COURS DE CRYPTOGRAPHIE
ENSTA - MA 307

Françoise Levy-dit-Vehel

Année 2003-2004

Table des matières

Chapitre 1

Principes de base

Ce chapitre introduit les notions de base de la cryptographie. Nous décrivons d'abord les techniques de chiffrement utilisées jusqu'à la deuxième moitié du XXème siècle - pour cela communément regroupées sous le terme cryptographie classique. Celles-ci, bien qu'obsolètes, font appel à des notions qui sont utiles dans l'élaboration de schémas actuels. Nous présentons ensuite la terminologie et les différents concepts qui fondent cette discipline : les problèmes qu'elle résout, les différents types d'attaques considérés, les modèles de sécurité...et terminons par un exemple de chiffrement "parfait" au travers duquel nous montrons que c'est justement l'impraticabilité d'un tel modèle qui a permis d'ouvrir la voie à la cryptographie moderne.

La cryptologie regroupe deux branches : la cryptographie, qui consiste en l'élaboration de systèmes (de chiffrement, mais aussi de signature ou d'authentification - notions que nous préciserons plus loin), et la cryptanalyse, qui s'applique à les casser (i.e. à y trouver des failles de sécurité, ou à l'extrême à retrouver le secret - la clef - sur lequel ils reposent). Ces deux aspects ne peuvent être dissociés ; en effet, tout nouveau système est, dès sa publication, soumis à la critique de la communauté cryptographique. Mieux encore, un système existant depuis plusieurs années, ayant fait l'objet d'une scrutation intense, et ayant résisté à la cryptanalyse, est réputé être sûr, même s'il n'existe pas de preuve mathématique de sa sécurité.

1.1 Cryptographie classique

Nous présentons ici des techniques de chiffrement anciennes (parfois datant de plusieurs centaines d'années). Leur sécurité est dérisoire vus les moyens de calcul actuels, mais les principes sur lesquels elles reposent (substitution, transposition), sont à la base des systèmes modernes à clef secrète (comme DES par exemple).

Nous noterons dans la suite m, le message à chiffrer ; m est un mot fini sur un alphabet quelconque $\mathcal{A}$ (par exemple, $\mathcal{A} = \{\text{A},\ldots,\text{Z}\}$, mais le plus souvent, $\mathcal{A} = \{0, 1\}$, vu que tout alphabet peut être codé à partir de l'alphabet binaire). L'opération de chiffrement sera notée E. Elle prend en paramètre le message m et un secret K, appelé clef de chiffrement. Le mot $c = E_K(m)$ est le message chiffré. c est un mot sur un alphabet $\mathcal{B}$, qui peut être différent de $\mathcal{A}$. A E correspond une opération $D = D_K$ dite de déchiffrement, vérifiant $D_K(E_K(m)) = m$.

Remarquons ici que la même clef K est utilisée pour le chiffrement et le déchiffrement. On parle alors de chiffrement symétrique, ou à clef secrète (par opposition aux systèmes de chiffrement - dits asymétriques - dans lesquels une seule des deux opérations fait intervenir une clef secrète).

1.1.1 Chiffrement par substitution

La substitution monoalphabétique :

Dans ce système, $\mathcal{A} = \mathcal{B} = \{\text{A},\ldots,\text{Z}\}$. Soit π, une permutation sur $\mathcal{A}$.
L'opération de chiffrement d'un message $m = m_1\, m_2\, m_3 \ldots$, $m_i \in \mathcal{A}$, est :

$$E_\pi(m) = \pi(m_1)\,\pi(m_2)\,\pi(m_3)\ldots$$

La clef secrète est la donnée de la permutation π. Soit $c = c_1\, c_2\, c_3 \ldots$, le chiffré obtenu. De manière évidente, l'opération de déchiffrement est

$$D_\pi(c) = \pi^{-1}(c_1)\,\pi^{-1}(c_2)\,\pi^{-1}(c_3)\ldots$$

Par exemple, si le texte clair est

UNPREMIEREXEMPLEDECHIFFREMENT

et la permutation est définie par :

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
d	i	r	y	v	o	h	e	z	x	w	p	t	b	g	f	j	q	n	m	k	s	u	a	c	l

alors le texte chiffré est :

KBFQVTZVQVAVTFPVYVREZOOQVTVBM

Un cas particulier simple de ce système est le décalage de chaque lettre d'un nombre de lettres fixé (décalage pris modulo 26). En utilisant la correspondance $A \leftrightarrow 0$, $B \leftrightarrow 1$ ..., la permutation s'écrit $\pi(x) = x + K \bmod 26$, où $0 \leq K \leq 25$ est la clef secrète. En d'autres termes, on fixe un décalage K de l'alphabet, et on applique ce décalage à chaque lettre du mot à chiffrer.

Le cas $K = 3$ est connu sous le nom de *chiffre de César* (utlisé par Jules César au premier siècle avant J.C.). Par exemple, le mot "CESAR" est chiffré en "FHVDU".

Dans un tel système, il suffit de connaître le décalage pour déchiffrer les messages. Etant donné qu'il y a seulement 26 décalages possibles, on peut les essayer tous (on dit alors qu'on effectue une *recherche exhaustive* sur l'espace des clefs) jusqu'à obtenir un clair intelligible (si l'on suppose que le message clair est du Français par exemple).

Dans le cas d'une permutation quelconque π des lettres, étant donné qu'il existe 26! (soit plus de 4×10^{26}) permutations possibles, la recherche exhaustive (sur l'espace des clefs, i.e. des permutations π) est à exclure. Pour attaquer un tel schéma[1], on utilise le fait que les langages naturels sont redondants. En effet, par exemple en Français, la lettre la plus fréquente est un "e", la lettre "q" est quasiment toujours suivie d'un "u", beaucoup de digrammes sont plus fréquents que d'autres (par exemple "le", "es", "ts"...). Dans un alphabet à 26 lettres, l'information moyenne par lettre est théoriquement de $log_2(26) \simeq 4,7$ bits. Mais en fait, l'information moyenne par lettre dans un texte Français est environ 1,2 bits ; le Français a donc une redondance d'à peu près 3,5 bits par lettre.
Cette redondance permet de faire une analyse statistique de la fréquence des lettres dans le texte chiffré : en effet, comme la distribution des fréquences des lettres est préservée par substitution simple, la lettre la plus fréquente dans le chiffré correspond au chiffré de la lettre la plus fréquente de la langue du texte clair. Ensuite, on regarde la fréquence des digrammes. On en déduit les clairs possibles pour les lettres de ces groupes . . . On procède ainsi jusqu'à ce que la recherche exhaustive sur les lettres manquantes soit faisable.

On parle de substitution monoalphabétique car chaque lettre est toujours chiffrée en la même lettre. Un chiffrement un peu plus compliqué consiste à utiliser différentes substitutions sur des portions différentes de texte clair ; ainsi, le chiffré d'une lettre dépend de la position de celle-ci dans le texte clair. Ce type de substitution est appelé polyalphabétique.

La substitution polyalphabetique

Dans un tel chiffrement, la clef K est un t-uplet de permutations $(\pi_1, \ldots, \pi_t)$. Le texte clair est divisé en blocs de t lettres, et la i-ème lettre de chaque bloc, $1 \leq i \leq t$, est chiffrée avec la permutation π_i.
A titre d'exemple, nous allons décrire le système de Vigenère (XVIème siècle).
Dans ce système, un t-uplet de permutations est en fait un *mot-clef* de longueur t, qui spécifie le décalage à effectuer à chaque lettre d'un bloc de clair de t lettres. Par exemple, si le mot-clef est CODE (on a alors $t = 4$), la première lettre de chaque bloc sera décalée de 2 positions dans l'alphabet, la deuxième de 14 positions, etc (en utilisant la correspondance $A \leftrightarrow 0$, $B \leftrightarrow 1$, ..., vue plus haut). Le mot-clef est utilisé autant de fois qu'il y a de blocs de longueur t dans le clair. Par exemple, le clair

CESYSTEMEESTPEUSUR,

avec le mot-clef CODE, donnera le chiffré

ESVCUHHQGSVXRSXWWF.

Une méthode de cryptanalyse de ce schéma, basée également sur une analyse statistique de fréquence des lettres, sera étudiée lors de la première séance de travaux pratiques.

[1]Sous réserve que celui-ci soit utilisé pour chiffrer des messages intelligibles.

1.1.2 Chiffrement par transposition

Dans les schémas que nous venons de voir, les lettres du clair sont remplacées par d'autres lettres. Dans un chiffrement par transposition, les lettres du clair sont inchangées, seules leurs positions le sont. Le changement de positions est défini par une permutation π (la clef), sur un ensemble de taille $t \in \mathbb{N}^*$. Le clair est alors divisé en blocs de taille t, et π est appliquée sur chacun des blocs. Par exemple, si le clair est ABCDEFGHIJ, et la permutation $\pi = (3\,5\,4\,1\,2)$ (donc $t = 5$), on obtient le chiffré CEDABHJIFG.

Formellement, si $m = (m_1 \dots m_t)$, $m_i \in \mathcal{A}$, alors le chiffré est $c = c_1 \dots c_t$, avec $c_i = m_{\pi(i)}$, et ce pour chaque bloc de clair m de taille t. Le déchiffrement fait intervenir la permutation inverse π^{-1}.

Décrivons maintenant une cryptanalyse simple de ce schéma : on suppose ici que l'attaquant dispose de t paires distinctes de t-uplets (m_j, c_j), avec $m_j = (m_{1,j}, \dots, m_{t,j})$, et $c_j = (c_{1,j}, \dots, c_{t,j})$, $1 \le j \le t$, (autrement dit, t couples de clairs-chiffrés - ce type d'attaque est appelé attaque *à clair connu*, comme on le verra en section 1.2.2).
Les deux matrices $(t \times t)$, $M = (m_{i,j})$ et $C = (c_{i,j})$, $1 \le i, j \le t$, sont reliées par l'équation suivante :

$$C = K_\pi M,$$

où $K_\pi = (k_{i,j})$ est la matrice $(t \times t)$ associée à π par :

$$k_{i,j} = 1 \text{ si } j = \pi(i), 0 \text{ sinon.}$$

Pour retrouver la clef π, il suffit donc de résoudre le système linéaire ci-dessus, i.e. calculer

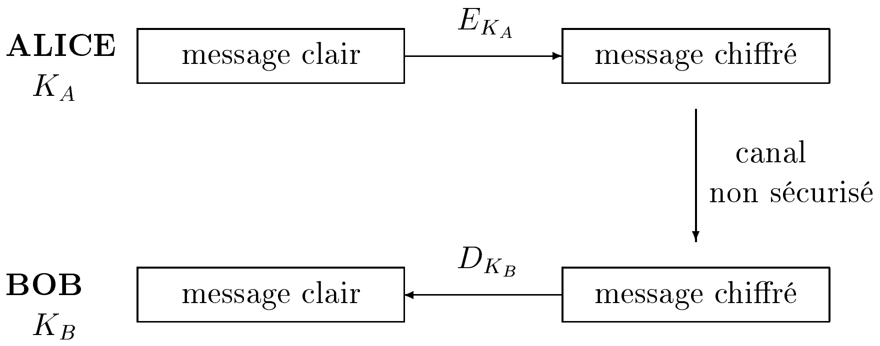
$$K_\pi = CM^{-1}.$$

Si M n'est pas inversible, l'attaquant devra utiliser d'autres couples de clairs-chiffrés.

D'une manière générale, une opération de chiffrement linéaire est vulnérable aux attaques à clair connu.

1.2 Terminologie et concepts

En cryptographie, les protagonistes d'une communication sont en général appelés Alice et Bob.
Le schéma général d'une communication chiffrée entre Alice et Bob est le suivant :



où E est l'algorithme de chiffrement, D l'algorithme de déchiffrement, et K_A, K_B sont les clefs d'Alice et de Bob respectivement. Elles vérifient :

$$D_{K_B} \circ E_{K_A} = Id.$$

Dans un schéma de chiffrement à clef secrète, $K_A = K_B$, et cette clef doit rester secrète. Dans un schéma de chiffrement à clef publique[2], E ne fait intervenir que des données publiques, K_A est une donnée publique, et K_B est secrète (ces deux clefs sont donc bien différentes). Nous aborderons les schémas cryptographiques à clef publique un peu plus tard dans ce cours.

Un schéma de chiffrement est dit *cassé* lorsqu'une entité, qui ne connait pas la clef secrète (donc autre qu'Alice ou Bob), peut retrouver le clair à partir du chiffré correspondant, en un temps en rapport avec la limite de validité de la clef[3].

1.2.1 Différents types de menaces et parades cryptographiques

Lorsqu'Alice et Bob, veulent communiquer à travers une ligne non sécurisée, ils s'exposent à divers dangers. Ces menaces sont "matérialisées" par un attaquant appelé Charlie, qui se place sur la ligne de communication entre Alice et Bob.
Charlie peut simplement écouter les communications entre Alice et Bob ; on parle d'*attaque passive*. Dans ce cas, la confidentialité des messages n'est plus assurée.
Il peut également agir sur les messages transmis : on parle *d'attaque active*.

On distingue essentiellement trois types d'attaques actives :
- **altération des messages :** Charlie peut manipuler les messages de différentes façons : modifier le contenu, mais aussi retarder (ou empêcher) la transmission, répéter l'envoi ...

[2] on dit aussi asymétrique

[3] en effet, il est inutile de casser un schéma en un temps à l'issue duquel la clef sera obsolète ; la durée de validité d'une clef dépend de son utilisation : une clef servant à protéger des secrets d'états doit avoir une durée de vie plus importante qu'une clef devant chiffrer le montant d'un achat d'une petite somme.

- **impersonification :** Charlie peut se faire passer pour Alice.
- **répudiation de messages :** cette attaque, d'un type particulier, consiste pour Alice à nier l'envoi d'un message antérieur.

L'objectif principal de la cryptographie est de proposer des parades pour ces quatre types de menaces :
- contre l'écoute des messages (i.e. pour préserver la confidentialité), on a recours à des algorithmes de *chiffrement*.
- contre l'altération des messages, on utilise des algorithmes d'*authentification* (de données).
- contre l'impersonification, on utilise des algorithmes *d'identification* (ou authentification d'entités).
- contre la répudiation, on a recours à des algorithmes de *signature*.

1.2.2 Classes d'attaques et modèles de sécurité

Supposons que les communications entre Alice et Bob soient protégées par chiffrement.
Le but de Charlie, qui est toujours sur la ligne, est de retrouver le message clair à partir du message chiffré qu'il peut écouter (il "casse" une instance du schéma de chiffrement), ou mieux retrouver la clef de chiffrement (on parle alors de cryptanalyse totale, ou "total break"). Nous supposerons toujours qu'il connaît le schéma utilisé, i.e. l'*algorithme* de chiffrement est toujours supposé public ; seule la *clef* de chiffrement est secrète. Cette hypothèse est connue sous le nom de *principe de Kerckhoff* (1883).

Pour arriver à ses fins, Charlie peut disposer de plusieurs niveaux d'attaques : Le premier niveau est *l'attaque à texte chiffré seul*. Dans ce cas, Charlie ne connait que le chiffré qui passe sur la ligne, ou éventuellement une collection de chiffrés, et cherche à reconstituer le clair correspondant à un chiffré donné.
Les autres niveaux d'attaques sont
- *l'attaque à clair connu :* Charlie a accès à des couples de messages (clairs, chiffrés), et utilise ces informations pour déchiffrer un message chiffré particulier.
- *l'attaque à clair choisi :* même contexte que précédemment, sauf qu'ici, il peut choisir les clairs des couples (clairs, chiffrés), (i.e. il a temporairement accès à la machine de chiffrement - vue comme une boîte noire - qui lui permet d'obtenir les chiffrés de messages clairs particuliers).
- *l'attaque à clair choisi adaptative :* c'est une attaque à clair choisi dans laquelle le choix des clairs dépend des chiffrés précédemment obtenus.
- *l'attaque à chiffré choisi :* Charlie choisit des chiffrés, et a accès aux clairs correspondants (i.e. il a temporairement accès à la boite noire de déchiffrement). Il veut ainsi retrouver le clair correspondant à un chiffré particulier.
- *l'attaque à chiffré choisi adaptative :* même attaque que précédemment, mais ici, le choix des chiffrés dépend des clairs précédemment obtenus.

Précisons que les deux dernières attaques s'appliquent aux schémas de chiffrement à clef publique.

Les attaques ci-dessus sont listées par ordre croissant de puissance. Par exemple, un schéma de chiffrement est plus vulnérable à une attaque à clair choisi qu'à une attaque à chiffré seul.
On considère qu'un schéma vulnérable aux attaques à chiffré seul n'est pas sûr. En général, on requiert qu'un schéma de chiffrement soit résistant aux attaques à clair choisi. Autrement dit, un bon schéma de chiffrement est un schéma pour lequel une attaque à clair choisi n'est pas meilleure que la recherche exhaustive. En outre, la résistance aux attaques à clair connu est considérée comme le strict minimum.

A titre d'exemple, les attaques sur les schémas classiques que nous avons mentionnées plus haut sont toutes à chiffré seul, exceptée celle concernant le schéma de chiffrement par transposition.

La sécurité d'un algorithme cryptographique peut-être évaluée sous différents modèles ; les trois plus importants sont :

- **La sécurité inconditionnelle.** C'est la mesure la plus forte pour un système cryptographique. Dans ce modèle, la connaissance du chiffré ne révèle aucune information sur le clair (au sens de la théorie de l'information). Autrement dit, avec les notations de la section A.1, on doit avoir $H(M \mid C) = H(M)$. Ainsi, même si l'adversaire a des ressources illimitées, il ne peut rien déduire sur le clair à partir de la seule connaissance du chiffré. La sécurité inconditionnelle est aussi appelée *sécurité parfaite*.
  Shannon a montré (en 1949) que, dans un système inconditionnellement sûr, il devait y avoir au moins autant de clefs que de messages clairs (voir section A.1). Une conséquence immédiate de ce résultat, dans le cas où la clef et le clair sont des mots sur le même alphabet, est que la longueur de la clef doit être au moins égale à celle du clair. Pour cette raison, les systèmes parfaits ne peuvent être utilisés en pratique. Néanmoins, nous en verrons un exemple dans la section suivante.
  Précisons qu'un tel modèle de sécurité n'est pas applicable aux schémas à clef publique (étant donné un chiffré c, il suffit d'appliquer l'algorithme de chiffrement à tous les clairs possibles pour retrouver celui correspondant à c).
- **La sécurité prouvée.** Un système est *prouvé sûr* lorsque la difficulté de le casser (i.e. trouver la clef secrète) est prouvée être équivalente à celle de résoudre un problème réputé difficile (comme la factorisation entière par exemple). Notons à ce propos qu'il existe des algorithmes à clef publique prouvés sûrs, mais que la sécurité prouvée pour les algorithmes à clef secrète reste un problème ouvert.
- **La sécurité calculatoire, ou pratique.** Dans ce modèle, le niveau de sécurité se mesure en nombre d'opérations nécessaires pour attaquer le système.
  On appelle *facteur de travail* $W(K)$, le travail minimum requis pour trouver la clef secrète K (mesuré en nombre d'opérations élémentaires, ou en temps de cycle, en utilisant la meilleure méthode connue). Cette définition doit se comprendre *à un moment donné* ; en effet, $W(k)$ varie avec les progrès des méthodes de cryptanalyse, et des technologies.
  Un système est dit *calculatoirement sûr* si $W(K)$ excède largement la puissance de calcul d'un adversaire éventuel.
  Ce modèle est le plus couramment utilisé.

Nous avons vu qu'un bon schéma de chiffrement devait être tel que la meilleure attaque est la recherche exhaustive sur l'espace des clefs. Cet espace de clefs doit donc être “grand”. A l'heure actuelle, pour les schémas à clef secrète, on considère qu'une clef de 40 bits (i.e. un espace de clefs de taille 2^{40}) offre une sécurité totalement insuffisante, 56 bits est pratiquement à la portée des ordinateurs actuels, et 64 bits constituent la borne inférieure d'une sécurité raisonnable, bien que l'on élabore et utilise de plus en plus des schémas dont les clefs font 128 bits.
A titre indicatif, le nombre de chaînes binaires de 64 bits est comparable au nombre de secondes écoulées depuis la création du système solaire ($\simeq 2 \times 10^{17}$), le nombre de nombres premiers de 75 chiffres est environ $5,2 \times 10^{72}$, et le nombre d'électrons dans l'univers est de l'ordre de $8,37 \times 10^{77}$ (soit du même ordre que le nombre de chaînes binaires de 256 bits : $2^{256} \simeq 1,2 \times 10^{77}$).

1.3 Un chiffrement parfait

Nous allons ici décrire un schéma de chiffrement inventé par Gilles Vernam en 1926 pour chiffrer les messages télégraphiques.

A l'origine, ce système chiffrait des message de l'alphabet A,...,Z, mais aujourd'hui, on le présente généralement sous sa forme binaire. L'espace des clairs est $\{0,1\}^n$, l'ensemble des chaînes binaires de taille n ($n \geq 1$ fixé). La clef k est également une suite binaire de taille n, et le chiffrement consiste simplement en la somme modulo 2 de chaque bit du clair avec chaque bit de la clef :
Pour $m = (m_1, \ldots, m_n) \in \{0,1\}^n$,

$$E_K(m) = (c_1, \ldots, c_n), \text{ avec } c_i = m_i + k_i \bmod 2, \ 1 \leq i \leq n.$$

Le déchiffrement est identique au chiffrement.

Lorsque la clef est choisie aléatoirement et uniformément dans $\{0,1\}^n$, et que chaque clef n'est utilisée que pour un seul chiffrement, ce système est appelé *masque jetable* ou *one-time pad*.

Théorème 1.1. *Le masque jetable est un chiffrement parfait.*

Preuve :
L'espace des clairs $\mathcal{P}$, des chiffrés $\mathcal{C}$ et des clefs $\mathcal{K}$ sont ici identiques, et égaux à $\{0,1\}^n$. Notons M, K, C, les variables aléatoires à valeurs dans $\mathcal{P}$, $\mathcal{K}$ et $\mathcal{C}$ respectivement, correspondant aux messages, clefs, et chiffrés. L'expression $P(M = m)$ indiquera la probabilité que la variable aléatoire M prenne la valeur $m \in \mathcal{P}$.
$\mathcal{K}$ étant muni de la distribution aléatoire uniforme sur $\{0,1\}^n$, on a :

$$P(K = k) = \frac{1}{2^n}, \ \forall k \in \mathcal{K}. \tag{1.1}$$

On doit montrer :

$$H(M \mid C) = H(M),$$

ou, de manière équivalente,

$$P(M = m \,|\, C = c) = P(M = m),\ \forall m \in \mathcal{P},\ \forall c \in \mathcal{C}.$$

La formule de Bayes donne :

$$P(M = m \,|\, C = c) = \frac{P(M = m)P(C = c|M = m)}{P(C = c)}.$$

On a :

$$P(C = c \,|\, M = m) = \sum_{\{k \in \mathcal{K},\, m = d_k(c)\}} P(K = k).$$

Dans le cas du one-time pad, étant donnés (m, c), il existe une unique clef $k \in \mathcal{K}$ telle que $m = d_k(c)$. (En effet, $m = d_k(c) = d_{k'}(c)$ implique $k_i + c_i = k'_i + c_i \bmod 2,\ \forall 1 \leq i \leq n$, donc $k'_i = k_i\ \forall i$). Ainsi (en utilisant (1.1)) :

$$P(C = c \,|\, M = m) = \frac{1}{2^n}.$$

Evaluons maintenant $P(C = c)$. Les distributions de probabilités sur $\mathcal{P}$ et $\mathcal{K}$ induisent une distribution de probabilité sur $\mathcal{C}$ de la manière suivante :

$$\begin{aligned} P(C = c) &= \sum_{\{(m,k) \in \mathcal{P} \times \mathcal{K},\ E_k(m) = c\}} P(K = k)P(M = m) \\ &= \frac{1}{2^n} \sum_{\{(m,k) \in \mathcal{P} \times \mathcal{K},\ E_k(m) = c\}} P(M = m). \end{aligned}$$

Pour $c \in \mathcal{C}$ fixé, $\forall m \in \mathcal{P},\ \exists!\, k \in \mathcal{K},\ E_k(m) = c$. D'où :

$$\sum_{\{(m,k) \in \mathcal{P} \times \mathcal{K},\ E_k(m) = c\}} P(M = m) = \sum_{m \in \mathcal{P}} P(M = m) = 1.$$

Il en résulte que

$$P(M = m \,|\, C = c) = P(M = m).$$

$\square$

Au sens du théorème de Shannon, on peut même dire que ce système est optimal (la taille de la clef est la plus petite possible). Il est à noter que ce système est parfait dans le cadre d'une attaque à chiffré seul, mais qu'il est vulnérable à une attaque à clair connu. En effet, K peut-être trivialement calculée à partir d'un couple (clair,chiffré). On voit ici que l'utilisation unique de la clef est primordiale.
Comme nous l'avons vu précédemment, un inconvénient majeur du masque jetable est que la clef doit être aussi longue que le message. Ceci rend difficile la distribution d'une telle clef (qui doit être réalisée par un canal sécurisé), d'autant que celle-ci ne peut servir qu'une seule fois. Cependant, ce système a été (l'est-il toujours ?) utilisé dans des contextes

militaires et diplomatiques (par exemple dans le cas du téléphone rouge entre Moscou et Washington pendant la guerre froide).

C'est ce constat d'impraticabilité des systèmes parfaits qui a conduit à l'essor de la cryptographie à partir de la deuxième moitié du XXème siècle, par la recherche de techniques permettant d'élaborer des systèmes sûrs selon d'autres modèles que la sécurité inconditionnelle.

Annexe A

Compléments mathématiques

A.1 Théorie de l'information

A.1.1 Rappels de probabilités

On rappelle ici quelques notions de base de la théorie des probabilités.

Soit Ω, un ensemble fini[1]. On note $\mathcal{F} = \mathcal{P}(\Omega)$, l'ensemble des parties de Ω, appelées évènements.

Une *probabilité* est une fonction P de $\mathcal{F}$ dans $[0,1]$, telle que :
- $P(\Omega) = 1$ et $P(\emptyset) = 0$.
- Si $(A_i)_{i\in I}$ est une collection d'évènements $A_i \in \mathcal{F}$, disjoints deux à deux, alors

$$P(\cup_{i\in I} A_i) = \sum_{i\in I} P(A_i).$$

Le triplet $(\Omega, \mathcal{F}, P)$ est appelé *espace probabilisé*.

Une probabilité P sur Ω est entièrement déterminée par la donnée de la collection $\{P(\{\omega\}),\ \omega \in \Omega\}$. Si, pour tout $\omega \in \Omega$, $P(\{\omega\}) = 1/|\Omega|$, alors P est la probabilité *uniforme* sur Ω.

La probabilité de l'évènement $A \in \mathcal{F}$ est donnée par :

$$P(A) = \sum_{\omega\in A} P(\{\omega\}).$$

Dans le cas où P est la probabilité uniforme, on a donc $P(A) = |A|/|\Omega|$.

Soit $A, B \in \mathcal{F}$, avec $P(B) \neq 0$. La probabilité *conditionnelle* de A sachant B, notée $P(A|B)$, est définie par

$$P(A|B) = \frac{P(A\cap B)}{P(B)}.$$

[1] Seul cas qui nous intéresse ici.

Cette quantité mesure la probabilité d'occurence de A, étant donné que B a eu lieu.
Deux évènements A et B sont *indépendants* si $P(A \cap B) = P(A)P(B)$. (Ceci correspond à $P(A|B) = P(A)$). On a la propriété suivante (formule de Bayes) :

$$P(A|B) = \frac{P(A)P(B|A)}{P(B)}.$$

Soit $(\Omega, \mathcal{F}, P)$, un *espace probabilisé.* Ω étant fini, toute fonction de Ω dans $\mathbb{R}$ ne peut prendre qu'un nombre fini de valeurs.
Une *variable aléatoire* (discrète) est une fonction X de Ω dans $\mathbb{R}$.

La *loi* de X, ou *distribution de probabilité* de X, est la donnée de la collection $\{P(\{X = x\}), x \in \mathbb{R}\}$, où $\{X = x\}$ est l'évènement $\{\omega \in \Omega,\ X(\omega) = x\}$. Par exemple, si X est à valeurs dans $\{1, \ldots, n\}$, et que $P(\{X = x\}) = 1/n$, $\forall x \in \{1, \ldots, n\}$, on dit que X suit la *loi uniforme* sur cet ensemble.

Soit X et Y, deux variables aléatoires sur $(\Omega, \mathcal{F}, P)$. La probabilité que X prenne la valeur x *et* Y prenne la valeur y est notée $P(\{X = x, Y = y\})$. Elle est égale à :

$$P(\{X = x, Y = y\}) = P(\{X = x\} \cap \{Y = y\}).$$

X et Y sont indépendantes si et seulement si, pour tous $x, y \in \mathbb{R}$,

$$P(\{X = x, Y = y\}) = P(\{X = x\})P(\{Y = y\}).$$

En se référant aux évènements correspondants, on exprime aisément $P(\{X = x, Y = y\})$ à l'aide de la probabilité conditionnelle $P(\{X = x\}|\{Y = y\})$; on peut également traduire la formule de Bayes en termes de variables aléatoires.

A.1.2 Entropie

Un cryptosystème est défini par l'espace de ses clefs, de ses messages, et par une transformation paramétrée par la clef. A chaque clef est associée une probabilité *a priori* : la probabilité de choisir cette clef parmi toutes les clefs de l'espace. De la même manière, à chaque message est associée une probabilité *a priori.* Ces probabilités sont également celles qu'a un attaquant de choisir un message et une clef ; autrement dit, elles représentent la connaissance *a priori* du contexte de la communication par l'attaquant.
Pour mesurer cette connaissance, ou encore la quantité d'information associée au choix d'une clef ou d'un message, on utilise des notions issues de la théorie de l'information, dont la principale est l'entropie.

L'entropie mesure l'information contenue dans un évènement lorsqu'il a lieu (*a posteriori*), ou, de manière équivalente, la quantité d'incertitude liée à cet évènement (*a priori*). Par exemple, soit X, une variable aléatoire discrète, et soit A, l'évènement $\{X = a\}$, cet évènement se produisant avec probabilité $P(A)$. Si $P(A) = 0,99$, on est presque assuré que A va se produire. Inversement, si $P(A) = 0,01$, il est raisonnable de croire que A ne va pas avoir lieu. Dans les deux cas, l'incertitude sur l'occurence (ou non) de A est faible. En

revanche, elle est maximale si $P(A) = 1/2$. En outre, elle ne dépend pas de la valeur a. Ainsi, la mesure de l'incertitude d'une variable aléatoire est une fonction de sa distribution de probabilité seulement.
On définit alors :

Définition A.1. *L'entropie d'une variable aléatoire X prenant chaque valeur x_i, $1 \leq i \leq n$ avec probabilité p_i est donnée par*

$$H(X) = -\sum_{i=1}^{n} p_i \lg(p_i). \tag{A.1}$$

On notera également $H(X) = H(p_1, \ldots, p_n)$. L'unité d'entropie est le bit.

On a : $H(X) \geq 0$, $\forall X$, et $H(X) = 0$ si, et seulement si, X est une variable aléatoire constante (l'un des p_i vaut 1). Si $p_i = 0$, $\lg(p_i)$ n'est pas défini. Mais la fonction $x \mapsto x\lg(x)$ est continue sur $\mathbb{R}^{+*}$ et $\lim_{x\to 0} x\lg(x) = 0$, donc on peut naturellement la prolonger par continuité en 0. Ainsi, on pose :

$$H(p_1, \ldots, p_{i-1}, 0, p_{i+1}, \ldots, p_n) = H(p_1, \ldots, p_{i-1}, p_{i+1}, \ldots, p_n).$$

Ce prolongement est également cohérent avec la quantité que nous voulons mesurer, puisqu'il n'y a aucune incertitude liée à un évènement se produisant avec probabilité 0. Pour la même raison, on supposera dans la suite que, dans la somme (A.1), les p_i sont tous non nuls.

Propriétés

Les propriétes suivantes se comprennent aisément avec la définition intuitive de l'entropie. Pour les preuves, on peut par exemple se référer à l'ouvrage "Codes and cryptography" de D. Welsh.

Proposition A.1.1. *Avec les notations de la définition précédente,*

$$H(X) \leq \lg n, \tag{A.2}$$

avec égalité si, et seulement si $p_i = 1/n$, $\forall 1 \leq i \leq n$.
Soit Y, une variable aléatoire de distribution de probabilité $(q_1, \ldots, q_m)$, $q_j = P[\{Y = y_j\}]$. On note $r_{ij} = P[\{X = x_i,\, Y = y_j\}]$, et $H(X,Y)$, l'entropie de la distribution jointe :

$$H(X,Y) = -\sum_{i=1}^{n}\sum_{j=1}^{m} r_{ij} \lg(r_{ij}).$$

Alors

$$H(X,Y) \leq H(X) + H(Y), \tag{A.3}$$

avec égalité si, et seulement si, X et Y sont indépendantes.

Preuve
Pour montrer l'inégalité (A.2), on considère la fonction $f : x \mapsto -x \lg x$, $0 \mapsto 0$ sur $[0,1]$. f est strictement concave, i.e. la courbe de f est au-dessus de toutes ses cordes, ce qui s'écrit :

$$\forall \lambda \in [0,1],\ \forall x, y \in [0,1], x \neq y,\ f(\lambda\, x + (1-\lambda)\, y) > \lambda\, f(x) + (1-\lambda)\, f(y).$$

Dans le cas où $x = y$, et seulement dans ce cas, l'expression précédente devient une égalité. Plus généralement, pour tous $\{\lambda_i\}_{1\leq i\leq n}$ tels que $\sum_{i=1}^n \lambda_i = 1$, on a, pour tous $y_i \in [0,1]$ non tous égaux :

$$f(\sum_{i=1}^n \lambda_i y_i) > \sum_{i=1}^n \lambda_i\, f(y_i),$$

avec égalité si, et seulement si, tous les y_i sont égaux.
En appliquant l'inégalité précédente avec $\lambda_i = \frac{1}{n}$ et $y_i = p_i$, $1 \leq i \leq n$, on obtient :

$$f(\sum_{i=1}^n \frac{p_i}{n}) = f(\frac{1}{n}) > \sum_{i=1}^n \frac{1}{n}\, f(p_i),$$

soit

$$-\frac{1}{n} \lg\, \frac{1}{n} > -\frac{1}{n} \sum_{i=1}^n p_i \lg\, p_i,$$

ou encore

$$\lg\, n > -\sum_{i=1}^n p_i \lg\, p_i = H(X).$$

Enfin, on a l´egalité si, et seulement si, tous les p_i sont égaux, et étant donné que $\sum_{i=1}^n p_i = 1$, si, et seulement si, $p_i = \frac{1}{n}$.
L'inégalité (A.3) est laissée à titre d'exercice. Remarquons qu'elle peut aisément être déduite du lemme suivant (à montrer également) :
Soit $\{p_i\}_{1\leq i\leq n}$, une distribution de probabilité. Alors le minimum, sur toutes les distributions de probabilité $\{q_i\}_{1\leq i\leq n}$, de

$$-\sum_i p_i \lg q_i,$$

est atteint lorsque $q_i = p_i$, $\forall 1 \leq i \leq n$. □

Soit X et Y, deux variables aléatoires. Pour toute valeur y de Y fixée, on a une distribution de probabilité conditionnelle $P(X|y)$, et l'entropie attachée à cette distribution est

$$H(X|y) = -\sum_x P[\{X = x\}|\{Y = y\}] \lg(P[\{X = x\}|\{Y = y\}]).$$

On définit alors l'entropie conditionnelle $H(X|Y)$ comme étant la moyenne pondérée (par $P[\{Y = y\}]$) de $H(X|y)$, pour toutes les valeurs y que peut prendre Y, soit :

$$H(X|Y) = -\sum_y P[\{Y = y\}] \sum_x P[\{X = x\}|\{Y = y\}] \lg(P[\{X = x\}|\{Y = y\}]).$$

$H(X|Y)$ mesure la quantité moyenne d'information sur X révélée par Y, ou encore l'incertitude restant sur X, une fois connue Y. L'entropie conditionnelle vérifie les propriétés suivantes :

Proposition A.1.2.
$$H(X, Y) = H(Y) + H(X|Y). \tag{A.4}$$

On déduit des deux propositions précédentes que

$$H(X|Y) \leq H(X),$$

avec égalité si, et seulement si, X et Y sont indépendantes.
Il est naturel que, si X et Y ne sont pas indépendantes, l'entropie de X diminue lorsque l'on connaît Y. En revanche, on n'a pas de résultat similaire pour $P(X|Y)$ et $P(X)$.

A.1.3 Entropie et cryptographie

Nous appliquons ici les résultats sur l'entropie à la mesure de l'information véhiculée par les différentes composantes d'un cryptosystème. Soit donc E, une fonction de chiffrement. On note $\mathcal{P}$, $\mathcal{K}$ et $\mathcal{C}$ l'espace des clairs, des clefs et des chiffrés. M, K et C sont les variables aléatoires, à valeurs dans $\mathcal{P}$, $\mathcal{K}$ et $\mathcal{C}$ respectivement, représentant le message, la clef et le chiffré. On suppose en outre que M et K sont indépendantes, ce qui signifie que la clef est choisie indépendamment du message à chiffrer (elle est en général choisie avant tout chiffrement). La probabilité que M prenne la valeur $m \in \mathcal{P}$ sera notée $P(M = m)$ (au lieu de $P(\{M = m\})$).
Les distributions de probabilités sur $\mathcal{P}$ et $\mathcal{K}$ induisent une distribution de probabilité sur $\mathcal{C}$ de la manière suivante :

$$P(C = c) = \sum_{\{(m,k) \in \mathcal{P} \times \mathcal{K},\ E_k(m) = c\}} P(K = k) P(M = m).$$

La principale quantité qui nous intéresse est $H(K|C)$, i.e. l'information sur la clef révélée par un chiffré[2]. Elle est exprimée par le théorème suivant, qui signifie que l'incertitude sur la clef, connaissant un chiffré, est au moins aussi grande que celle sur le message, connaissant un chiffré.

Théorème A.1.1.
$$H(K|C) = H(M|C) + H(K|M, C). \tag{A.5}$$

Preuve :
D'après (A.4),
$$H(M|C) = H(M, C) - H(C),$$
et toujours d'après (A.4) (avec $X = K$ et $Y = \{M, C\}$),
$$H(M|C) = H(K, M, C) - H(K|M, C) - H(C).$$

[2]Cette quantité est appelée *key equivocation*.

De même :

$$H(K|C) = H(K,C) - H(C) = H(M,K,C) - H(M|K,C) - H(C).$$

Mais $H(M|K,C) = 0$ (étant donnés un chiffré et une clef, le clair correspondant est uniquement déterminé). On en déduit donc le résultat cherché. □

Remarque : Une autre expression pour $H(K|C)$, se déduisant aisément de (A.4), est

$$H(K|C) = H(K) + H(M) - H(C). \tag{A.6}$$

La notion suivante fonde le concept de sécurité inconditionnelle :

Définition A.2. *Un cryptosystème $E(\mathcal{P},\mathcal{K},\mathcal{C})$ est dit* parfaitement secret[3] *s'il vérifie*

$$H(M|C) = H(M).$$

Autrement dit, la probabilité *a posteriori* que le clair soit m, étant donné que le chiffré observé est c, est égale à la probabilité *a priori* que le clair soit m ; ou encore, la connaissance de c ne donne aucune information sur le clair.

Malheureusement, dans un tel système, le nombre de clefs est au moins aussi grand que le nombre de messages :

Théorème A.1.2. *(Shannon, 1949)*
Soit $E(\mathcal{P},\mathcal{K},\mathcal{C})$, un cryptosystème parfait. Alors

$$|\mathcal{K}| \geq |\mathcal{P}|.$$

Preuve
$E(\mathcal{P},\mathcal{K},\mathcal{C})$ est parfait, donc $H(M|C) = H(M)$, ce qui équivaut à $P(M = m|C = c) = P(M = m)$, $\forall m \in \mathcal{P}$, $\forall c \in \mathcal{C}$. La formule de Bayes

$$P(M = m|C = c) = \frac{P(C = c|M = m)P(M = m)}{P(C = c)},$$

donne $P(C = c|M = m) = P(C = c)$.
Si $P(C = c) = 0$, alors on peut enlever c de $\mathcal{C}$ (en effet, ce chiffré n'est jamais atteint). On suppose donc $P(C = c) > 0$, $\forall c \in \mathcal{C}$.
Soit $m \in \mathcal{P}$ fixé. On a $P(C = c|M = m) = P(C = c) > 0$, $\forall c \in \mathcal{C}$. Donc, pour tout $c \in \mathcal{C}$, il existe au moins une clef k telle que $E_k(m) = c$. Ceci implique que $|\mathcal{K}| \geq |\mathcal{C}|$. Le résultat voulu en découle, par le fait que tout schéma de chiffrement vérifie $|\mathcal{C}| \geq |\mathcal{P}|$ (injectivité de E, garantissant l'unicité du déchiffrement). □

On note que, si les messages et les clefs sont respectivement des n-uplets et des l-uplets sur un même alphabet $\mathcal{A}$ (par exemple $\mathcal{A} = \{0,1\}$), pouvant prendre toute valeur dans ces ensembles, alors $|\mathcal{K}| = |\mathcal{A}|^l$, et $|\mathcal{P}| = |\mathcal{A}|^n$; le théorème de Shannon revient alors à dire que, dans un système parfait, la clef doit être au moins aussi longue que le message.

[3] on dira également parfait.

Entropie et langages naturels

Soit L, un langage sur un alphabet $\mathcal{A}$, et $E(\mathcal{P},\mathcal{K},\mathcal{C})$, un cryptosystème. On suppose que E chiffre des messages intelligibles dans L ; on a donc $\mathcal{P}=\mathcal{A}^*$, et

$$m\in\mathcal{P},\ P(\{M=m\})>0 \Leftrightarrow m\in L.$$

Soit $m=(m_1,\ldots,m_N)$, un clair intelligible de taille N, et $c=(c_1,\ldots,c_N)=E_k(m)$.
En général, étant donné c, il existe plusieurs couples $(\tilde{m},\tilde{k})\in\mathcal{P}\times\mathcal{K}$, tels que $E_{\tilde{k}}(\tilde{m})=c$, une seule de ces clefs ($\tilde{k}=k$) donnant le clair originel m. (Par exemple, si $\mathcal{A}=\{$A,...,Z$\}$, et E_k consiste en le décalage de k positions de chaque lettre du message (modulo 26), il peut y avoir plusieurs k tels que le décalage de $(26-k)$ lettres de $(c_1,\ldots,c_N)$ résulte en un clair intelligible).

On se propose ici d'évaluer la longueur moyenne de chiffré N_0 pour laquelle la connaissance d'un chiffré $c=(c_1,\ldots,c_{N_0})$ permet de déterminer la clef correspondante de manière unique. N_0 est appelée la *distance d'unicité*[4] (associée au langage L) du cryptosystème.

Soit donc L, un langage naturel sur l'alphabet $\mathcal{A}$. L'entropie par lettre de L, notée H_L, est définie par l'information moyenne contenue dans chaque lettre de L, dans une suite *intelligible*. (Par exemple, une suite *aléatoire* de lettres de $\mathcal{A}=\{$A,...,Z$\}$ possède une entropie par lettre de $\lg(26)\simeq 4,76$).
En première approximation de H_L, on peut prendre $H(P)$, où P est la variable aléatoire dont la loi est la probabilité d'occurence de chaque lettre dans L. Mais la succession des lettres de L n'étant pas aléatoire, une meilleure approximation de H_L serait $\frac{H(P^2)}{2}$, où P^2 désigne la variable aléatoire dont la loi est la probabilité d'occurence des digrammes dans L. En poursuivant ce raisonnement, on définit P^n comme étant la variable aléatoire dont la distribution de probabilité est celle de tous les groupes de n lettres dans L, et on pose

$$H_L=\lim_{n\to\infty}\frac{H(P^n)}{n}.$$

On définit en outre

$$R_L=1-\frac{H_L}{\lg(|\mathcal{A}|)}.$$

R_L est la redondance[5] de L.

Revenons maintenant au calcul de la distance d'unicité. En notant C_N, la variable aléatoire pouvant prendre comme valeurs les chiffrés de longueur N, on cherche donc N_0 tel que :

$$H(K|C_{N_0})=0.$$

D'après (A.6), et avec des notations évidentes :

$$H(K|C_N)=H(K)+H(M_N)-H(C_N).$$

En outre, on fait les hypothèses suivantes :

[4]Cette notion n'est définie que pour les systèmes chiffrant des messages intelligibles.
[5]En effet, $\lg(|\mathcal{A}|)$ est l'entropie par lettre d'une suite aléatoire de $\mathcal{A}$.

- Les clefs sont choisies aléatoirement et uniformément dans $\mathcal{K}$, i.e. $P(K=k) = 1/|\mathcal{K}|$, et donc $H(K) = \lg(|\mathcal{K}|)$.
- On approche $H(M_N)$ par NH_L.
- On approche $H(C_N)$ par $N\lg(|\mathcal{A}|)$ (i.e. on suppose que les cryptogrammes suivent une distribution aléatoire et uniforme).

Ainsi :

$$H(K|C_N) = 0 \Leftrightarrow \lg(|\mathcal{K}|) + NH_L - N\lg(|\mathcal{A}|) = 0.$$

En résolvant en N, on trouve

$$N_0 = \frac{\lg(|\mathcal{K}|)}{\lg(|\mathcal{A}|) - H_L}.$$

Par exemple, si L est l'Anglais, on a $|\mathcal{A}| = 26$ et $H_L \simeq 1,25$. En utilisant un chiffrement par substitution, soit $|\mathcal{K}| = 26!$, on obtient :

$$N_0 \simeq \frac{\lg(26!)}{\lg(26) - 1,25} \simeq 25,$$

autrement dit, un chiffré d'en moyenne 25 lettres suffit à déterminer la clef de manière unique.

Note : une autre expression de la distance d'unicité, mettant en évidence son lien avec la redondance R_L, est :

$$N_0 = \frac{\lg(|\mathcal{K}|)}{R_L \lg(|\mathcal{A}|)}.$$

A.2 Complexité

Le but de la théorie de la complexité est de fournir des méthodes permettant de classifier les problèmes de calcul selon les ressources nécessaires pour les résoudre. Les ressources mesurées sont principalement le temps et l'espace mémoire. Une telle classification ne doit pas dépendre d'un modèle de calcul particulier, mais doit plutôt évaluer la difficulté intrinsèque du problème.
Ainsi, nous définirons les différentes classes de complexité à partir de la notion d'*algorithme*, sans "descendre" au niveau de la modélisation des algorithmes par des machines de Turing[6]. Cette approche, évidemment incomplète, nous paraît être celle qui offre le plus de clarté, dans un cours dont le but n'est pas d'approfondir la théorie de la complexité, mais pour lequel cette discipline, quoique centrale, reste un "outil". Néanmoins, pour un exposé complet des machines de Turing et de cette théorie en général, on pourra consulter l'excellent livre de M. Garey et S. Johnson "*computers and intractability : a guide to the theory of NP-completeness*".

[6]considérées comme le "modèle universel" de calcul, les autres modèles - circuits booléens ou machines à accès direct - pouvant être simulés par celui-ci.

A.2.1 Définitions de base

Partant donc d'un algorithme - c'est-à-dire d'une procédure séquentielle de calcul prenant en entrée une variable et produisant une réponse après un temps fini - on définit les notions suivantes :

- La *taille* de l'entrée est le nombre de bits nécessaires à sa représentation en base 2. Par exemple, la taille d'un entier n est $1 + \lfloor \lg(n) \rfloor$, où lg désigne le logarithme en base 2. On approximera cette taille par $\lg(n)$.
- Le *temps de calcul* d'un algorithme est le nombre d'opérations élémentaires éxécutées par celui-ci jusqu'à son arrêt. L'unité de mesure est le plus souvent une opération binaire (arithmétique ou logique), mais cela peut aussi être une comparaison, une instruction machine, une multiplication modulaire ...
- Le temps de calcul *dans le pire cas* est une borne supérieure sur le temps de calcul en considérant n'importe quelle entrée de taille fixée, exprimée comme fonction de la taille de l'entrée.
- Le temps de calcul *moyen* est la moyenne[7] du temps de calcul sur toutes les entrées d'une taille fixée ; c'est une fonction de la taille de l'entrée.

En toute rigueur, les définitions suivantes s'entendent *dans le pire cas*, mais en cryptographie, on s'intéresse surtout à la complexité moyenne ; en effet, on requiert que le temps de calcul nécessaire pour casser un système soit prohibitif *pour presque toutes les instances* de ce système, et non pas pour des instances isolées.

Définition A.3. *Soit $\mathcal{A}$, un algorithme prenant en entrée une donnée de taille N.*
- *$\mathcal{A}$ est dit* polynômial *si son temps de calcul est une fonction de la forme $O(P(N))$, où P est un polynôme "de petit degré". En particulier, $\mathcal{A}$ est linéaire si son temps est en $O(N)$.*
- *si le temps de calcul de $\mathcal{A}$ est en $O(exp(\alpha N))$, $\alpha \in \mathbb{R}^{+*}$, $\mathcal{A}$ est dit* exponentiel.
- *$\mathcal{A}$ est* sous-exponentiel *s'il a un temps de calcul de la forme $exp(o(N))$.*

En général, on aura à évaluer la complexité d'algorithmes prenant en entrée un entier n. On aura alors $N = \lg n$. En outre, pour un tel entier, on rencontrera souvent la complexité sous-exponentielle suivante :

$$O(exp((c + o(1))(\ln n)^{\alpha}(\ln \ln n)^{1-\alpha})),\ c \in \mathbb{R}^{+*},\ 0 < \alpha < 1,$$

et en particulier $O(exp(c\sqrt{\ln n \ln \ln n}))$.

A titre d'exemple, on compare la croissance asymptotique de quelques fonctions d'une variable réelle x :

$$1 < \ln \ln x < \ln x < exp(\sqrt{\ln x \ln \ln x}) < x^{\epsilon} < x^{c} < x^{\ln x} < c^{x} < x^{x},$$

où $0 < \epsilon < 1 < c$.

[7] en choisissant une distribution de probabilité, en général la distribution uniforme, sur les données d'une taille fixée ; le temps de calcul moyen est alors l'espérance de la variable aléatoire "temps de calcul".

A.2.2 Classes de complexité

On a recours à un algorithme pour résoudre un *problème*, c'est-à-dire une question qui possède en général plusieurs paramètres, dont les valeurs ne sont pas spécifiées. Une *instance* d'un problème est la donnée de ce problème sur une valeur particulière de ses paramètres.

Un algorithme *résout* un problème Π lorsque, appliqué à une instance quelconque de Π, il produit toujours une réponse pour cette instance. En particulier, si Π est un problème de calcul[8], $\mathcal{D}_\Pi$, l'ensemble des instances de Π, et $\mathcal{S}_\Pi(I)$ l'ensemble des solutions pour une instance I, un algorithme A résout Π si, prenant en entrée $I \in \mathcal{D}_\Pi$, A retourne un élément de $\mathcal{S}_\Pi(I)$ si $\mathcal{S}_\Pi(I) \neq \emptyset$, et "non" si $\mathcal{S}_\Pi(I) = \emptyset$.

Les classes de complexité que nous allons maintenant définir s'appliquent à des problèmes de *décision*, c'est-à-dire des problèmes dont la solution (le cas échéant) est "oui" ou "non". La raison en est double : d'une part les problèmes de décision sont fortement liés à la théorie des langages, dans le sens où l'ensemble des instances d'un problème Π pour lesquelles la réponse est "oui" (que l'on notera Y_Π dans la suite) définit un langage, qui est donc "reconnaissable" par l'algorithme résolvant Π. D'autre part, les problèmes de décision et les problèmes de calcul sont étroitement liés. Par exemple, tout problème de décision Π peut être formulé en un problème de calcul. En effet, il suffit de définir l'ensemble des solutions de Π sur une instance I par :

$$\mathcal{S}_\Pi(I) = \{\text{oui}\}, \text{ si } I \in Y_\Pi, \ \mathcal{S}_\Pi(I) = \emptyset, \text{ si } I \in \mathcal{D}_\Pi \setminus Y_\Pi.$$

En outre, la plupart des problèmes de calcul Π admettent un analogue décisionnel $\tilde{\Pi}$, qui, *en général*, n'est "pas plus difficile" que Π, dans le sens où, lorsqu'une solution polynômiale de Π existe, on peut en dériver une réponse à $\tilde{\Pi}$ en temps polynômial ; et donc inversement, si $\tilde{\Pi}$ est difficile, alors *en général* Π l'est aussi. Enfin, de nombreux problèmes de décision ont été prouvés "au moins aussi difficiles" que leur analogue calculatoire.
Ainsi, l'étude des problèmes de décision fournit également des résultats sur les problèmes de calcul associés.

L'appartenance d'un problème à une classe de complexité s'entend pour une instance générale du problème. En particulier, si un problème est difficile, cela ne veut pas dire que toutes ses instances le sont (voir par exemple le problème du voyageur de commerce).

Soit Π, un problème de décision. Avec les notations précédentes, on définit le complémentaire Π^c de Π comme étant le problème dont l'ensemble des instances conduisant à une réponse "oui" est

$$Y_{\Pi^c} = D_\Pi - Y_\Pi.$$

Définition A.4. – *La classe* **P** *est l'ensemble des problèmes de décision admettant un algorithme polynômial pour les résoudre.*
– *La classe* **NP** *est l'ensemble des problèmes de décision pour lesquels une réponse "oui" peut être vérifiée en temps polynômial, en utilisant une information supplémentaire notée* S.

[8]où la question consiste à rechercher une solution.

– *On définit également la classe*

$$\mathbf{co}-\mathbf{NP} = \{\Pi^c,\ \Pi \in \mathbf{NP}\}.$$

Premières remarques

• Alors que la classe **P** est synonyme de “résolvabilité en temps polynômial”, la classe **NP** isole la notion de “vérifiabilité en temps polynômial”.

• Le fait qu'un problème Π de la classe **NP** puisse être vérifiable en temps polynômial implique que l'information S intervenant dans cette vérification est de taille polynômiale en la taille de l'entrée de Π. Néanmoins, cela ne donne aucune information sur la difficulté d'obtenir S. Ce qui est stipulé ici est que, si une instance I appartient à Y_Π, alors un tel S existe, et dans ce cas la question $I \in Y_\Pi$? peut être vérifiée en temps polynômial.

• On a l'équivalence :

$$\Pi \in \mathbf{P} \Leftrightarrow \Pi^c \in \mathbf{P}.$$

En effet, étant donné un algorithme polynômial résolvant Π, le même algorithme, dans lequel on a échangé les sorties “oui” et “non”, résout Π^c.
En revanche, si $\Pi \in \mathbf{NP}$, une réponse “oui” au problème Π^c sur une instance I n'est pas toujours vérifiable en temps polynômial. Autrement dit, $\Pi \in \mathbf{NP}$ n'implique pas $\Pi^c \in \mathbf{NP}$. Si c'était le cas, on aurait **NP=co-NP**.

Pour fixer les idées, prenons le problème d'optimisation suivant :

Problème du voyageur de commerce :
Soit $C = \{c_1, \ldots, c_m\}$, un ensemble de villes. Trouver un parcours[9] de toutes les villes de C dont la longueur est minimale.

Le problème de décision associé est :
Soit $C = \{c_1, \ldots, c_m\}$, un ensemble de villes, et $B \in \mathbb{N}$. Existe-t-il un parcours de toutes les villes de C dont la longueur est inférieure à B ?

Etant donné qu'il y a $m!$ parcours possibles, ce problème n'appartient pas à **P**. En effet, si on suppose que le calcul d'un parcours prend $1\mu s$, alors, pour $m = 10$ villes, le calcul de tous les parcours prend $3,6$ s., pour $m = 15$, 15 jours, et pour $m = 25$, 490 milliards d'années...

En revanche, il est clair que, si l'on possède un parcours $S =< c_{\pi(1)}, \ldots, c_{\pi(m)} >$, i.e. une permutation π de l'ensemble des positions $\{1, \ldots, m\}$, alors on peut vérifier en temps polynômial que la somme des distances entre deux villes “consécutives” de S est inférieure à B, i.e. :

$$\sum_{i=0}^{m-1} d(c_{\pi(i)}, c_{\pi(i+1)}) + d(c_{\pi(m)}, c_{\pi(1)}) \leq B.$$

Donc ce problème appartient à **NP**.
D'autre part, le problème Π^c correspondant au problème décisionnel du voyageur de commerce est :

[9] En revenant au point de départ.

Etant donnés $C = \{c_1, \ldots, c_m\}$, un ensemble de villes, et $B \in \mathbb{N}$, est-il vrai qu'il n'existe aucun parcours de longueur inférieure à B ?

Pour ce problème, une réponse "oui" ne peut être vérifiée d'une façon autre que celle consistant à passer en revue tous les parcours possibles ; en d'autres termes, une instance quelconque - c'est-à-dire ici la donnée du couple (C, B) - de ce problème pour laquelle la réponse est "oui" ne peut être vérifiée en temps polynômial.

Notons néanmoins que sur, l'instance $(C, B < m \times \min_{i \neq j} d(c_i, c_j))$, la réponse "oui" peut être vérifiée en temps polynômial $O(m \lg(m))$ (temps nécessaire à la recherche de la distance minimale entre deux villes de C). Cette instance du problème Π^c est donc "facile".

Lien entre les classes de complexité

On sait que

$$\mathbf{P} \subseteq \mathbf{NP} \quad \text{et} \quad \mathbf{P} \subseteq \textbf{co-NP}.$$

Mais les questions

$$\mathbf{P} = \mathbf{NP}\,?, \quad \mathbf{NP} = \textbf{co-NP}\,? \quad \text{ou} \quad \mathbf{P} = \mathbf{NP} \cap \textbf{co-NP}\,?$$

restent irrésolues, bien qu'il soit conjecturé que la réponse à ces trois questions est négative.

Pour comparer les problèmes, on utilise la notion de *réductibilité* que nous introduisons maintenant.

Définition A.5. *Soit Π_1 et Π_2, deux problèmes.*
- Π_1 *est dit* polynômialement réductible *à* Π_2 *s'il existe un algorithme résolvant* Π_1 *qui utilise comme sous-routine un algorithme B résolvant* Π_2*, et qui fonctionne en temps polynômial dès lors que B fonctionne en temps polynômial. On note alors* $\Pi_1 \leq_P \Pi_2$.
- Π_1 *et* Π_2 *sont* calculatoirement équivalents *si* $\Pi_1 \leq_P \Pi_2$ *et* $\Pi_2 \leq_P \Pi_1$.

Remarques

• Si $\Pi_1 \leq_P \Pi_2$, alors Π_2 est "au moins aussi difficile" que Π_1.
• La relation $\leq_P$ est transitive. De plus, si Π_1 et Π_2 sont des problèmes de décision, alors $\Pi_2 \in \mathbf{P}$ et $\Pi_1 \leq_P \Pi_2$ implique $\Pi_1 \in \mathbf{P}$.

La notion de réductibilité nous permet de définir la classe importante suivante (classe **NPC**) :

Définition A.6. *Soit* Π*, un problème de décision.* Π *est* **NP***-complet si :*
- $\Pi \in \mathbf{NP}$,
- $\Pi' \leq_P \Pi,\ \forall\, \Pi' \in \mathbf{NP}$.

Les problèmes **NP**-complets sont les problèmes "les plus difficiles" de **NP**, dans le sens où ils sont au moins aussi difficiles que n'importe quel problème de **NP**.

La définition de la classe **NPC** implique que l'existence d'un seul problème Π appartenant à la fois à **NPC** et **P** suffirait à prouver $\mathbf{P} = \mathbf{NP}$. Cette assertion étant hautement improbable, on en déduit que l'appartenance d'un problème à **NPC** fournit une preuve de sa difficulté.

Soit Π_1, et Π_2, deux problèmes de décision. Si $\Pi_1 \in$ **NP**, $\Pi_2 \in$ **NPC**, et $\Pi_2 \leq_P \Pi_1$, alors $\Pi_1 \in$ **NPC**. Ceci conduit à la méthode suivante de preuve d'appartenance d'un problème décisionnel Π_1 à **NPC** :

1. prouver que $\Pi_1 \in$ **NP**,
2. choisir $\Pi_2 \in$ **NPC**,
3. prouver que $\Pi_2 \leq_P \Pi_1$.

Cette méthode a un sens si la classe **NPC** est non-vide. Heureusement, l'initiateur de la théorie[10] de la **NP**-complétude, S. Cook, a montré qu'il existait un problème de la classe **NP**, appelé "satisfiabilité" ou "SAT", tel que tout autre problème de cette classe lui était polynômialement réductible. SAT est un problème de logique booléenne, qui peut s'énoncer de la manière suivante :

Etant donné un ensemble $U = \{u_1, \ldots, u_n\}$ de variables booléennes, et une collection C de clauses (formules logiques) sur U, existe-t-il une affectation des variables qui satisfait simultanément toutes les clauses de C ?

Par exemple, soit $U = \{u_1, u_2\}$, et $C = \{\{u_1, \bar{u_2}\}, \{\bar{u_1}, u_2\}\}$ (où $\{a, b\}$ représente la disjonction des variables a et b). C est satisfiable s'il existe des valeurs de u_1 et u_2 telles que les deux clauses $c_1 = \{u_1, \bar{u_2}\}$ et $c_2 = \{\bar{u_1}, u_2\}$ soient vraies. Il est clair que $u_1 = u_2 = 1$ convient ; donc C est satisfiable.
Il est facile de voir que $C = \{\{u_1, \bar{u_2}\}, \{\bar{u_1}\}, \{u_2\}\}$ n'est pas satisfiable.

On ne connaît pas d'algorithme polynômial permettant de résoudre ce problème. En revanche, étant donnée une affectation des variables, on peut vérifier en temps polynômial qu'elle convient. Donc ce problème est dans **NP**. Le preuve d'appartenance à **NPC** n'est pas développée ici, et peut être trouvée dans l'ouvrage de M. Garey et S. Johnson.

En fait, il existe de nombreux problèmes **NP**-complets dans des disciplines diverses, comme par exemple la combinatoire, la théorie des nombres, la logique...
Le problème décisionnel du voyageur de commerce mentionné plus haut a également été prouvé **NP**-complet.

Nous terminons cette partie par la définition des problèmes **NP**-durs, qui s'applique à des problèmes calculatoires ou décisionnels.

Définition A.7. *Un problème Π est* **NP**-*dur s'il existe un problème* **NP**-*complet Π' tel que $\Pi' \leq_P \Pi$.*

Il est à noter qu'un problème **NP**-complet est aussi **NP**-dur.
De plus, soit Π, un problème de calcul, et $\tilde{\Pi}$, le problème décisionnel associé. Si $\tilde{\Pi} \leq_P \Pi$, et que $\tilde{\Pi}$ est **NP**-complet, alors Π est **NP**-dur. Autrement dit, pour prouver qu'un problème de calcul Π associé à un problème décisionnel $\tilde{\Pi}$ **NP**-complet est **NP**-dur, il suffit de montrer que l'existence d'un algorithme polynômial résolvant Π permet de résoudre $\tilde{\Pi}$ en temps polynômial.

[10]Cette théorie date de 1971.

A.3 Algèbre

On rappelle ici des notions élémentaires d'algèbre discrète.

Soit a, $b \in \mathbb{Z}$. On note (a, b), leur p.g.c.d.
La fonction ϕ d'Euler est définie, pour tout $n \in \mathbb{N}^*$, par :

$$\phi(n) = |\{x \in \mathbb{N},\ 1 \leq x \leq n,\ (x, n) = 1\}|.$$

Propriétés :
- Si p est premier, $\phi(p) = p - 1$.
- Si $(m, n) = 1$, $\phi(mn) = \phi(m)\phi(n)$.
- Soit $n \in \mathbb{N}$, et $n = \prod_{i=1}^{k} p_i^{e_i}$, p_i premier. Alors $\phi(n) = \prod_{i=1}^{k} p_i^{e_i - 1}(p_i - 1)$.

A.3.1 Groupes finis

Dans cette section, G désigne un groupe noté multiplicativement, d'élément neutre 1.
G est dit *fini* s'il contient un nombre fini d'éléments. L'*ordre* de G est le cardinal de G et est noté $|G|$.

Si G est fini, pour tout $g \in G$, il existe $k = k_g$ tel que $g^k = 1$ dans G. Le plus petit entier k vérifiant une telle égalité est appelé l'*ordre* de g et est noté $ord(g)$.

Théorème A.3.1. *Soit G, un groupe fini, et H, un sous-groupe de G. Alors $|H|$ divise $|G|$ (on note $|H| \mid |G|$).*

Preuve :
Soit H, un sous-groupe de G, et notons $r = |H|$, $n = |G|$. Pour prouver le résultat, on introduit la relation suivante :

$$x, y \in G,\ x\,\mathcal{R}\,y \Leftrightarrow xy^{-1} \in H.$$

On montre aisément que $\mathcal{R}$ est une relation d´équivalence. On note xH, la classe d'équivalence de $x \in G$.
Considérons la fonction

$$\begin{array}{rcl} f_x\ :H & \longrightarrow & xH \\ g & \longmapsto & xg. \end{array}$$

f_x est une bijection de H dans xH. Donc H et xH ont même cardinal, et ceci pour tout $x \in G$. Autrement dit, toutes les classes d'équivalence ont le même nombre d'éléments, à savoir r. Les classes d'équivalence formant une partition de G, si l'on note s, le nombre de classes, on a donc $n = rs$. $\square$

Soit $a \in G$. L'ensemble

$$< a >= \{a^i,\ 0 \leq i \leq ord(a) - 1\},$$

est un sous-groupe de G ; c'est le *sous-groupe cyclique* de G engendré par a.
Du théorème précédent, on déduit :

Corollaire A.3.1. *Soit G, un groupe fini, et $a \in G$. Alors $ord(a) \mid |G|$.*

Théorème A.3.2. *Soit G, un groupe fini, et $n = |G|$. Alors $g^n = 1 \, \forall g \in G$.*

Preuve :
Soit $g \in G$, et $k_g = \mathrm{ord}(g)$. Par le corollaire A.3.1, $k_g = |< g >| \mid n$, i.e. $n = rk_g$, $r \in \mathbb{N}^*$, donc $g^n = (g^{k_g})^r = 1$. □

Définition A.8. *Un groupe fini G est* cyclique *s'il existe $a \in G$, tel que $G =< a >$. Dans ce cas, on dit que a est un* générateur *de G.*

Proposition A.3.1. *Tout sous-groupe d'un groupe cyclique est cyclique.*

Preuve :
Soit H, un sous-groupe de G, et a, un générateur de G. Notons d, le plus petit entier positif tel que $a^d \in H$ (H contient au moins une puissance positive de a car, si $a^n \in H$, alors $a^{-n} \in H$). Soit $a^s \in H$, $s \in \mathbb{N}$. La division euclidienne de s par d donne $s = dq + r$, $0 \leq r < d$. Or, $a^s(a^{-d})^q = a^r \in H$, ce qui contredit la minimalité de d, sauf si $r = 0$; et donc tout élément $a^s \in H$ vérifie $d|s$. Ainsi, $H =< a^d >$. □

Théorème A.3.3. *Soit G, un groupe cyclique d'ordre n et a, un générateur de G.*
- *(i) Soit $1 \leq k \leq n$. Alors $ord(a^k) = n/(n,k)$. En particulier, a^k engendre G si, et seulement si, $(n,k) = 1$.*
- *(ii) Soit d, un diviseur de n. Les éléments de G d'ordre d sont les éléments de la forme $a^{rn/d}$, avec $(r,d) = 1$.*
- *(iii) Pour chaque d divisant n, G possède exactement un sous-groupe d'ordre d et $\phi(d)$ éléments d'ordre d.*

Preuve :
(i)- On a : $(a^k)^{n/(n,k)} = (a^n)^{k/(n,k)} = 1$, donc $n/(n,k)$ est un multiple de $ord(a^k)$. D'autre part, tout entier m vérifiant $(a^k)^m = 1$ est tel que $n \mid km$. Or,

$$n|km \Leftrightarrow \frac{n}{(n,k)} \mid \frac{km}{(n,k)}.$$

$n/(n,k)$ et $k/(n,k)$ étant premiers entre eux, il s'ensuit que $n/(n,k) \mid m$, ce qui prouve le résultat.
(ii)- Soit $d \mid n$. D'après (i), il suffit de déterminer pour quels entiers k, $1 \leq k \leq n$, on a $n/(n,k) = d$, i.e. $(n,k) = n/d$. Un tel k doit clairement être un multiple de n/d, soit $k = rn/d$. De plus,

$$(n, r\frac{n}{d}) = \frac{n}{d} \Leftrightarrow (d\frac{n}{d}, r\frac{n}{d}) = \frac{n}{d} \Leftrightarrow (r,d) = 1.$$

(iii)- Soit $d \mid n$. Un élément de G est d'ordre d si, et seulement s'il est de la forme $a^{rn/d}$, $(r,d) = 1$. Il y a donc exactement $\phi(d)$ éléments d'ordre d dans G. Le sous-groupe engendré par $a^{n/d}$ est d'ordre d, et contient tous les éléments d'ordre d. C'est le seul car, tout sous-groupe de G étant cyclique, tout autre sous-groupe d'ordre d devrait contenir un élément d'ordre d. □

Il résulte du théorème précédent qu'un groupe cyclique d'ordre n admet $\phi(n)$ générateurs.

Corollaire A.3.2. *Soit $n \in \mathbb{N}^*$. Alors*

$$n = \sum_{d|n} \phi(d).$$

Preuve :
Soit G, un groupe cyclique d'ordre n. Chaque élément de G ayant un unique ordre d divisant n, et étant donné qu'il y a $\phi(d)$ éléments d'ordre d, le membre de droite compte le nombre total d'éléments de G. □

Ce corollaire peut permettre un calcul plus aisé de $\phi(n)$.

A.3.2 L'anneau des entiers modulo n : $\mathbb{Z}_n$

Soit a, $b \in \mathbb{Z}$. On dit que a est congru à b modulo n, et on note $a \equiv b \bmod n$, si $n|(a-b)$.

On note $\mathbb{Z}_n$, l'anneau des entiers modulo n. C'est l'anneau quotient de $\mathbb{Z}$ par la relation d'équivalence :

$$a \mathcal{R} b \Leftrightarrow a - b \equiv 0 \bmod n.$$

La classe de $a \in \mathbb{Z}$ est notée $[a]$.
Soit $a \in \mathbb{Z}$. Il existe $q, r \in \mathbb{Z}$, tels que $a = nq + r$, $r < n$. Alors, $[a] = [r]$. Ainsi, chaque entier est congru à un unique résidu dans $\{0, \ldots, n-1\}$. D'où,

$$\mathbb{Z}_n = \{[0], \ldots, [n-1]\}.$$

On notera simplement $0, \ldots, n-1$, les éléments de $\mathbb{Z}_n$, étant entendu qu'il s'agit en fait des plus petits représentants positifs de leur classe d'équivalence.
Les additions, soustractions, multiplications dans $\mathbb{Z}_n$ sont effectuées modulo n.

Exemple :
$\mathbb{Z}_{25} = \{0, \ldots, 24\}$. On a $19 + 8 \equiv 2 \bmod 25$, $6 \times 8 \equiv 23 \bmod 25$.

Proposition A.3.2. *Le groupe $(\mathbb{Z}_n, +)$ est cyclique d'ordre n.*

Preuve :
Il est clair que $[1]$ engendre $\mathbb{Z}_n$. □

Soit $a \in \mathbb{Z}_n$. a est *inversible* modulo n, si l'équation $ax \equiv 1 \bmod n$ admet une solution dans $\mathbb{Z}_n$. Dans ce cas, on note a^{-1}, l'inverse de a dans $\mathbb{Z}_n$.

Proposition A.3.3. *Soit $a \in \mathbb{Z}_n$. a est inversible modulo n si, et seulement si, $(a, n) = 1$.*

Preuve :
Supposons a inversible dans $\mathbb{Z}_n$. Alors il existe $x \in \mathbb{Z}_n$, et $k \in \mathbb{Z}$, tels que $ax + kn = 1$. Tout diviseur de a et n divise alors 1. Réciproquement, si $(a, n) = 1$, il existe $u, v \in \mathbb{Z}$ tels que $au + nv = 1$. a est donc inversible modulo n, d'inverse $a^{-1} = u$. □

A.3.3 Le groupe multiplicatif $\mathbb{Z}_n^*$

Soit $n \in \mathbb{N}^*$. De la proposition précédente, on déduit que l'ensemble $\{a \in \mathbb{Z}_n,\ (a,n) = 1\}$, muni de la multiplication modulo n, forme un groupe d'ordre $\phi(n)$, noté $\mathbb{Z}_n^*$. L'élément neutre de ce groupe est [1].

Exemple :
$\mathbb{Z}_{21}^* = \{1, 2, 4, 5, 8, 10, 11, 13, 16, 17, 19, 20\}$. On a $\phi(21) = \phi(3)\phi(7) = 2 \times 6 = 12$. On donne ci-après les ordres des éléments de $\mathbb{Z}_{21}^*$:

a	1	2	4	5	8	10	11	13	16	17	19	20
$ord(a)$	1	6	3	6	2	6	6	2	3	6	6	2

On remarque que le plus grand ordre possible pour un élément de $\mathbb{Z}_{21}^*$ est 6. Plus généralement, si $n = pq$, p et q premiers, alors l'ordre maximal d'un élément de $\mathbb{Z}_n^*$ est $p.p.c.m(p-1, q-1)$.

Le théorème suivant est à la base du cryptosystème RSA :

Théorème A.3.4. *(Euler)*
Soit $n \in \mathbb{N}^*$, et $a \in \mathbb{Z}$, $(a,n) = 1$. Alors

$$a^{\phi(n)} \equiv 1 \bmod n.$$

Preuve :
Soit $a \in \mathbb{Z}$, $(a,n) = 1$.
– Si $a < n$, alors $a \in \mathbb{Z}_n^*$. Le résultat s'en déduit, puisque $|\mathbb{Z}_n^*| = \phi(n)$.
– Si $a > n$, alors il existe $\tilde{a} \in \mathbb{Z}_n^*$, tel que $\tilde{a} \equiv a \bmod n$. On a alors $a^{\phi(n)} \equiv \tilde{a}^{\phi(n)} \equiv 1 \bmod n$.

□

Il est à noter que, d'après ce théorème, l'inverse de a modulo n peut être calculé par $a^{\phi(n)-1} \bmod n$.

Corollaire A.3.3. *(Fermat)*
Soit p, un premier et $a \in \mathbb{Z}$, $(a,p) = 1$. Alors

$$a^{p-1} \equiv 1 \bmod p.$$

Remarque : Soit p premier. On a, pour tout $a \in \mathbb{Z}$, $a^p \equiv a \bmod p$.

Théorème A.3.5. *Si n est premier, alors $\mathbb{Z}_n^*$ est cyclique.*

La preuve de ce résultat sera donnée dans la section suivante, où l'on montrera plus généralement que le groupe multiplicatif d'un corps fini est cyclique.

Exemple :
$\mathbb{Z}_{19}^* = \{1, \ldots, 18\}$ est cyclique d'ordre 18, et admet $\alpha = 2$ pour générateur. A titre d'illustration du théorème A.3.3 (iii), on donne ci-après les sous-groupes de $\mathbb{Z}_{19}^*$, leurs générateurs, ainsi que leur ordre.

Sous-groupe	Générateurs	Ordre
$\{1\}$	1	1
$\{1, 18\}$	18	2
$\{1, 7, 11\}$	$7, 11$	3
$\{1, 7, 8, 11, 12, 18\}$	$8, 12$	6
$\{1, 4, 5, 6, 7, 9, 11, 16, 17\}$	$4, 5, 6, 9, 16, 17$	9
$\{1, \dots, 18\}$	$2, 3, 10, 13, 14, 15$	18

A.3.4 Corps finis

Dans cette partie, seules quelques preuves seront données. Pour un exposé plus complet, on pourra se référer à l'ouvrage "Finite fields", de R. Lidl et H. Niederreiter.

Soit K, un corps. S'il existe un entier strictement positif n tel que $nx = 0\ \forall x \in K$, alors le plus petit tel entier n est appelé *caractéristique* de K. Sinon, K est dit de caractéristique nulle.

Un corps fini est un corps contenant un nombre fini d'éléments. Il est donc nécessairement de caractéristique non nulle. Mieux :

Théorème A.3.6. *Tout corps fini est de caractéristique p, p premier.*

Un exemple de corps de caractéristique p est donné par le théorème suivant

Théorème A.3.7. *Si p est premier, l'anneau quotient $\mathbb{Z}_p$ est un corps, que l'on note $\mathbb{F}_p$.*

Preuve :
Soit $a, b \in \mathbb{Z}$. Alors $[a][b] = [ab] = 0 \Leftrightarrow p \,|\, ab \Leftrightarrow (p \,|\, a$ ou $p \,|\, b) \Leftrightarrow ([a] = 0$ ou $[b] = 0)$. Donc $\mathbb{Z}_p$ ne contient pas de diviseur non nul de zéro. □

Comme on le verra par la suite, tout corps à p éléments est isomorphe à $\mathbb{F}_p$.

Soit L, un corps, et K, un sous-corps[11] de L. Alors L est une *extension (de corps)* de K. K est un *corps premier* s'il ne contient pas de sous-corps (au sens de l'inclusion stricte). L'intersection de tous les sous-corps de L est appelé le *sous-corps premier* de L (c'est évidemment un corps premier).

Théorème A.3.8. *Le sous-corps premier d'un corps L de caractéristique p (resp. 0) est isomorphe à $\mathbb{F}_p$ (resp. $\mathbb{Q}$).*

Un corps fini L contenant un sous-corps K peut être vu comme un espace vectoriel de dimension finie sur K. Sa dimension est alors notée $[L : K]$, et est aussi appelée *degré de (l'extension) L*. Le théorème suivant lie le nombre d'éléments de L à $[L : K]$, où K est le sous-corps premier de L :

Théorème A.3.9. *Un corps fini L possède p^m éléments, où le premier p est la caractéristique de L, et l'entier m est le degré de L sur son sous-corps premier.*

[11] i.e. un sous-ensemble de L qui est un corps pour les opérations induites par L.

Preuve :
L étant fini, il est de caractéristique p premier. Soit $K \cong \mathbb{F}_p$, le sous-corps premier de L, et soit $(b_1, \ldots, b_m)$, une base de L sur K. Chaque élément de L admet une représentation unique de la forme $a_1 b_1 + \ldots + a_m b_m$, $a_i \in K$. Chaque a_i pouvant prendre p valeurs, il en résulte que L possède exactement p^m éléments. □

A ce stade, on ne sait toujours pas si un corps à p^m éléments existe. Donnons à présent une méthode de construction d'un tel corps, à partir du corps $\mathbb{F}_p$.

Pour cela, on rappelle la notion d'irréductibilité : soit K, un corps. Un polynôme $g(x) \in K[x]$ de degré $m \geq 0$ est dit irréductible dans $K[x]$, si $g(x) = b(x)c(x)$, $b(x), c(x) \in K[x]$ implique $b(x)$ ou $c(x)$ appartient à K.
On note $(g(x))$, l'idéal de l'anneau $K[x]$ engendré par $g(x)$, i.e. $(g(x)) = \{f(x)g(x),\ f(x) \in K[x]\}$.

On rappelle également la division Euclidienne de $f(x)$ par $g(x)$, $\deg f(x) \geq \deg g(x)$, les deux polynômes étant à coefficients dans un corps K : $\exists\, q(x), r(x) \in K[x]$, tels que $f(x) = q(x)g(x) + r(x),\ \deg r(x) < \deg g(x)$.

Le résultat suivant est à la base de la construction :

Théorème A.3.10. *Soit $g(x) \in K[x]$. L'anneau quotient de polynômes $K[x]/(g(x))$ est un corps si, et seulement si, $g(x)$ est irréductible dans $K[x]$.*

Soit $g(x)$, un polynôme irréductible dans $\mathbb{F}_p[x]$, de degré m. Par définition,

$$\mathbb{F}_p[x]/(g(x)) = \{f(x) \in \mathbb{F}_p[x],\ \deg(f(x)) < m\},$$

donc le nombre d'éléments de $\mathbb{F}_p[x]/(g(x))$ est égal au nombre de polynômes de degré strictement inférieur à m à coefficients dans $\mathbb{F}_p$, soit p^m.
Sous réserve qu'un tel $g(x)$ existe (voir théorème A.3.11), on obtient ainsi un corps à p^m éléments.

Cette construction permet de déduire une représentation effective des éléments du corps. Pour cela, précisons les notions d'élément algébrique et de polynôme minimal :

Définition A.9. *Soit K, un corps fini, et L, une extension de K.*
- *Un élément $\alpha \in L$ est dit* algébrique *sur K s'il est solution d'une équation polynômiale (non triviale) à coefficients dans K, i.e. s'il existe $a_0, \ldots, a_l$, $a_i \in K$ non tous nuls, tels que*
$$a_l \alpha^l + \ldots + a_1 \alpha + a_0 = 0.$$
- *Soit $\alpha \in L$, un élément algébrique sur K. Alors le polynôme de $K[x]$ unitaire*[12] *engendrant l'idéal $\{h(x) \in K[x],\ h(\alpha) = 0\} \subset K[x]$, est appelé* polynôme minimal de α sur K.

Le polynôme minimal vérifie les propriétés suivantes :

[12] i.e. de coefficient dominant égal à 1.

Proposition A.3.4. *Soit K, un corps fini, et L, une extension de K. Soit $\alpha \in L$, un élément algébrique sur K, et $g(x) \in K[x]$, le polynôme minimal de α sur K. Alors*
- *$g(x)$ est irréductible dans $K[x]$.*
- *Pour tout $f(x) \in K[x]$, on a $f(\alpha) = 0$ si, et seulement si, $g(x)$ divise $f(x)$.*
- *$g(x)$ est le polynôme unitaire de $K[x]$ de plus petit degré admettant α pour racine.*

Soit maintenant L, une extension[13] de $\mathbb{F}_p$, et $\alpha \in L$, un élément algébrique sur $\mathbb{F}_p$. Soit $g(x) \in \mathbb{F}_p[x]$, le polynôme minimal de α sur $\mathbb{F}_p$, et m, le degré de $g(x)$.
On note $\mathbb{F}_p(\alpha)$, le plus petit corps contenant $\mathbb{F}_p$ et α. Considérons alors l'application "évaluation en α" :

$$\begin{array}{rcl} \Phi\ : \mathbb{F}_p[x] & \longrightarrow & L \\ f(x) & \longmapsto & f(\alpha), \end{array}$$

D'après le choix de $g(x)$, le noyau de Φ est $(g(x))$. Donc $\mathrm{Im}\Phi \cong \mathbb{F}_p[x]/(g(x))$. De plus, il est facile de voir que $\mathrm{Im}\Phi = \mathbb{F}_p(\alpha)$.
Ainsi, tout élément d'une extension de degré m de $\mathbb{F}_p$ s'écrit comme polynôme en α de degré strictement inférieur à m.

La validité de la construction d'une extension de corps de degré n est assurée par le

Théorème A.3.11. *Pour tout corps fini F, et tout entier positif n, il existe un polynôme irréductible dans $F[x]$, de degré n.*

Notons que tout polynôme unitaire irréductible dans $F[x]$ est le polynôme minimal d'un élément α dans une extension de F.

Si L est un corps à $q = p^m$ éléments, tout $a \in F$ vérifie $a^q = a$. En effet, c'est clair si $a = 0$. D'autre part, l'ensemble des éléments non nuls de L, muni de la multiplication, forme un groupe d'ordre $q-1$, donc $a^{q-1} = 1$, $\forall a \in L \setminus \{0\}$.
Considérons le polynôme $Q(x) = x^q - x$ de $K[x]$. En tant que polynôme de $L[x]$, $Q(x)$ admet au plus q racines[14] dans L. Mais les éléments de L sont tous racines de $Q(x)$. Donc $Q(x)$ se factorise complètement dans L :

$$x^q - x = \prod_{a \in L} (x - a).$$

On peut alors énoncer le résultat suivant, qui résume l'existence et l'unicité des corps finis.

Théorème A.3.12. *Pour tout premier p et tout entier positif m, il existe un corps fini à p^m éléments. De plus, tout corps à p^m éléments est isomorphe au corps des racines du polynôme $x^{p^m} - x$ sur $\mathbb{F}_p$.*

Etant donné l'unicité (à isomorphisme près) énoncée dans le théorème ci-dessus, on parlera *du* corps à $q = p^m$ éléments, que l'on notera $\mathbb{F}_q$.

[13]Par exemple $\mathbb{R}$ ou $\mathbb{C}$.
[14]Si L est un corps, un polynôme non nul de degré n dans $F[x]$ admet au plus n racines dans L.

Exemple :
$\mathbb{F}_{25}$ est le corps à 25 éléments construit à partir d'un polynôme $g(x)$ irréductible de degré 2, à coefficients dans $\mathbb{F}_5$. On peut prendre par exemple $g(x) = x^2 - 2 \in \mathbb{F}_5[x]$. Une racine de $g(x)$ est $\alpha = \sqrt{2}$. $\mathbb{F}_{25}$ étant un espace vectoriel de dimension 2 sur $\mathbb{F}_5$, il est formé des vecteurs (a, b), $a, b \in \mathbb{F}_5$, ou de manière équivalente, des polynômes en α de la forme $a\alpha + b$. L'addition de deux éléments de $\mathbb{F}_{25}$, $a\alpha + b$ et $a'\alpha + b'$, est l'élément $(a + a')\alpha + (b + b')$, les additions étant ici modulo 5. En utilisant le fait que $g(\alpha) = 0$, la multiplication est définie par :

$$(a\alpha + b)(a'\alpha + b') = (ab' + a'b)\alpha + (bb' + 2aa').$$

On indique ci-après la forme des sous-corps de $\mathbb{F}_q$:

Proposition A.3.5. *Soit $\mathbb{F}_q$, le corps fini à $q = p^m$ éléments. Tout sous-corps de $\mathbb{F}_q$ est d'ordre p^r, où $r \in \mathbb{N}$, $r \mid m$. Réciproquement, pour tout diviseur positif r de m, il existe un unique sous-corps de $\mathbb{F}_q$ à p^r éléments.*

Notons $\mathbb{F}_q^*$, le groupe multiplicatif du corps $\mathbb{F}_q$.

Théorème A.3.13. *$\mathbb{F}_q^*$ est un groupe cyclique d'ordre $q - 1$.*

Preuve :
Il suffit de prouver l'existence d'un élément d'ordre $q - 1$.
Soit $q - 1 = \prod_{i=1}^{k} p_i^{e_i}$, la décomposition de $q - 1$ en facteurs premiers, $e_i > 0$, $1 \le i \le k$. Pour $1 \le i \le k$, le polynôme $x^{(q-1)/p_i} - 1$ a au plus $(q - 1)/p_i$ racines dans $\mathbb{F}_q$. On a $(q - 1)/p_i < q - 1$, donc il existe un élément non nul $a_i \in \mathbb{F}_q$, qui ne soit pas racine de ce polynôme. Posons $b_i = a_i^{(q-1)/p_i^{e_i}}$. Alors b_i est d'ordre exactement $p_i^{e_i}$. En effet, $b_i^{p_i^{e_i}} = 1$, donc l'ordre de b_i divise $p_i^{e_i}$, et $b_i^{p_i^{e_i - 1}} = a_i^{(q-1)/p_i} \neq 1$. On laisse en exercice la vérification du fait que l'élément $b = \prod_{i=1}^{k} b_i$ est d'ordre $q - 1$. $\square$

Cette preuve donne une méthode de calcul d'un générateur de $\mathbb{F}_q^*$. Un générateur de $\mathbb{F}_q^*$ est appelé un *élément primitif* de $\mathbb{F}_q$.

Mentionnons encore une propriété fondamentale du corps $\mathbb{F}_{p^m}$.

Théorème A.3.14. *Soit $f(x)$, un polynôme irréductible dans $\mathbb{F}_p[x]$, de degré m. Alors $\mathbb{F}_{p^m}$ est le corps des racines de $f(x)$, et, si α désigne une racine de $f(x)$ dans $\mathbb{F}_{p^m}$, toutes ses racines sont données par les m éléments distincts α, α^p, ..., $\alpha^{p^{m-1}}$ de $\mathbb{F}_{p^m}$. En particulier, toutes les racines de $f(x)$ ont même ordre*[15].

Ordre d'un polynôme et polynômes primitifs

Outre le degré, il existe un paramètre important attaché à un polynôme non nul sur un corps fini : son ordre. Dans la suite, q désigne une puissance de premier, soit $q = p^r$; en toute généralité, on se place dans le corps fini $\mathbb{F}_q$ et son extension de degré m, $\mathbb{F}_{q^m}$.

[15]En tant qu'éléments de $\mathbb{F}_{p^m}$.

Théorème A.3.15. *Soit $f(x) \in \mathbb{F}_q[x]$, un polynôme de degré $m \geq 1$, avec $f(0) \neq 0$. Alors il existe $e \in \mathbb{N}^*$, $e \leq q^m - 1$, tel que $f(x)$ divise $x^e - 1$.*

Preuve :
L'anneau quotient $\mathbb{F}_q[x]/(f(x))$ contient q^m classes, dont $q^m - 1$ classes non nulles. D'autre part, les q^m classes $[x^j] = x^j + (f(x))$, $0 \leq j \leq q^m - 1$, sont toutes non nulles. En effet, $x^j \equiv 0 \bmod f(x) \Leftrightarrow f(x) \mid x^j \Rightarrow f(0) = 0$. Donc il existe $0 \leq e < j \leq q^m - 1$, tels que $[x^e] = [x^j]$, i.e. $f(x) \mid x^{j-e}(x^e - 1)$. Mais $f(x) \not| x^{j-e}$. □

Soit $f(x)$ comme dans le théorème précédent. Le plus petit entier e tel que $f(x)$ divise $x^e - 1$ est appelé *l'ordre* de $f(x)$. Si $f(0) = 0$, l'ordre de $f(x)$ est défini comme étant égal à l'ordre de $g(x)$, où l'on a posé $f(x) = x^k g(x)$, $g(0) \neq 0$.

Du théorème A.3.14, énoncé dans le cas de $\mathbb{F}_{p^m}$ mais valable pour tout corps $\mathbb{F}_{q^m}$, on déduit :

Proposition A.3.6. *Soit $f(x)$, un polynôme irréductible dans $\mathbb{F}_q[x]$, de degré m, avec $f(0) \neq 0$. Alors l'ordre de $f(x)$ est égal à l'ordre d'une quelconque de ses racines dans le groupe multiplicatif $\mathbb{F}^*_{q^m}$. En particulier, l'ordre de $f(x)$ divise $q^m - 1$.*

Un polynôme $f(x) \in \mathbb{F}_q[x]$ de degré $m \geq 1$, est dit *primitif sur* $\mathbb{F}_q$, si c'est le polynôme minimal (sur $\mathbb{F}_q$) d'un élément primitif de $\mathbb{F}_{q^m}$. On a alors la caractérisation suivante :

Proposition A.3.7. *Un polynôme $f(x) \in \mathbb{F}_q[x]$ de degré $m \geq 1$, est primitif sur $\mathbb{F}_q$ si, et seulement si*
- *$f(x)$ est unitaire,*
- *$f(0) \neq 0$,*
- *l'ordre de $f(x)$ est $q^m - 1$.*

A.3.5 Algorithmes élémentaires

La plupart des techniques cryptographiques font appel à des opérations modulaires. Pour évaluer la complexité des algorithmes présentés dans ce cours, on s'appuie sur les complexités[16] des opérations de base suivantes (a et b sont inférieurs à n) :

$$\begin{array}{cc}
(a \pm b) \bmod n & O(\lg n) \\
a \times b \bmod n & O((\lg n)^2) \\
a^{-1} \bmod n & O((\lg n)^2) \\
a^k \bmod n,\ k < n & O((\lg n)^3)
\end{array}$$

Des algorithmes effectuant une multiplication modulaire peuvent être trouvés dans Knuth, "The art of computer programming", vol.2. On donne ci-après une méthode de calcul d'un inverse modulaire, ainsi qu'un algorithme d'exponentiation modulaire.

[16]En nombre d'opérations binaires.

Algorithme d'Euclide étendu

L'algorithme d'Euclide de calcul du p.g.c.d. de deux entiers a et b peut être modifié pour calculer un inverse modulaire.
On rappelle que cet algorithme - basé sur l'observation[17] $(a, b) = (b, a \bmod b)$ - consiste à itérer cette propriété jusqu'à ce que le reste $a \bmod b$ soit nul. Le dernier reste non nul fournit alors le p.g.c.d. de a et b.

Soit $d = (a, b)$. Par l'identité de Bezout, il existe $u,\ v \in \mathbb{Z}$, tels que $au+bv = d$. L'algorithme d'Euclide étendu permet d'obtenir u et v.
Pour trouver l'inverse de b modulo n lorsque $(b, n) = 1$, il suffit de l'appliquer avec $a = n$: l'entier v fournit alors l'inverse cherché.

Le principe de l'algorithme est le suivant : Notons $r_0 = a,\ r_1 = b$. Soit $\{q_j\}_{1\leq j\leq m}$, la suite de quotients obtenus par l'algorithme d'Euclide, i.e. :

$$r_0 = q_1r_1 + r_2, 0 < r_2 < r_1, \ldots, r_{m-2} = q_{m-1}r_{m-1} + r_m,\ 0 < r_m < r_{m-1},\ r_{m-1} = q_mr_m.$$

On a donc $d = r_m$. Soit $\{t_j\}_{1\leq j\leq m}$, la suite définie par :

$$t_0 = 0,\ t_1 = 1,\ \text{et } t_j = t_{j-2} - q_{j-1}t_{j-1} \bmod r_0,\ 2 \leq j \leq m.$$

Alors, on peut montrer (par récurrence) que les t_j vérifient

$$r_j \equiv t_jr_1 \bmod r_0,\ 0 \leq j \leq m.$$

Ainsi, $t_m = v$, avec les notations ci-dessus.
Si $a, b \leq n$, on peut vérifier que la complexité de cet algorithme est en $O((\lg n)^2)$.

Algorithme "square and multiply" d'exponentiation modulaire

Le calcul de $a^k \bmod n$, $a, k < n$, intervient fréquemment en cryptographie (exponentiation El Gamal en base g, RSA ...).
Soit $k = \sum_{i=0}^{t} k_i2^i$, $k_i \in \mathbb{F}_2$, la représentation binaire de k, avec $k_t = 1$. On remarque que

$$a^k = \prod_{i=0}^{t} a^{k_i2^i} = (a^{2^0})^{k_0} \ldots (a^{2^t})^{k_t}.$$

Ceci conduit à l'algorithme suivant :

Square-and-multiply(a, k)
- $B \leftarrow 1$
- Pour $i = t$ à 0
  - $B \leftarrow B^2 \bmod n$
  - Si $k_i = 1$, $B \leftarrow B \times a \bmod n$

[17] On suppose $a > b$.

– retourner (B).

Cet algorithme nécessite $t+1$ élévations au carré, et $w(k)$ multiplications, où $w(k) = |\{0 \leq i \leq t,\ k_i = 1\}|$ est le poids de k. Si k est de taille n et choisi aléatoirement, $w(k)$ vaut en moyenne $(\lg n)/2$. Le calcul de $a^k \bmod n$ utilise alors environ $3(\lg n)/2$ multiplications[18] modulaires, et est donc bien en $(\lg n)^3$ opérations binaires.

Théorème des restes chinois

Considérons l'application

$$\begin{array}{rcl} f\ : \mathbb{Z}_m & \longrightarrow & \mathbb{Z}_{m_1} \times \ldots \times \mathbb{Z}_{m_k} \\ x & \longmapsto & (x \bmod m_1, \ldots, x \bmod m_k), \end{array}$$

où les m_i sont des entiers premiers entre eux deux à deux, et $m = \prod_{i=1}^k m_i$.
Etant donné un système de congruences $\{x = a_i \bmod m_i\}_{1\leq i\leq k}$, $a_i \in \mathbb{Z}_{m_i}$, $1 \leq i \leq k$, le théorème des restes chinois[19] donne une construction d'une solution x, fournissant ainsi une preuve d'existence d'un antécédent par f de tout élément de $\mathbb{Z}_{m_1} \times \ldots \times \mathbb{Z}_{m_k}$. L'unicité de la solution vient alors du fait que, f étant une surjection entre deux ensembles de même cardinalité, elle est bijective.
La solution est calculée de la manière suivante : soit

$$N_i = \frac{m}{m_i}, \text{ et } S_i = N_i^{-1} \bmod m_i,\ 1 \leq i \leq k.$$

Alors, $x = \sum_{i=1}^k a_i N_i S_i \bmod m$ est une solution du système. En effet, si $j \neq i$, $a_i N_i S_i \equiv 0 \bmod m_j$, puisque $m_j \mid N_i$, donc

$$x \bmod m_j \equiv a_j N_j S_j \bmod m_j \equiv a_j N_j (N_j^{-1} \bmod m_j) \bmod m_j \equiv a_j,$$

et ceci pour tout $1 \leq j \leq k$.
Cet algorithme sert à accélerer les calculs dans le déchiffrement (ou la signature) RSA : pour calculer $m = c^d \bmod n$, $n = pq$, on calcule $a_p = c^{d \bmod (p-1)} \bmod p$, $a_q = c^{d \bmod (q-1)} \bmod q$, et on combine a_p et a_q par CRT pour retrouver m. Le calcul de a_p et a_q est beaucoup moins coûteux que le calcul de $c^d \bmod n$, car les modules sont de tailles moitié moindres.

[18] Si on considère qu'une multiplication et un carré modulaires ont la même complexité.
[19] noté CRT dans le cours.