   À la suite de cet échange, les deux correspondants savent quelles sont les informations pertinentes et jettent celles qui ont été traitées dans des bases différentes à l'émission et à la réception.

4. *Réconciliation*

   Les correspondants choisissent au hasard et publient des numéros de symboles binaires ainsi que la somme modulo 2 des symboles qui portent ces numéros. À chaque fois qu'un symbole binaire est ainsi révélé, un des termes de la somme est éliminé de manière à maintenir totale l'incertitude d'un adversaire sur les symboles restants.

   Cette étape permet d'évaluer la quantité d'information acquise par un éventuel adversaire. Si les correspondants sont d'accord sur toutes les valeurs publiées, c'est que l'adversaire n'a acquis aucune information sur le secret partagé, puisque son action provoque forcément des erreurs. Et si certaines valeurs ne coïncident pas, c'est qu'un adversaire a intercepté certains photons pour les retransmettre. La proportion de valeurs qui ne coïncident pas permet d'évaluer l'information acquise par l'adversaire sur la donnée partagée entre les correspondants.

5. *Distillation*

   Les correspondants distillent alors leur information en la réduisant. Pour cela, ils choisissent une fonction au hasard dans une famille universelle (voir encadré 6.7, page 142), ils publient cette fonction, puis l'appliquent à la séquence d'information qui leur reste. À l'issue de cette distillation, la quantité résiduelle d'information d'un éventuel adversaire sur le secret partagé est réduite à moins d'un bit.

Cette étape est également complétée par l'application d'un algorithme de correction d'erreurs afin d'éliminer les erreurs sur la donnée partagée entre les deux correspondants.

Si un adversaire intercepte les photons transmis pour les observer, il ne saura reproduire le même photon que celui envoyé par l'émetteur que s'il a choisi la même base de codage que lui. Cela ne peut survenir que dans un cas sur deux en moyenne. Dans le cas contraire, ce qu'il renvoie est totalement décorrélé de ce qui a été envoyé. Un adversaire qui intercepte l'échange provoque donc en moyenne un taux d'erreur de 25 %.

Encadré 7.6. Le théorème de distillation.

La quantification de l'information résiduelle de l'adversaire lors de l'échange quantique repose sur le théorème de distillation. Si X est un vecteur aléatoire binaire de dimension n, si G est un élément aléatoire d'une famille universelle qui réduit n symboles binaires à r, alors l'incertitude de Shannon sur le vecteur $G(X)$, connaissant la fonction G, satisfait la minoration suivante :

$$H\big(G(X) \mid G\big) \geq r - \frac{2^{r-R(X)}}{\ln(2)}$$

où $R(X)$ désigne l'entropie de Rényi du vecteur aléatoire X.

Supposons par exemple que deux correspondants disposent d'une information de 1 000 symboles binaires qu'ils veulent maintenir secrète. Pour tous les autres, ce secret est une variable aléatoire X entachée d'une certaine incertitude. Supposons également qu'un adversaire connaisse chaque symbole binaire de X avec une probabilité égale à $3/4$. Cette information sur X est quantifiée par une incertitude de Rényi égale à $1\,000 \times r_2(0{,}75) \approx 678$ bits. Les correspondants choisissent publiquement un élément aléatoire d'une famille universelle réduisant les 1 000 symboles binaires à 600. Ils partagent maintenant un vecteur $Y = G(X)$ réduit à 600 symboles binaires. Le théorème de distillation énonce que l'incertitude de l'adversaire sur le vecteur Y sera majoré par :

$$H(Y \mid G) \geq 600 - \frac{2^{600-678}}{\ln(2)} \approx 600 - 4 \cdot 10^{-24}$$

qui est une quantité très proche de 600 bits. L'incertitude de l'adversaire sur Y est quasi totale.

La sécurité de ce protocole repose de manière cruciale sur les propriétés des deux canaux de transmission. L'information ne doit être portée que par des photons uniques. Si elle l'était par deux photons, il n'est pas exclu qu'un dispositif de mesure permette d'acquérir leur état. De même, les informations de réconciliation sont publiques, mais le canal sur lequel elles sont transmises doit absolument être intègre. Si cela n'était pas le cas, un intrus entre les deux correspondants pourrait négocier le secret avec chacun des partenaires. Certaines réalisations utilisent une

même fibre optique pour les deux canaux. Pour assurer l'intégrité, les échanges sont signés numériquement. La propriété s'appuie alors sur la cryptographie à clé publique classique, ce qui remet en question l'intérêt de tels dispositifs. Les hypothèses qui permettent d'établir la sécurité inconditionnelle de l'échange quantique ne sont alors pas rigoureusement satisfaites.

4 En conclusion

La cryptographie quantique permet le partage d'une clé entre deux interlocuteurs particuliers, disposant d'une liaison spécifique capable de transmettre entre eux des photons uniques. Cette exigence d'unicité des photons transmis interdit tout dispositif d'amplification et impose aujourd'hui une limite physique à la distance de transmission. La recherche a posé les principes des *relais quantiques* qui pourraient réémettre un photon sans avoir accès à son état. Mais dans un avenir immédiat, il ne s'agit aucunement d'un procédé concurrent des systèmes à clé publique qui opèrent sur un réseau ouvert.

Le calcul quantique, quant à lui, pose des problèmes techniques qui semblent insurmontables dès qu'il s'agit de rendre cohérents des registres de quelques dizaines de qubits, et il est peu probable que cela constitue une méthode applicable pour factoriser des entiers de grande taille dans un avenir proche. Le problème à surmonter est l'interaction des particules avec l'environnement qui détruit l'information portée par les qubits intriqués d'un registre. Un calculateur consomme l'essentiel de ses ressources à la correction des erreurs dues à ces interactions.

Ces travaux ont cependant un impact pour construire des alternatives crédibles à la cryptographie à clé publique si le modèle de calcul quantique devenait réalisable. La recherche tente de définir des procédés reposant sur des problèmes difficiles autres que la factorisation ou le calcul des logarithmes et qui résisteraient à un calculateur quantique.

La nature de la cryptologie

Au terme de cette exploration de différents aspects de la cryptologie, en guise de conclusion, interrogeons-nous sur la nature de cette activité.

Une activité sociale

La cryptologie s'est développée pour répondre à des besoins manifestes de l'activité humaine, comme la préservation du caractère privé de certains échanges, ou l'authentification du correspondant. Tous les dispositifs que nous portons sur nous et qui réalisent des opérations cryptologiques rendent *in fine* un service à des personnes qui veulent protéger des informations transmises dans le cadre de relations sociales, que ces relations s'inscrivent dans la sphère publique ou privée. Citons David Kahn, figure incontournable de l'histoire de cette discipline :

> *La cryptologie est par définition une activité sociale, et peut être ainsi examinée d'un point de vue sociologique. Elle est une communication secrète, et la communication est sans doute l'activité humaine la plus variée et la plus complexe. Elle ne comprend pas seulement les mots, mais les gestes, les expressions du visage, le ton de la voix, et même le silence. (...)*
>
> *La rétention d'information constitue l'élément essentiel de ce qu'on appelle le « secret ». Toutes les manifestations du secret, le camouflage, le déguisement, les portes verrouillées, ont en commun l'idée de base de ne pas communiquer d'information. Sa forme extrême est le silence (mis en défaut dans le cauchemar orwellien dans sa forme extrême d'espionnage – la détection et l'interprétation des ondes cérébrales). Une investigation exhaustive du concept de secret demanderait de comprendre tous les aspects du comportement culturel, (...) parce que le secret est l'antithèse de la communication, et la communication fait de l'homme un être social. La cryptographie combine cette antithèse en une seule opération ; on pourrait la définir en un mot comme une « communication non communicante ».*
>
> [8] David Kahn, *The Codebreakers*, p. 752

Mathématiques et cryptologie

Si, aujourd'hui, la cryptologie fait massivement appel à de nombreux champs des mathématiques, jusqu'à faire figure d'une branche des applications de cette discipline, il n'en a pas toujours été ainsi. Elle a longtemps été pratiquée comme une manipulation du langage, inscrite dans des jeux d'écriture. Les seules techniques apparentées aux mathématiques se sont longtemps limitées au comptage pour le décryptement, introduit par les sciences arabes dès le IX^{e} siècle de notre ère.

L'utilisation explicite du calcul et des structures mathématiques, en particulier algébriques, en cryptologie semble avoir débuté avec le chiffre de Lester Hill en 1929, qui utilise des matrices modulo 26, initiant un mouvement qui a conduit jusqu'à la situation actuelle, qui place la cryptologie au cœur des mathématiques appliquées.

Le mouvement entre les deux disciplines, mathématique et cryptologie, s'est opéré dans les deux sens. La cryptologie emprunte aux mathématiques au point qu'on peut se demander s'il existe un domaine des mathématiques qui ne sera pas un jour exploité par la cryptologie. Les systèmes à clés publiques comme le RSA utilisent la théorie des nombres, mais aussi la géométrie algébrique avec les courbes elliptiques, domaines principalement développés aux XIX^{e} et XX^{e} siècles. Mais réciproquement, la cryptologie fournit également des sujets aux mathématiciens, jusqu'à justifier aujourd'hui des pans entiers de leur activité de recherche. Certains problèmes, comme la factorisation des entiers, le logarithme dans les groupes finis, ou le comptage des points sur les courbes elliptiques, ont connu un regain d'intérêt considérable en raison de leur position centrale en cryptologie.

Avec le développement de la théorie cryptologique, présentée au chapitre 6, nous assistons même à l'émergence d'une nouvelle discipline de nature mathématique, tout comme la théorie des probabilités s'est imposée à partir des années 1930 dans le prolongement du calcul des probabilités.

Un art ou une science ?

Mais le développement de la théorie cryptographique, en introduisant la notion de *sécurité prouvable* et de *preuve de sécurité*, pose-t-il vraiment les jalons de ce qu'il conviendrait d'appeler une *science cryptographique* ? Cette nouvelle discipline constitue-t-elle la nouvelle face de la cryptologie ? Certains cryptologues revendiquent avec force le caractère scientifique de leur activité. Les auteurs Jonathan Katz et Yehuda Lindell écrivent en 2007 dans la préface de leur ouvrage *Introduction à la cryptographie moderne* :

> « *(...) les constructions cryptographiques peuvent être prouvées sûres à l'égard d'une définition de la sécurité clairement énoncée et relativement à une hypothèse cryptographique bien définie. Ceci est l'essence de la cryptographie moderne, et qui a changé la cryptographie d'art en science.* ».

Cette affirmation d'une cryptographie devenue science est à rapprocher de la pureté revendiquée au XIX^e^ siècle par certains mathématiciens comme Cayley, fondateur de l'école britannique moderne des mathématiques pures, à une époque où, précisément, les mathématiques appliquées connaissent un développement considérable, et prennent une part de plus en plus importante dans la formation des ingénieurs. Une telle opposition entre le récit d'une discipline idéalisée et le mouvement de la réalité des faits constitue une *inversion narrative.*

Le terme de *preuve de sécurité* est fortement remis en question par d'autres auteurs comme Ann Hibner Koblitz, Neal Koblitz et Alfred Menezes qui insistent sur le fait que l'objet de la cryptologie est de construire des dispositifs performants devant apporter garantie et confiance aux personnes qui les utilisent. Établir la preuve de l'authenticité d'un message, ou assurer que les informations qu'il contient n'ont pas été disséminées ne se réduit pas à appliquer un théorème, mais s'appuie sur une expertise et sur le développement d'arguments contradictoires. Sans nier l'importance des résultats apportés par cette nouvelle théorie cryptographique, ces auteurs doutent de la sécurité réelle d'un procédé qui ne serait assorti que d'une *preuve de sécurité* de nature mathématique. Ils insistent sur l'ingéniosité déployée par les adversaires pour compromettre la sécurité de procédés pourtant prouvés sûrs. Dans un article *Courbes elliptiques : Le parcours tortueux d'un changement de paradigme* paru en 2008, ils écrivent :

> *Une des raisons pour laquelle la cryptographie comporte un élément subjectif si fort est que la spéculation en est un élément central. Quand on décide du type de cryptographie à utiliser (...), lorsqu'on choisit le type de protocole pour une application donnée (...), on doit faire un pari sur les développements futurs pour évaluer les problèmes fondamentaux de sécurité du système. On doit se demander : quel type d'adversaire aurai-je des chances de rencontrer, et quel sera probablement son meilleur angle d'attaque ? Y aura-il des progrès pour réduire le temps de résolution d'un problème qu'on suppose aujourd'hui insoluble ? Le calcul quantique deviendra-t-il praticable ? Quelle nouvelle attaque par canaux secondaires sera inventée ?*

Ces auteurs voient une inversion narrative dans la revendication appuyée d'une cryptologie devenant science :

> *Peut-être est-ce à cause de ces éléments fortement contingents dans ce domaine que les chercheurs ressentent de plus en plus le besoin de quitter*

leur domaine pour assurer au public que [la cryptologie] est en train de devenir une science, que des garanties absolues de sécurité peuvent être données (« sécurité prouvable »), et que les cryptographes suivent fidèlement le modèle idéal [de recherche et de développement de la science].

La confiance des acteurs sociaux dans le système d'échanges numériques reste l'objectif essentiel de la cryptographie. L'établissement de cette confiance s'alimente de preuves mathématiques de sécurité sans pouvoir se passer d'études de résistance menées par les travaux de cryptanalystes ingénieux, construites avec habileté et tâtonnements.

Autres implications scientifiques et techniques

Les calculs cryptographiques ne sont plus faits manuellement. Ils sont aujourd'hui réalisés dans des calculateurs ou des dispositifs spécialement conçus pour cet usage. Pour cette raison, la cryptologie investit l'informatique et l'électronique numérique pour la conception de calculateurs performants. L'évolution récente de cette discipline fait aussi largement appel aux sciences physiques. Un calcul ne peut plus se concevoir sans prendre en compte le dispositif qui l'exécute. Les nouvelles attaques de nature physique sur les canaux secondaires que sont la mesure de la consommation, du rayonnement électromagnétique ou du temps de calcul, exploitent les caractéristiques matérielles et techniques du dispositif.

Si le calculateur quantique voyait le jour, peut-être serions-nous contraints d'utiliser la réponse quantique au problème de l'échange de clé. Si les progrès de la théorie de la calculabilité faisaient disparaître certains mondes d'Impagliazzo, peut-être serions-nous contraints de revenir au procédé inconditionnellement sûr de Vernam. Même si son avenir suscite de nombreuses interrogations et à moins d'un bouleversement improbable dans les directions qui viennent d'être évoquées, en ce début du XXI^{e} siècle, la cryptologie s'installe durablement comme une discipline au carrefour entre sciences, société et industrie.

Solutions

Avant-propos : les situations qui font intervenir un calcul cryptologique dans le récit de l'avant-propos sont dans l'ordre :

1. Authentification de l'abonné et chiffrement de la parole en téléphonie mobile.
2. Accès des facteurs parisiens aux immeubles à l'aide d'un badge qui les authentifie et autorise l'accès pendant l'horaire prévu de leur tournée.
3. Passage des tourniquets du métro avec le passe Navigo.
4. Accès aux locaux protégés à l'aide d'un badge d'accès.
5. Démarrage des véhicules après authentification de la puce incluse dans la clé de contact.
6. Chargement et dépenses effectués avec une carte de paiement Moneo.
7. Sortie d'un local protégé à l'aide d'un badge.
8. Location d'un Vélib avec la carte Navigo.
9. Signature numérique des applications téléchargées sur les téléphones mobiles.
10. Chiffrement des données médicales confidentielles avec la carte Vitale.
11. Protection des transactions effectuées avec une carte bancaire.
12. Signature numérique des mises à jour des systèmes d'exploitation des ordinateurs.
13. Signature des déclarations de revenus effectuées sur internet.
14. Sécurisation du commerce en ligne sur internet.
15. Accès conditionnel aux programmes audiovisuels payants.

page 2 : Le texte en clair qu'aurait pu transmettre Jules César à son intendant est :

> *Iis Gallis qui proxime hostes fuerant magnas pecunias imperavi populo romano, quas intra decem dies accipies. M. Tullio aes alienum solvere poteris emereque a P. Valerio villam ejus centum milibus nummum, id est pretium quod convenerat.*
>
> J'ai réclamé pour le peuple romain, à ces Gaulois qui étaient très récemment encore nos ennemis, de fortes sommes d'argent, que tu recevras sous dix jours. Tu pourras régler ma dette à Marcus Tullius, et acheter à Publius Valerius sa maison au prix de cent mille sesterces, comme il avait été convenu.
>
> D'après Suzanne Fleixas.

Il était certainement préférable qu'on ne sache pas que Jules César a détourné le tribut des populations soumises pour régler ses dettes personnelles.

page 13 : Pars vite et reviens tard. (Titre d'un roman de Fred Vargas)

page 85 : *A good glass in the bishop's hotel in the devil's seat forty-one degrees and thirteen minutes north-east side shoot from the left eye of the death's-head a bee-line from the tree through the shot fifty feet out*

Un bon verre dans l'hôtel de l'évêque dans la chaise du diable quarante et un degrés et treize minutes nord-est quart de nord principale tige septième branche côté est lâchez de l'œil gauche de la tête de mort une ligne d'abeille de l'arbre à travers la balle cinquante pieds au large.

Traduction Charles Baudelaire.

page 87 : *Dear uncle,*

Nothing is so easy as to perform what you perfectly understand, and when you know how, it will be equally easy to decipher this, in the mean time it will puzzle your brain-box ever. Your affectionate nephew,

Henry.

Cher oncle,

Rien n'est plus facile à réaliser que ce que vous comprenez parfaitement, et quand vous savez comment faire, il sera tout aussi facile de déchiffrer cela, en même temps cela déroutera toujours votre cerveau. Votre neveu affectueux,

Henry.

page 88 :

Comme je descendais des Fleuves impassibles,
Je ne me sentis plus guidé par les haleurs :
Des Peaux-Rouges criards les avaient pris pour cibles,
Les ayant cloués nus aux poteaux de couleurs.

Extrait de *Le bâteau ivre*, Arthur Rimbaud.

page 92 :

De la musique avant toute chose,
et pour cela préfère l'impair.

Extrait de *L'art poétique*, Paul Verlaine.

Bibliographie

[1] L'art du secret. Dossier *Pour la Science*, juillet/oct 2002.

[2] Les codes secrets, 3000 ans de cryptologie. Dossier Hors-Série n° 53, *Science & Vie Junior*, juillet 2003.

[3] Cryptographie : vos secrets sont-ils bien gardés ? état des lieux... *MISC* Hors-Série n° 5, avril/mai 2012.

[4] De Jules César à Enigma, Codes et langages secrets. *Les cahiers de Science & Vie*, novembre 2012.

[5] P. BARTHÉLÉMY, R. ROLLAND et P. VÉRON : *Cryptographie, principes et mise en œuvre.* Hermès, Paris, 2e édition, mai 2012.

[6] D. BROWN : *Forteresse digitale.* J.-C. Lattès, 2007.

[7] P. GUILLOT : *Courbes elliptiques, une présentation élémentaire pour la cryptographie.* Hermès, 2010.

[8] D. KAHN : *The Codebreakers, The Story of Secret Writing.* Scribner, 1996.

[9] B. MARTIN : *Codage, cryptologie et applications.* Presses polytechniques et universitaires romandes, 2004.

[10] D. MÜLLER : *Les Codes secrets décryptés.* City Édition, 2007.

[11] A. POLI et P. GUILLOT : *Algèbre, confidentialité et intégrité en multimédia.* Hermès, 2009.

[12] S. SINGH et C. COQUERET : *Histoire des codes secrets : De l'Égypte des Pharaons à l'ordinateur quantique.* J.-C. Lattès, 1999.

[13] S. VAUDENAY : *La fracture cryptographique.* Focus science. Presses polytechniques et universitaires romandes, 2011.

[14] P. WRIGHT et P. GREENGRASS : *Spycatcher, l'autobiographie sincère d'un officier supérieur du Renseignement britanique.* Robert Laffont, 1987.

[15] G. ZÉMOR : *Cours de cryptographie.* Cassini, 2000.

Index