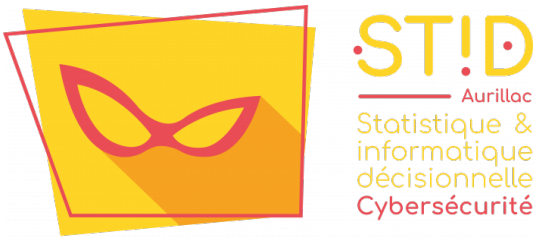

Cours et exercices

Université Clermont Auvergne : IUT Aurillac

Département STID

Introduction à la cryptologie

Écris par :
Axel DURBET

Derniers correctifs le : 3 mars 2022

Table des matières

Chapitre 1

Cryptologie et Stéganographie, le béa ba

Dans ce chapitre, nous parlerons de stéganographie et de cryptologie. Le but est de comprendre à quoi ça sert, pourquoi c'est différent et en quoi utiliser les deux, c'est bien.

1.1 La stéganographie

La stéganographie est l'art de la dissimulation de communication. Son objectif est de pouvoir dissimuler des données qui doivent être tenues secrètes dans un support paraissant anodin. Ces données ne doivent pouvoir être récupérées que par une personne ayant connaissance de l'algorithme utilisé pour la dissimulation, ainsi que de l'éventuel clé saisie. Il existe de nombreuses méthodes en stéganographie pour cacher un message comme montrer dans la figure 1.1.

Ici, le texte est caché dans n'importe quelle forme et seul l'image, l'audio ou la vidéo est visible pour les personnes.

Quelques exemples de messages cachés

Exercice 1.1.1 (Rootme : Steganomobile) *Après avoir extrait les données d'un téléphone mobile retrouvé sur les lieux d'un crime, les enquêteurs ont récupéré cette suite de chiffres... Peut-être un numéro de téléphone ?*
Les chiffres : $222 - 33 - 555 - 555 - 7 - 44 - 666 - 66 - 33$

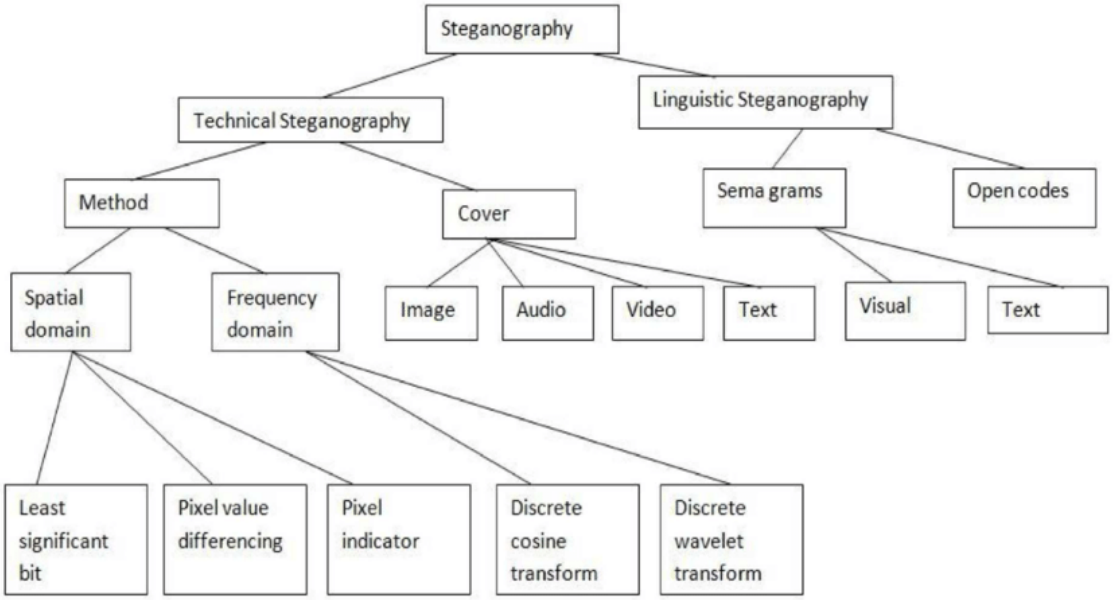

FIGURE 1.1 – Les sous familles de la stéganographie.

Exercice 1.1.2 (Rootme : George et Alfred) *On trouve dans la littérature des exemples raffinés de lettres utilisant la stéganographie. Voici un exemple connu de correspondance entre George Sand et Alfred de Musset où des messages intimes sont camouflés.*

Le message de George Sand :

Je suis très émue de vous dire que j'ai
bien compris, l'autre jour, que vous avez
toujours une envie folle de me faire
danser. Je garde un souvenir de votre
baiser et je voudrais que ce soit
là une preuve que je puisse être aimée
par vous. Je suis prête à vous montrer mon
Affection toute désintéressée et sans cal-
cul. Si vous voulez me voir ainsi
dévoiler, sans aucun artifice mon âme
toute nue, daignez donc me faire une visite
Et nous causerons en amis et en chemin.

Je vous prouverai que je suis la femme
sincère capable de vous offrir l'affection
la plus profonde et la plus étroite
Amitié, en un mot, la meilleure amie
que vous puissiez rêver. Puisque votre
âme est libre, alors que l'abandon où je
vis est bien long, bien dur et bien souvent
pénible, ami très cher, j'ai le coeur
gros, accourez vite et venez me le
faire oublier. À l'amour, je veux me sou-
mettre.

Alfred de Musset a répondu ceci :

Quand je vous jure, hélas, un éternel hommage
Voulez-vous qu'un instant je change de langage
Que ne puis-je, avec vous, goûter le vrai bonheur
Je vous aime, ô ma belle, et ma plume en délire
Couche sur le papier ce que je n'ose dire
Avec soin, de mes vers, lisez le premier mot
Vous saurez quel remède apporter à mes maux.

De la même manière George Sand a répondu ceci :

Cette grande faveur que votre ardeur réclame
Nuit peut-être à l'honneur, mais répond à ma flamme.

Exercice 1.1.3 (**Rootme : Point à la ligne**) *"Rien de trop est un point dont on parle sans cesse et qu'on n'observe point." Trouvez le message secret dans la figure 1.2.*

Exercice 1.1.4 (**Chimie**) *Un chimiste envoie un message "3.1 2.2 15 18.1 81.1 37.1 8 7 10.2 55.2 22.1 27 7" à un autre chimiste. Ils ont dû utiliser un objet de leur quotidien. Trouvez le message caché.*

Plus généralement, allez faire un tour sur root-me pour en apprendre plus sur les méthodes utilisées en stéganographie.

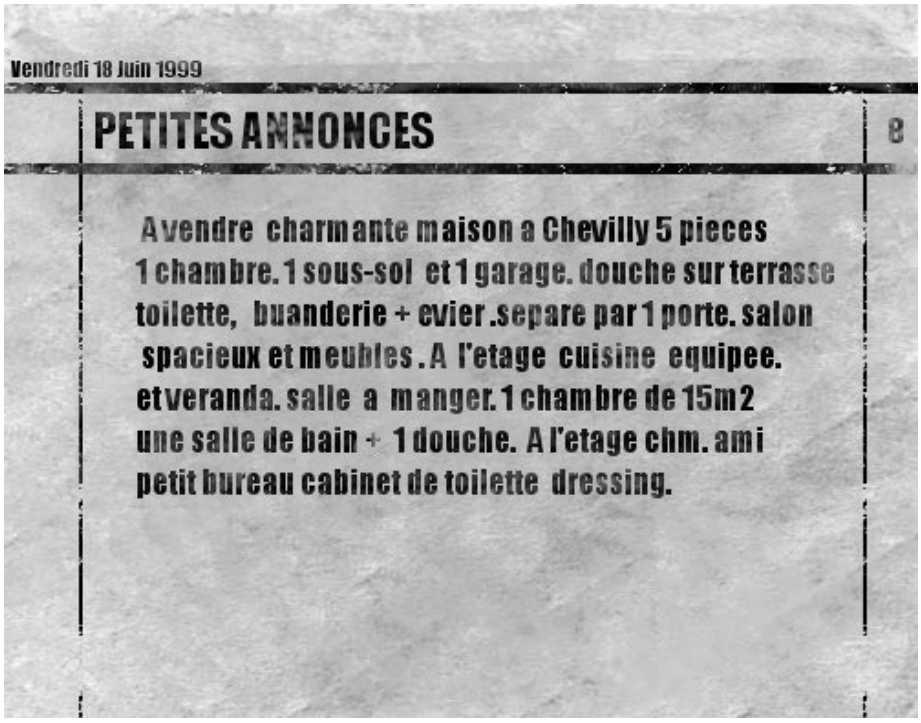

FIGURE 1.2 –

1.2 La cryptographie

La cryptologie est la "science du secret". Elle englobe la cryptographie littéralement "l'écriture secrète" et la cryptanalyse, l'analyse de cette dernière. C'est une discipline très ancienne, car les Spartiates l'utilisaient déjà (la scytale). Cependant, ce n'est que récemment que nous avons fait d'énormes progrès dans le domaine. Cette discipline est liée à beaucoup d'autres, par exemple l'arithmétique modulaire, l'algèbre, la théorie de la complexité, la théorie de l'information ou encore les codes correcteurs d'erreurs.

La cryptographie répond à de nombreuses attentes (voir Figure 1.3) dont les trois principales sont :

1. L'intégrité d'un message.
2. L'authenticité d'un message.
3. La confidentialité d'un message.

Parfois, on demande des propriétés supplémentaires comme la "Non-répudiation" (On ne peut pas nier avoir envoyé un message).

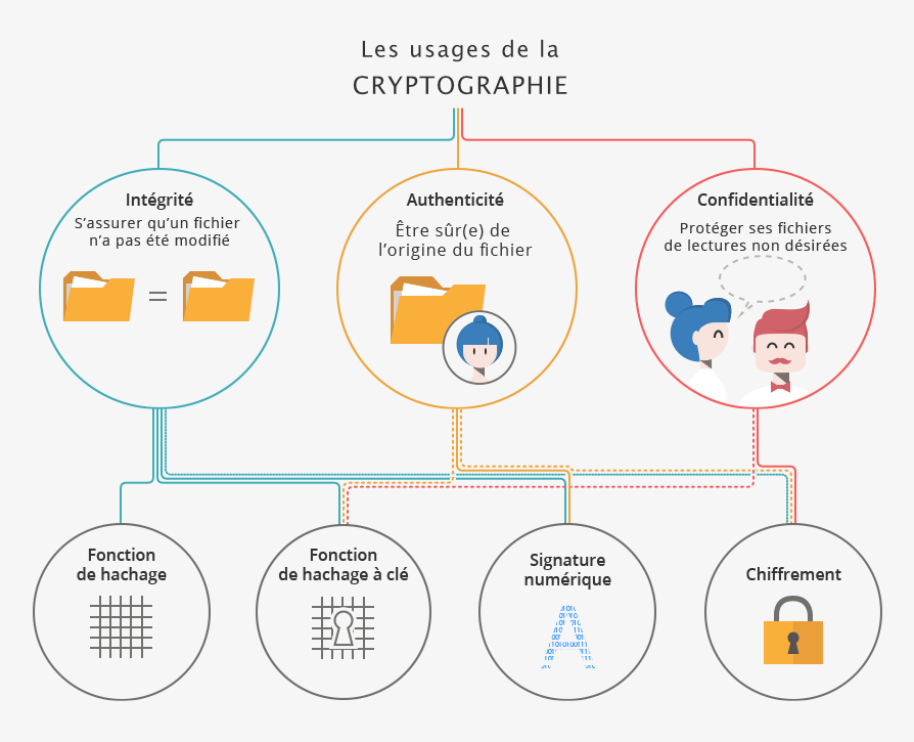

FIGURE 1.3 – Les usages de la cryptographie (CNIL)

Exercice 1.2.1 *Proposez des exemples de situation nécessitant chacun des quatre objectifs.*

Il existe deux types d'attaques qui motivent les propriétés ci-dessus : les attaques passives et actives.

Définition 1.2.1 (Les attaques passives) *L'attaquant intercepte les messages transmis avec l'intention de lire et d'analyser les messages sans les modifier.*

Exemple 1.2.0.1 *Eve écoute les conversations entre Alice et Bob, mais n'intervient pas. C'est une menace pour la confidentialité des échanges.*

Exemples de menaces :
— Récupération d'identifiants et de mots de passe
— Récupération de données personnelles : email, âge, sexe, (publicité ciblée)...

Définition 1.2.2 (Les attaques actives) *Dans ce cas de figure, l'attaquant intercepte les messages et modifie les informations.*

Exemple 1.2.0.2 *Eve modifie les conversations entre Alice et Bob. C'est une menace pour l'intégrité des messages.*

Exemples de menaces :
— Destruction de messages.
— Usurpation d'identité.
— Modification du montant d'une transaction bancaire.

Historiquement, la crypto visait à sécuriser les communications et on trouve des traces de l'utilisation de la cryptographie chez les Romains. Les deux participants partagent un secret : la clé K (pour les romains, un bâton d'officier). Celui qui envoie, chiffre le message avec K. Celui qui reçoit, déchiffre le message avec K.

La cryptographie utilise 3 fonctions majeures : *Gen*, *Enc* et *Dec* qui sont respectivement l'algorithme de génération des clés, l'algorithme de chiffrement et de déchiffrement.

Remarque 1.2.0.1 $Dec_k(Enc_k(m)) = Enc_k(Dec_k(m)) = m$ *pour toute clé* k.

Enfin, le principe de base de la cryptographie : le principe de Kerckhoffs.

Définition 1.2.3 (Principe de Kerckhoffs) *Auguste Kerckhoffs : "La méthode de chiffrement ne doit pas être secrète, et elle doit pouvoir tomber entre les mains de l'ennemi sans provoquer de problème".*
Autrement dit, la sécurité d'un système ne doit se baser que sur le caractère secret de la clé.

Pourquoi seulement garder la clé secrète et pas l'algorithme en entier ?

1. Il est plus facile de garder secret une petite clé que tout un algorithme.
2. Si k n'est plus secret, il est plus simple de changer la clé que tout un algorithme.

Dans le monde d'aujourd'hui, les algorithmes sont publiques pour plusieurs raisons :
— Beaucoup de personnes vont regarder/améliorer les algorithmes.
— Il vaut mieux révéler que les failles de sécurité sont trouvées par des gens sympas que par des gens malveillants.
— Le code d'un algo peut être cassé (reverse engineering). La clé (nombre aléatoire) ne fait pas partie du code.
— Un design publique rend possible l'établissement de standard (DES, AES, RSA, SHA, ...).

Et ceux qui ne font pas ça ?

Demandez à Apple ce qu'il est arrivé au système de chiffrement d'iMessage en 2016. Ils ont vite compris la leçon et depuis, leurs algorithmes de chiffrement sont accessibles.

Chapitre 2

Rappels d'arithmétiques

Le but de ce chapitre est de manipuler les théorèmes et définitions de base de l'arithmétique. Au début, on se contentera de l'arithmétique de collège qui nous permettra de préparer l'arithmétique modulaire. Le chapitre est en trois parties, une partie cours, une partie méthode et une partie exercice.

2.1 Définitions, propriétés et théorèmes

Définition 2.1.1 (Divise, multiple, divisible) *Soient a et b deux entiers et $b \neq 0$. Alors, on dit que a est **divisible** par b ou que a est un **multiple** de b ou encore que b **divise** a, s'il existe un entier q tel que $a = bq$. On notera alors $b \mid a$, et on lit b **divise** a. Autrement dit, b **divise** a si $\frac{a}{b}$ est un entier.*

Proposition 2.1.1 *Les propriétés suivantes sont vraies pour tout entier a,b et c non nuls :*

- *Si $c \mid a$ et $c \mid b$ alors, pour tout entier u et v, $c \mid au + bv$.*
- *Si $a \mid b$ et $b \mid a$ alors $a = b$ ou $-b$.*
- *Si $c \mid b$ et $b \mid a$ alors $c \mid a$ (On dit que la relation $\mid$ est transitive).*
- *$a \mid a$ (On dit que la relation $\mid$ est réflexive).*
- *$1 \mid a$ (1 divise tout le monde).*
- *$a \mid 0$ (tout le monde divise 0).*

Exercice 2.1.1 (Preuve de cours) *Prouvez que les propriétés ci-dessus sont correctes.*

Définition 2.1.2 (Nombre premier) *On dit qu'un entier naturel $n \geq 2$ est premier s'il n'est divisible que par 1 et lui-même.*

Exercice 2.1.2 (Cours) *Donnez la liste des nombres premiers plus petits que 15.*

Proposition 2.1.2 *Si n n'est pas un nombre premier alors il est divisible par un nombre premier plus petit que $\sqrt{n}$.*

Théorème 2.1.1 (Décomposition en facteur premier ou DFP) *Tout entier $n \geq 2$ se décompose en un produit de puissance de nombre premier. Autrement dit,*

$$n = p_1^{k_1} \times \cdots \times p_f^{k_f}$$

avec $p_1 < \cdots < p_s$ et les k_i des entiers naturels non nuls. Il est important de noter que cette décomposition est unique.

Exemple 2.1.0.1 $123454321342 = 2 \times 11 \times 563 \times 1279 \times 7793$

Exercice 2.1.3 (Cours) *Écrire la DFP de* $121, 101, 33, 392$.

Théorème 2.1.2 (Division Euclidienne) *Pour tout entier a et b, il existe un unique couple d'entiers (q, r) tel que $a = bq + r$ avec $0 \leq r \leq \mid b \mid$.*
On dit que q est le quotient et r le reste de la division euclidienne de a par b.

Exemple 2.1.0.2 *Pour $a = 101$ et $b = 11$ alors* $101 = \underbrace{9}_{q} \times 11 + \underbrace{2}_{r}$.

Exercice 2.1.4 (Cours) *Écrire la division Euclidienne de* 121 *par* 7, *de* 17 *par* 2 *et de* 33 *par* 44.

Définition 2.1.3 (PGCD) *Soient a et b deux entiers non tous deux nuls. On appelle plus grand commun diviseur, le plus grand entier positif qui divise à la fois a et b. On le note : $PGCD(a, b)$.*

Exercice 2.1.5 (Cours) *Calculez le PGCD de* 15 *et* 35.

Définition 2.1.4 (Premiers entre eux) *Soient a et b deux entiers, on dit que a et b sont premiers entre eux si $PGCD(a, b) = 1$.*

Exercice 2.1.6 (Cours) *Montrez que* 32 *et* 7 *sont premiers entre eux.*

Exercice 2.1.7 (Cours) *Montrez que si $PGCD(a, b) = d$ alors $\frac{a}{d}$ et $\frac{b}{d}$ sont entiers et premiers entre eux.*

Théorème 2.1.3 (Bézout) *Soient a et b deux entiers non tous deux nuls. Si $d = PGCD(a, b)$ alors il existe deux entiers u et v tels que $d = au + bv$. Le couple u et v n'est pas unique.*

Exercice 2.1.8 (Cours) *Soient $a = 111$ et $b = 27$, calculez $d = PGCD(a, b)$ et donnez la forme de tous les couples (u, v) tel que $au + bv = d$.*

2.2 Méthode

2.2.1 Calculer un PGCD

Une façon efficace de calculer un PGCD c'est d'appliquer l'algorithme d'Euclide. Celui-ci consiste à effectuer des divisions Euclidienne successives jusqu'au dernier reste non nul qui sera alors notre PGCD.

Par exemple, si on veut calculer le PGCD de 99 et 27, on procède de la sorte :
On fait la division euclidienne de 99 par 27.

$$99 = 27 \times 3 + 18$$

On reprend notre 27 et on effectue la division Euclidienne avec le reste précédent 18.

$$27 = 18 \times 1 + 9$$

On reprend notre 18 et on effectue la division Euclidienne avec le reste précédent 9.

$$18 = 2 \times 9 + 0$$

Le reste est nul, on s'arrête et on regarde le dernier reste non nul. Celui-ci étant 9, on en déduit que $PGCD(99, 27) = 9$.

2.2.2 Trouver un couple pour l'identité de Bézout

Pour trouver un couple, il existe deux méthodes simples, la première consiste à "remonter" l'algorithme d'Euclide et la deuxième consiste à appliquer une version étendue de l'algorithme d'Euclide.

Méthode 1 : Remonter l'algorithme d'Euclide
On reprend notre exemple précédent $a = 99$ et $b = 27$:

$$a - 3b = 18$$

En utilisant la ligne dessous, on remplace 18 :

$$b = a - 3b + 9$$

$$9 = 4b - a$$

Donc le couple $(-1, 4)$ est solution et $-1 \times 99 + 4 \times 27 = 9$. Pour cette méthode, il est important de faire les calculs doucement et de remplacer une chose à la fois pour éviter les erreurs.

De manière plus formelle :
Pour trouver les coefficients pour deux entiers A et B :
On applique l'algorithme d'Euclide dans le bon sens, donc :

$$\begin{aligned} A &= B \times Q + R \\ A_1 &= B_1 \times Q_1 + R_1 \text{(avec B1=R et A1=B)} \\ &\vdots \\ A_n &= B_n \times Q_n + 0 \end{aligned}$$

On a donc $PGCD(A, B) = B_n$.
Ensuite, on remonte en écrivant $R_{n-1} = A_{n-1} - B_{n-1} \times Q_{n-1}$, et en remplaçant avec les valeurs du dessus. Comme dans l'exemple au-dessus.

Méthode 2 : Algorithme d'Euclide étendu

C'est une méthode plus compliquée, mais plus rapide et fiable que la précédente. On donne un exemple de son fonctionnement dans la figure 2.1

r													=	u	×	a	+	v	×	b
120													=	1	×	120	+	0	×	23
23													=	0	×	120	+	1	×	23
5	=	120	-	5	×	23							=	1	×	120	+	-5	×	23
3	=	23	-	4	×	5	=	1×23	-	4	×	(1×120 - 5×23)	=	-4	×	120	+	21	×	23
2	=	5	-	1	×	3	=	(1×120 - 5×23)	-	1	×	(-4×120 + 21×23)	=	5	×	120	+	-26	×	23
1	=	3	-	1	×	2	=	(-4×120 + 21×23)	-	1	×	(5×120 - 26×23)	=	-9	×	120	+	47	×	23

FIGURE 2.1 – Exemple du fonctionnement de l'algorithme d'Euclide étendu

L'idée principale de l'algorithme est d'effectuer les mêmes étapes que pour l'algorithme d'Euclide, mais en exprimant à chaque itération le reste comme une combinaison linéaire de a et b les deux entiers que l'on étudie. Puisque le dernier reste est le PGCD, celui-ci sera alors exprimé comme une combinaison linéaire de a et b

et on aura notre égalité de Bézout.

Déroulement de l'algorithme
On note $r_0 = a$ et $r_1 = b$. On cherche u et v tels que $PGCD(r0, r1) = u \times r_0 + v \times r_1$.

$$\begin{aligned} r_0 &= q_1 \times r_1 + r_2 \\ r_1 &= q_2 \times r_2 + r_3 \\ r_2 &= u_2 \times r_0 + v_2 \times r_1 \\ r_3 &= u_3 \times r_0 + v_3 \times r_1 \\ &\vdots \\ r_k = u_k \times r_0 + v_k \times r_1 \end{aligned}$$

De cette façon, $PGCD(a, b) = r_k$, $u = u_k$ et $v = v_k$. De cette façon, on a à la fois le PGCD et les coefficients de Bézout.

2.3 Exercices d'applications et de réflexions

Les exercices nommés sont des compléments aux cours, gardez donc une trace de ces exercices, car cela vous sera utile dans la suite.

2.3.1 Sur papier

Exercice 2.3.1 *Montrez que* $PGCD(ab, ac) = a \times PGCD(b, c)$.

Exercice 2.3.2 (Le Lemme de Gauss) *Montrez que, si* $a \mid bc$ *et* $PGCD(a, b) = 1$, *alors* $a \mid c$.

Exercice 2.3.3 *Calculez les PGCD suivants :* $PGCD(46328, 12379)$, $PGCD(34860, 4853)$, $PGCD(42098, 34345)$, $PGCD(30076, 12669, 2)$, $PGCD(4567, 11111111111111)$.

Exercice 2.3.4 *Montrez qu'il existe une infinité de nombres premiers.*
Indice 0 : Par l'absurde.

Exercice 2.3.5 (Vrai ou Faux) *Dites si les assertions suivantes sont vraies ou fausses :*

1. 15 *est divisible par* 5.

2. 6 *divise* 24.
3. 12 *est un multiple de* 24.
4. 42 *est divisible par* 7.
5. 18 *est un multiple de* 6.

Exercice 2.3.6 (Un partage de bonbon pas très équitable) *Julien doit partager 73 bonbons équitablement entre ses 3 sœurs et ses cousins. Lui ne gardera que le reste. Il a 5 cousins. Combien doit-il appeler de cousins pour qu'il lui reste le plus de bonbons ?*

Exercice 2.3.7 *Soit n un entier, montrez que $6 \mid n(n+1)(n+2)$.*

Exercice 2.3.8 *Résoudre les équations suivantes :*

1. $2x + 5y = 3$;
2. $323x - 391y = 612$;
3. $162x + 207y = 27$;
4. $221x + 247y = 15$.

Exercice 2.3.9 (Une drole de propriété) *Démontrer que l'on ne change pas le PGCD de deux entiers en multipliant l'un d'entre eux par un entier premier avec l'autre.*

Exercice 2.3.10 (Magicien ou mathématicien ?) *Je vais deviner votre date de naissance. Prenez votre jour de naissance. Multipliez-le par 31. Prenez votre mois de naissance. Multipliez-le par 12. Ajoutez ces deux nombres. Combien trouvez-vous ? 811 vous me dites ? Et bien vous êtes né un 25 mars ! En plus, c'est vrai. Mais comment a-t-il fait ?*

Exercice 2.3.11 *Montrer que l'équation $x^3 - x^2 + x + 1 = 0$ n'a pas de solutions rationnelles.*
Rappel : Les rationnelles s'écrivent $\frac{a}{b}$ avec $b \neq 0$ et $b \neq 1$.

Exercice 2.3.12 ($\sqrt{2}$ ou la hantise de Pythagore) *Démontrez que $\sqrt{2}$ n'est pas un rationnel.*

2.3.2 Exercice sur Ordinateur

Exercice 2.3.13 *Écrire une fonction divEc qui prend en entrée deux nombres a et b et qui retourne le quotient et le reste de la division Euclidienne de a par b.*

Exercice 2.3.14 *Écrire une fonction PGCD qui prend en entrée deux nombres a et b et qui retourne leurs PGCD.*

Exercice 2.3.15 *Écrire une fonction bezout qui prend en entrée deux nombres a et b et qui retourne u et v tel que $au + bv = PGCD(a, b)$.*

Chapitre 3

Arithmétique modulaire

Le but de ce chapitre est de se familiariser avec l'arithmétique modulaire pour préparer à la cryptographie.

3.1 Définitions, propriétés et théorèmes

En arithmétique modulaire, on travaille dans un espace un peu particulier que l'on appelle communément "anneau". Le plus connu, le plus utilisé et celui dans lequel on va travailler est $\mathbb{Z}/n\mathbb{Z}$. Celui-ci à la particularité d'être un "corps" dans certains cas qui dépendent du n choisi.

3.1.1 Introduction de $\mathbb{Z}/n\mathbb{Z}$

Définition 3.1.1 (Congruence modulo n) *Soit $n \leq 2$. On dit que deux entiers a et b sont congrus modulo n si $a - b$ est un multiple de n, autrement dit s'il existe un entier q tel que $a - b = qn$. On écrit alors $a \equiv b[n]$.*

Définition 3.1.2 (Représentant (ou classe) dans $\mathbb{Z}/n\mathbb{Z}$) *Pour tout entier a, il est représenté dans $\mathbb{Z}/n\mathbb{Z}$ par r le reste de la division euclidienne de a par n. Autrement dit, $a \equiv r[n]$. On dit aussi que a est dans la classe de r modulo n.*

Exercice 3.1.1 (Cours) *Montrez que 7 et 14 sont dans la même classe modulo 2.*

Définition 3.1.3 *$a \equiv b[n]$ si et seulement si a et b ont le même reste dans la division euclidienne par n.*

Les restes possibles dans la division euclidienne par n sont : $0, 1, \ldots, (n-1)$. Il y a donc exactement n classes distinctes modulo n et $\mathbb{Z}/n\mathbb{Z} = \{0, 1, 2, ..., (n-1)[n]\}$.

Il est donc fréquent de représenter $\mathbb{Z}/n\mathbb{Z}$ comme un cercle d'où le nom d'anneau. Les modulos, on les utilise tous les jours sans même y faire attention, pour lire l'heure, pour compter avec certaines unités de mesure (deux douzaines de ...), ... Ainsi, une montre ou une horloge est une parfaite représentation de ce qu'il se passe modulo 12 (voir un exemple ici 3.1).

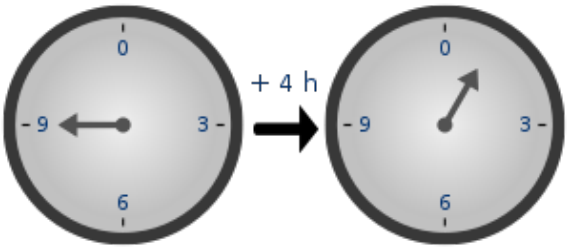

FIGURE 3.1 – Exemple d'une addition modulo 12

De plus, on à quelques relations pratique avec $\equiv$.

Proposition 3.1.1 *Les relations suivantes sont vraies :*
- *Pour tout a entier, $a \equiv a[n]$ ($\equiv$ est Réflexive).*
- *Pour tout a, b des entiers si $a \equiv b[n]$ alors $b \equiv a[n]$ ($\equiv$ est Symétrique).*
- *Pour tout a, b, c des entiers si $a \equiv b[n]$ et $b \equiv c[n]$ alors $a \equiv c[n]$ ($\equiv$ est Transitive).*

3.1.2 Calculs dans $\mathbb{Z}/n\mathbb{Z}$

Définition 3.1.4 *L'addition et la multiplication sont possibles dans $\mathbb{Z}/n\mathbb{Z}$ et elles ont les propriétés suivantes :*
Si $a_1 \equiv r_1[n]$ et $a_2 \equiv r_2[n]$ alors $a_1 + a_2 \equiv r_1 + r_2[n]$ et $a_1 \times a_2 \equiv r_1 \times r_2[n]$.

Pour manipuler des nombres plus petits, on définit donc l'addition et la multiplication de la manière suivante :

Définition 3.1.5 (Addition) *Soient a et b deux entiers alors $a+b[n] = (a[n])+(b[n])[n]$.*

Définition 3.1.6 (Multiplication) *Soient a et b deux entiers alors $a \times b[n] = (a[n]) \times (b[n])[n]$.*

Exercice 3.1.2 (Cours) *Calculez* $23445 + 98765$ *modulo* 10 *et* 456709×567654 *modulo* 2.

Comme $\mathbb{Z}/n\mathbb{Z}$ est un espace étrange, on ne peut pas faire de divisions, mais on peut trouver des inverses qui nous permettent de contourner ce problème.

Définition 3.1.7 (Inversible de $\mathbb{Z}/n\mathbb{Z}$) *Soit a un entier, on dit que a est inversible modulo n ou que $a[n]$ est inversible dans $\mathbb{Z}/n\mathbb{Z}$, s'il existe b tel que $ab = 1[n]$. L'ensemble des inversibles de $\mathbb{Z}/n\mathbb{Z}$ est noté $(\mathbb{Z}/n\mathbb{Z})^\times$. On note $a^{-1}[n]$ l'inverse de $a[n]$.*

Théorème 3.1.1 *a est inversible modulo n si et seulement si $PGCD(a, n) = 1$.*

Exercice 3.1.3 (Cours) *Donnez $(\mathbb{Z}/11\mathbb{Z})^\times$ et $(\mathbb{Z}/6\mathbb{Z})^\times$.*

Corollaire 3.1.1 *Si p est premier alors $(\mathbb{Z}/n\mathbb{Z})^\times = (\mathbb{Z}/n\mathbb{Z})^* = (\mathbb{Z}/n\mathbb{Z})\backslash\{0\}$*

Exercice 3.1.4 (Cours) *Montrez que le corollaire 3.1.1 est vrai.*

Proposition 3.1.2 (Inverse) *Si a est inversible modulo n alors, l'inverse de a modulo n est donné par une relation de Bézout entre a et n.*

Exercice 3.1.5 *Donnez les inverses de* 5,7 *et* 3 *modulo* 11.

Une fois que tout cela est posé, on remarque que lorsque n est premier, on a des propriétés intéressantes :

Proposition 3.1.3 *Si p est un nombre premier, tout élément non nul de $\mathbb{Z}/p\mathbb{Z}$ est inversible.*

Théorème 3.1.2 (Le petit théorème de Fermat) *Soit $a \in \mathbb{Z}/p\mathbb{Z}$ avec p premier, alors $a^{p-1} \equiv 1\,[p]$*

Il se passe quelque chose d'étrange dans $\mathbb{Z}/n\mathbb{Z}$, il peut exister des diviseurs de 0.

Définition 3.1.8 (Diviseurs de 0) *Un diviseur de 0 de $\mathbb{Z}/n\mathbb{Z}$ est un entier a tel que $a \not\equiv 0[n]$ et tel qu'il existe $b \not\equiv 0[n]$ tel que $ab \equiv 0[n]$.*

3.2 Tests de primalité

Maintenant qu'on sais faire un peu d'arithmétiques modulaire, on peut parler des tests de primalité ou comment savoir si un entier est premier.

3.2.1 Les tests probabilistes

Les tests probabilistes ont une chance de se tromper avec une probabilité p mais sont très rapide. De ce fait, on peut les répéter de manière à avoir une bonne probabilité de ne pas se tromper. En effet, si ces tests disent qu'un entier n n'est pas premier alors ils sont fiables à 100%. Cependant, s'ils affirment que n est premier alors il y a une chance qu'ils se trompent. On dit alors que n est probablement premier.

Test de Fermat

Ce test se base sur le petit théorème de Fermat 3.1.2.

Définition 3.2.1 (Test de Fermat) *On effectue les étapes suivantes pour un entier n testé :*

1. *On tire aléatoirement a entre 2 et $n-1$.*
2. *Si $a^{n-1} \equiv 1\,[n]$ alors n est probablement premier.*
3. *Sinon n n'est pas premier par contraposé du petit théorème de Fermat.*

Ce test échoue pour tous les nombres de Carmichaël et il y en a un nombre infini. Par exemple : $561 = 3 \times 11 \times 17$ n'est pas premier et pourtant, pour tout entier $a < 561$, on a $a^5 61 \equiv a\,[561]$.

Cependant, les nombres de Carmichaël sont rares donc la probabilité de tombé dessus tend vers 0 lorsque la taille des entiers augmente.

Test de Miller-Rabin

Ce test est plus performant que le précédent et se base sur la proposition suivante :

Proposition 3.2.1 (Rabin (Admis)) *Soit p un nombre premier et $p - 1 = 2^s \times d$ avec d impair alors, pour tout entier a tel que p ne divise pas a, on a :*

$$a^d \equiv 1\,[p]$$

ou

$$\exists r \in \{0, \ldots, s-1\}\,|a^{d \times 2^r} \equiv -1\,[p]$$

C'est une conséquence du petit théorème de Fermat.

Définition 3.2.2 (Test de Miller-Rabin) *On effectue les étapes suivantes pour un entier n testé :*

1. *On tire aléatoirement a entre 2 et $n-1$.*
2. *On calcule s et d tel que $n-1 = 2^s \times d$ avec d impair.*
3. *Calculer $a^d \equiv x\,[n]$.*
4. *Si $x = 1$ ou $x = n-1$ alors n n'est pas premier.*
5. *Sinon, faire $s-1$ fois :*
   — $x = x^2\,[n]$
   — *Si $x = n-1$ alors n n'est pas premier.*
6. *Si on arrive ici alors n est probablement premier.*

L'avantage, c'est que les nombres de Carmichaël ne passent pas ce test contrairement à celui de Fermat. La probabilité que ce test se trompe est inférieure à $\frac{1}{4}$.

3.2.2 Tests déterministes

Ces tests-là sont plus lourds, mais ne peuvent pas se tromper.

Le test naïf

Le plus coûteux de tous, on va tester si un entier n est divisible par un nombre premier plus petit que $\sqrt{n}$. Si oui alors n n'est pas premier et si non, alors n est premier. En pratique, il n'est pas raisonnable d'utiliser cette méthode sur de grands entiers.

Test de Lucas-Lehmer (admis)

Ce test est moins couteux que le précédent, mais on préfèrera toujours les tests probabilistes plus rapides. Il repose sur le théorème suivant :

Théorème 3.2.1 (Lucas-Lehmer) *Un entier $n > 2$ est premier si et seulement s'il existe a un entier entre 2 et $n-1$ tel que $a^{n-1} \equiv 1\,[n]$ et pour tout facteur q de $n-1$, on a $a^{(n-1)/q} \not\equiv 1\,[n]$.*

3.3 Exercices

Les exercices nommés sont des compléments aux cours, gardez donc une trace de ces exercices, car cela vous sera utile dans la suite.

Exercice 3.3.1 *Répondez aux questions suivantes :*

1. *Écrire les tables d'addition et de multiplication dans* $\mathbb{Z}/3\mathbb{Z}$ *et* $\mathbb{Z}/8\mathbb{Z}$.
2. *Donnez* $(\mathbb{Z}/3\mathbb{Z})^*$ *et* $(\mathbb{Z}/8\mathbb{Z})^*$.
3. *Résoudre :*
   — $3x - 4 \equiv 3[8]$
   — $2x - 1 \equiv 1[3]$
   — $6x \equiv 3[8]$
   — $2x + 1 \equiv 0[3]$

Exercice 3.3.2 (Règles de divisibilité expliquées) *Répondez aux questions suivantes :*

1. *Modulo* b*, qu'implique le fait que* b *divise* a *?*
2. *Soit* $a = \overline{a_m \dots a_1 a_0}^{10}$ *un entier en base* 10*. En se servant de la question précédente expliquer les règles de divisibilités par* 2, 5, 9, 11 *et* 101.
3. *Montrez que* 78775514327 *est divisible par* 101.

Exercice 3.3.3 *Soit* m *un entier naturel non nul. Le but de l'exercice est de démontrer que, quels que soient les entiers* a *et* b, $a \equiv b\,[m] \leftrightarrow m|(a-b)$.

1. *Sachant que* $a \equiv b\,[m]$*, écrire les divisions euclidiennes de* a *et* b *par* m.
2. *En déduire que* $a - b$ *est un multiple de* m.
3. *On a montré que* $a \equiv b\,[m] \rightarrow m|(a-b)$.
4. *Écrire la division euclidienne de* a *par* m*. On notera* r *le reste de cette division euclidienne.*
5. *Traduire la relation* $m|(a-b)$.
6. *En déduire que* a *et* b *ont le même reste dans la division euclidienne par* m.
7. *Conclure.*

Exercice 3.3.4 *Trouver les inverses de* 37 *modulo* 139*, de* 88 *modulo* 103*, de* 24 *modulo* 107.

Exercice 3.3.5 *L'entier* 583 *est-il inversible modulo* 679 *? Si oui, calculez son inverse.*

Exercice 3.3.6 (Exponentiation) *Trouvez le reste de la division euclidienne de* 2021^{2021} *par* 13.

Exercice 3.3.7 *Donnez-les* x *dans* $\mathbb{Z}/7\mathbb{Z}$ *tel que* $x^2 \equiv 3[7]$.

Exercice 3.3.8 (Tiré du Bac Métropole 2015) *On considère l'équation* (E) *à résoudre dans* $\mathbb{Z}$ *:*$7x - 5y = 1$.

a. Vérifier que le couple $(3,4)$ *est solution de* (E).

b. Montrer que le couple d'entiers (x,y) *est solution de* (E) *si et seulement si*

$$7(x-3) = 5(y-4).$$

c. Montrer que les solutions entières de l'équation (E) *sont exactement les couples* (x, y) *d'entiers relatifs tels que :*

$$x = 5k + 3 \text{ et } y = 7k + 4$$

avec $k \in \mathbb{Z}$.

2. Une boîte contient 25 *jetons, des rouges, des verts et des blancs. Sur les* 25 *jetons il y a* x *jetons rouges et* y *jetons verts. Sachant que* $7x - 5y = 1$*, quels peuvent être les nombres de jetons rouges, verts et blancs ?*

Exercice 3.3.9 (Les systèmes aussi) *Résoudre les système suivants :*

$$\begin{cases} 3x + y = 6 & [11] \\ 7x - 2y = 2 & [11] \end{cases} \tag{3.1}$$

$$\begin{cases} 4x + 2y = 1 & [13] \\ x - y = 11 & [13] \end{cases} \tag{3.2}$$

$$\begin{cases} 3x + y = 0 & [6] \\ x - 2y = 2 & [6] \end{cases} \tag{3.3}$$

Exercice 3.3.10 (Machine) *Implémentez et testez tous les tests de primalités présentés.*

Chapitre 4

Représentation numérique

Pour la suite, nous allons devoir être capable d'écrire des entiers sous différentes formes, différentes bases. Dans cette partie, nous allons voir comment nous pouvons passer d'une base à une autre et même comment faire des opérations dans ces bases. Les bases les plus utilisées de nos jours sont les bases 2, 10 et 16 mais on peut écrire un entier dans n'importe quelle base, même base 1986757 c'est possible bien que peu utile. Ainsi, la base 2 s'appelle binaire, la base 10 décimale et la base 16 hexadécimale.

4.1 Les bases ou comment représenter les entiers différemment

Définition 4.1.1 (Entier en base k) *Un entier en base k va s'écrire $\overline{a_i \dots a_0}^k$. De plus, on a l'égalité suivante*

$$\overline{a_i \dots a_0}^k = a_0 + a_1 \times k^1 + \cdots + a_i \times k^i$$

Ainsi, $\overline{10101}^2 = 1 + 4 + 16 = 21 = {}^{16}$.

Remarque 4.1.0.1 *En base* 16, $a = 10$, $b = 11$, $c = 12$, $d = 13$, $e = 14$, $f = 15$.

Proposition 4.1.1 (Existence d'une décomposition en base fixée (Admis)) *Soit b un entier au moins égal à 2. Tout entier naturel n peut s'écrire sous la forme d'une somme de termes de la forme $x_i^{b_i}$ où les nombres i sont des entiers naturels et les nombres x_i sont des entiers naturels compris entre 0 et $b-1$. Les entiers entre 0 et $b-1$ constituent les chiffres de la base b.*

Proposition 4.1.2 *Le nombre de chiffres dans l'écriture en base b d'un entier naturel n est égal à $\left\lfloor \frac{\ln(n)}{\ln(b)} \right\rfloor + 1$*

Bon c'est super, mais comment on fait pour passer d'un entier en base 10 à un entier en base k ? Et bien la réponse est simple, on fait encore et toujours des divisions euclidiennes. Cette méthode s'appelle la méthode des divisions en cascade.

Définition 4.1.2 (Méthode des divisions en cascade) *Soit $b > 2$ un entier et n positif.*

1. *On calcule le reste r_0 de la division entière de n par b et le quotient q_1.*
2. *On calcule le reste r_1 de la division entière de q_1 par b et le quotient q_2.*
3. *On calcule le reste r_2 de la division entière de q_2 par b et le quotient q_3.*
4. *...*
5. *Et ainsi de suite jusqu'à obtenir un quotient nul qj.*

Alors $n = r_j b^j + r_{j-1} b^{j-1} + \cdots + r_0$. En d'autres termes, les restes successifs sont les chiffres de l'écriture de n en base b, ce que l'on écrit sous la forme $n = \overline{r_j r_{j-1} \dots r_1 r_0}^b$.

Exercice 4.1.1 *Écrire 23456 en base 12 et $\overline{deadbeef}^{16}$ en base 20.*

4.1.1 La représentation des données dans un ordinateur

En informatique, on parle tout le temps de bit et d'octet, car c'est sous cette forme que votre ordinateur représente l'information.

Définition 4.1.3 (Le bit) *Le bit est une unité élémentaire en informatique qui comprend deux états : 0 ou 1. Cette unité sert d'information qui peut-être codée électroniquement. Le bit est donc la base de l'informatique. On utilise aussi ce même système à deux états en vrai ou faux dans les systèmes logiques.*

Définition 4.1.4 (L'octet) *L'octet est une unité de mesure en informatique qui mesure la capacité de stockage en mémoire ou sur un disque dur. Les tailles de fichier sont par exemple exprimées en octet. Un octet est égal à 8 bits, ce qui permet de coder 256 combinaisons différentes. Les couleurs sont notamment codées sur des octets. Le symbole de l'octet est o auquel on peut appliquer un multiplicateur, comme c'est le cas de toute unité de mesure (mètre, litres, etc) : comme le kilooctet (ko), le mégaoctet (Mo), le gigaoctet (Go) ou le téraoctet (To), voir figure 4.1.*

Remarque 4.1.1.1 *En anglais, octet se nomme byte, ce qui peut porter à confusion avec le bit. Pour résumer* 1 *octet* $=$ 1 *byte puisque c'est la même chose. Comme* 1 *octet* $=$ 8 *bits, on obtient :* 1 *bytes* $=$ 1 *octet* $=$ 8 *bits.*

Multiples de l'octet : préfixes SI et mésusages

Nom	Symbole	Valeur	Mésusage[3]
kilooctet	ko	10^3	2^{10}
mégaoctet	Mo	10^6	2^{20}
gigaoctet	Go	10^9	2^{30}
téraoctet	To	10^{12}	2^{40}
pétaoctet	Po	10^{15}	2^{50}
exaoctet	Eo	10^{18}	2^{60}
zettaoctet	Zo	10^{21}	2^{70}
yottaoctet	Yo	10^{24}	2^{80}

Multiples de l'octet : préfixes binaires

Nom	Symbole	Valeur
kibioctet	kio	2^{10}
mébioctet	Mio	2^{20}
gibioctet	Gio	2^{30}
tébioctet	Tio	2^{40}
pébioctet	Pio	2^{50}
exbioctet	Eio	2^{60}
zébioctet	Zio	2^{70}
yobioctet	Yio	2^{80}

FIGURE 4.1 – Table des multiples des octets.

Dans le monde, il y a 10 sortes de personnes : ceux qui comprennent le binaire, et les autres.

4.2 Exercice

Exercice 4.2.1 *Faire les conversions suivantes :*

— *Bits en Octet :*

1. 43114
2. 52437
3. 47397
4. 547
5. 90886
6. 70
7. 9712
8. 738

— *Bits en kilooctet :*

1. 4628
2. 74622
3. 281
4. 32473
5. 37
6. 1816
7. 833

— *Gigaoctet en kibioctet :*

1. 22451
2. 84730
3. 41081
4. 59710
5. 25318
6. 63638

— *Yobioctet en bits :*

1. 20
2. 97
3. 82
4. 44

Exercice 4.2.2 *Écrire les entiers sous la forme demandée :*

— *Binaire en décimal*

1. 11001011
2. 00110101
3. 10000011
4. 10001111
5. 11100011
6. 00000100
7. 00010010
8. 00111111
9. 10101010
10. 01010101
11. 11111110
12. 10010010

— *Binaire en hexadécimal*

1. 11001100
2. 11110001
3. 00110001
4. 11000010
5. 10100100
6. 10100111
7. 11101100
8. 11111100
9. 00111111
10. 00000011
11. 00001110
12. 10000001

— *Décimal en hexadécimal*

1. 888
2. 233
3. 21
4. 9
5. 75
6. 188
7. 56
8. 4
9. 121
10. 94
11. 201
12. 777

— *Décimal en Binaire*

1. 213
2. 9
3. 67
4. 99
5. 23
6. 143
7. 6
8. 1
9. 197
10. 252
11. 999
12. 739

— *Hexadécimal en Binaire*

1. $0x45$
2. $0xFA$
3. $0x5D$
4. $0x99$
5. $0x03$
6. $0x6B$
7. $0xDD$
8. $0xFE$
9. $0x22$
10. $0x18$
11. $0xDE$
12. $0xBE$

— *hexadécimal en ternaire*

1. $0x5A$
2. $0xCC$
3. $0x97$
4. $0x40$
5. $0x07$
6. $0x3D$
7. $0xF1$
8. $0xFB$
9. $0x82$
10. $0xE4$
11. $0xAD$
12. $0xEF$

Chapitre 5

Cryptographie symétrique

Le but de ce chapitre est de faire un tour rapide de la cryptographie symétrique et d'étudier la cryptographie historique. On va particulièrement s'intéresser aux fonctions de chiffrement qui se basent entièrement sur l'arithmétique modulaire.

5.1 Chiffrements historiques

En cryptographie antique, on chiffre les messages en faisant de l'arithmétique modulaire sur l'alphabet. Ainsi, on utilise une représentation de l'alphabet dans $\mathbb{Z}/26\mathbb{Z}$ que l'on peut représenter comme suit dans la figure 5.1

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

FIGURE 5.1 – Représentation de l'alphabet

5.1.1 Le chiffrement par décalage

Un des plus anciens chiffrements est celui dit de César. Il consiste à faire un décalage de 3 vers la droite des lettres de l'alphabet. Ce décalage se traduit par le fait

d'ajouter 3 à notre lettre modulo 26. Le déchiffrement quant à lui se fait en décalant les lettres vers la gauche de 3 ce qui se traduit par une soustraction modulo 26.

Définition 5.1.1 (Chiffrement de César) *Soit* $x \in \mathbb{Z}/26\mathbb{Z}$, $y \equiv x + 3[26]$.

Définition 5.1.2 (Déchiffrement de César) *Soit* $y \in \mathbb{Z}/26\mathbb{Z}$, $x \equiv y - 3[26]$.

Exemple 5.1.1.1 *"CESAR" se chiffre en "FHVDU". En effet,* $C = 2$ *et* $2 + 3 = 5 = F$, $\ldots$ *enfin* $R = 17$ *et* $17 + 3 = 20 = U$.

Exercice 5.1.1 *Chiffrez le message "OK", avec la même clé déchiffrez "FRRO".*

Le chiffrement de César est un cas particulier d'un chiffrement un peu plus complexe, le chiffrement par décalage. Celui-ci consiste à faire un décalage de k vers la droite ou la gauche des lettres de l'alphabet. Le déchiffrement consiste à faire le décalage inverse. Un tel k est appelé clé privée.

Définition 5.1.3 (Chiffrement par décalage) *Soient* $x \in \mathbb{Z}/26\mathbb{Z}$, $k \in \mathbb{Z}/26\mathbb{Z}$, $y \equiv x + k[26]$.

Définition 5.1.4 (Déchiffrement par décalage) *Soient* $y \in \mathbb{Z}/26\mathbb{Z}$, $k \in \mathbb{Z}/26\mathbb{Z}$, $x \equiv y - k[26]$.

Exemple 5.1.1.2 *Avec la clé* 7, *le mot "PAN" se chiffre en "WHU".*

Exercice 5.1.2 *On donne la clé* 24, *chiffrez le message "OK", avec la même clé déchiffrez "LM"*

La sécurité d'un système de chiffrement dépend du nombre de clés possibles. Pour ce système, il y a en tout 26 clés possibles. Il est rapide de toutes les essayer, même à la main.

5.1.2 Chiffrement affine

Pour augmenter le nombre de clés possibles, on peut chiffrer avec une fonction affine de la forme $ax+b[26]$ avec le couple (a,b) qui forme notre clé. Naïvement, on penserait que le nombre de clés est alors $26^2 = 676$ mais il est en réalité moindre. En effet, a doit être inversible modulo 26 pour rendre le déchiffrement possible sans ambiguïté. Ainsi, le nombre de clés possibles est $12 \times 26 = 312$. Cependant, c'est toujours plus que le chiffrement précédent.

Définition 5.1.5 (Chiffrement affine) *Soient* $x \in \mathbb{Z}/26\mathbb{Z}$ *et* $(a,b) \in (\mathbb{Z}/26\mathbb{Z})^* \times \mathbb{Z}/26\mathbb{Z}$, $y \equiv ax+b[26]$.

Définition 5.1.6 (Déchiffrement affine) *Soient* $x \in \mathbb{Z}/26\mathbb{Z}$ *et* $(a,b) \in (\mathbb{Z}/26\mathbb{Z})^* \times \mathbb{Z}/26\mathbb{Z}$, $x \equiv a^{-1}(y-b)[26]$.

Exemple 5.1.2.1 *Avec la clé* $5,2$, *le mot "OUI" se chiffre en "U". En effet :*
- $O = 14$ *donc* $5 \times 14 + 2 = 72 \equiv 20[26]$ *et* $20 = U$.
- $U = 20$ *donc* $5 \times 20 + 2 = 102 \equiv 24[26]$ *et* $24 = Y$.
- $I = 8$ *donc* $5 \times 8 + 2 = 42 \equiv 16[26]$ *et* $16 = Q$.

Exercice 5.1.3 *On donne la clé* $(7,4)$, *chiffrez le message "OK", avec la même clé déchiffrez "OK".*

5.1.3 Le chiffrement puissance

Toujours dans le but d'avoir plus de clés possibles, le chiffrement puissance à été introduit. Celui-ci consiste à chiffrer avec les fonctions de la forme $ax^c + b$ en ayant (a,b,c) comme clé. En faisant ça, on augmente la taille des clés à 1248.

Définition 5.1.7 (Chiffrement puissance) *Soient* $x \in \mathbb{Z}/26\mathbb{Z}$ *et* $(a,b,c) \in (\mathbb{Z}/26\mathbb{Z})^* \times \mathbb{Z}/26\mathbb{Z} \times (\mathbb{Z}/12\mathbb{Z})^*$, $y \equiv ax^c + b[26]$.

Définition 5.1.8 (Déchiffrement puissance) *Soient* $x \in \mathbb{Z}/26\mathbb{Z}$ *et* $(a,b,c) \in (\mathbb{Z}/26\mathbb{Z})^* \times \mathbb{Z}/26\mathbb{Z} \times (\mathbb{Z}/12\mathbb{Z})^*$, $x \equiv (a^{-1}(y-b))^{c^{-1}[12]}[26]$.

Exercice 5.1.4 *On donne la clé* $(7,4,5)$, *chiffrez le message "OK", avec la même clé déchiffrez "OK".*

5.1.4 Chiffrement par substitution

Une autre méthode pour chiffrer est d'utiliser une table de substitution. Celle-ci attribue à chaque lettre une autre comme le montre l'exemple de la figure 5.1.4.

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

FIGURE 5.2 – Exemple d'une table de substitution

On remarque ainsi que les chiffrements précédents sont des cas particuliers de celui-ci. Il y a maintenant $26!$ clés possibles soit environ 4×10^{26}. Cependant, la clé est grande, ici, c'est la table entière qui sert de clé.

5.1.5 Le chiffrement de Hill

Le chiffrement de Hill consiste à chiffrer le message en substituant les lettres du message, non plus lettre à lettre, mais par groupe de lettres. Il permet ainsi de rendre le cassage du code par observation des fréquences plus long et fastidieux, car ce n'est plus la fréquence des lettres qu'il faut observer, mais la fréquence de groupe de lettre. C'est un des tout premiers chiffrements par bloc.

Définition 5.1.9 (Chiffrement) *On numérote chaque lettre de 0 à 25. On écrit ensuite notre message m que l'on découpe en bloc de longueur n. Ainsi, $m = m_1||\dots||m_k$ avec chaque m_i de longueur n. La clé de chiffrement K est une matrice carrée de taille $p \times p$ telle que $PGCD(det(K), 26) = 1$ et le chiffré de chaque bloc m_i est $c_i = K \times m_i$ [26]. Donc le chiffré c est : $K \times m_1||\dots||K \times m_k$.*

Exemple 5.1.5.1 *On veut chiffrer "TEXTEACHIFFRER" avec $p = 2$ et $K = \begin{bmatrix} 3 & 5 \\ 6 & 17 \end{bmatrix}$.*

On commence par coder le message :

$$TEXTEACHIFFRER \leftarrow (19; 4; 23; 19; 4; 0; 2; 7; 8; 5; 5; 17; 4; 17)$$

On regroupe les lettres par paires car $p = 2$ créant ainsi 7 vecteurs de dimension deux, la dernière paire étant complétée aléatoirement :

— $X_1 = (19; 4)$ — $X_3 = (4; 0)$ — $X_5 = (8; 5)$ — $X_7 = (4; 17)$
— $X_2 = (23; 19)$ — $X_4 = (2; 7)$ — $X_6 = (5; 17)$

On multiplie ensuite ces vecteurs par la matrice K modulo 26 :

— $Y_1 = K \times (19; 4) = (25; 0)$
— $Y_2 = K \times (23; 19) = (8; 19)$
— $Y_3 = K \times (4; 0) = (12; 24)$
— $Y_4 = K \times (2; 7) = (15; 1)$
— $Y_5 = K \times (8; 5) = (23; 3)$
— $Y_6 = K \times (5; 17) = (22; 7)$
— $Y_7 = K \times (4; 17) = (19; 2)$

On obtient alors le chiffré : $ZAITMYPBXDWHTB$.

Définition 5.1.10 (Déchiffrement) *Soit $c = K \times m_1 || \dots || K \times m_k$ un chiffré et K la matrice de chiffrement. On inverse la matrice K et on note son inverse D. Puis on calcule $m = D \times c_1 || \dots || D \times c_k$.*

Exemple 5.1.5.2 *La matrice $K = \begin{bmatrix} 3 & 5 \\ 6 & 17 \end{bmatrix}$ est inversible modulo 26 car $det(K) = 3 \times 17 - 6 \times 5 = 21$ et $PGCD(21, 26) = 1$.*

Ainsi, avec la formule de l'inversion de matrice, $D = det(K)^{-1} \begin{bmatrix} 17 & -5 \\ -6 & 3 \end{bmatrix} = 5 \times \begin{bmatrix} 17 & -5 \\ -6 & 3 \end{bmatrix} = \begin{bmatrix} 7 & 1 \\ 22 & 15 \end{bmatrix}$.

Il n'y a plus qu'à déchiffrer les blocs :

— $X_1 = D \times (25; 0) = (19; 4)$
— $X_2 = D \times (8; 19) = (23; 19)$
— $X_3 = D \times (12; 24) = (4; 0)$
— $X_4 = D \times (15; 1) = (2; 7)$
— $X_5 = D \times (23; 3) = (8; 5)$
— $X_6 = D \times (22; 7) = (5; 17)$
— $X_7 = D \times (19; 2) = (4; 17)$

Et le message déchiffré est : $TEXTEACHIFFRER$.

On rappelle tout de même la formule de Laplace qui nous permet d'inverser des matrices.

Définition 5.1.11 (Formule de Laplace) *Le produit de A et la transposée de sa comatrice donne :*

$$A \times {}^t com(A) = {}^t com(A) \times A = det(A) \times Id$$

Donc si $det(A) \neq 0$, A est inversible et $A^{-1} = \dfrac{1}{det(A)} \times {}^t com(A)$.

On rappelle aussi ce qu'est une comatrice.

Définition 5.1.12 (Comatrice) *La comatrice d'une matrice A est la matrice qui a pour coordonnée les cofacteurs de la matrice A. De plus, les cofacteurs se calculent de la manière suivante :*

$$(com(A))_{i,j} = (-1)^{i+j} \times det(A_{i,j})$$

avec $A_{i,j}$ la sous-matrice carrée de taille $n-1$ déduite de A en supprimant la i-ème ligne et la j-ème colonne.

5.1.6 Chiffrement polyalphabétique

Le représentant le plus connu des chiffrements polyalphabétiques est celui de Vigenère. Décrit pour la première fois par Giovan Battista Bellaso en 1553, le principe de celui-ci est simple : on prend une clé qui est une suite de caractères, idéalement une phrase, et on l'ajoute à notre message modulo 26 pour en sortir un chiffré. C'est le premier chiffrement qui est dit cryptographiquement sûr si la taille de la clé est identique à la taille du message. Cela veut dire, simplement qu'il est impossible d'obtenir des informations sur le message ou la clé en ayant uniquement accès au chiffré. Bien sûr, cette propriété n'est valable que si la clé n'est pas réutilisée, auquel cas, de l'information commence à fuiter. La taille des clés est donc de 26^n avec n la taille du message.

Définition 5.1.13 (Chiffrement de Vigenère) *Soient $m \in (\mathbb{Z}/26\mathbb{Z})^n$ un message et $k \in (\mathbb{Z}/26\mathbb{Z})^n$ une clé alors, $c \equiv m+k[26]$. On additionne les vecteurs m et k dans $\mathbb{Z}/26\mathbb{Z}$.*

Définition 5.1.14 (Déchiffrement de Vigenère) *Soient $c \in (\mathbb{Z}/26\mathbb{Z})^n$ un message chiffré et $k \in (\mathbb{Z}/26\mathbb{Z})^n$ une clé alors, $m \equiv c-k[26]$. On soustrait les vecteurs c et k dans $\mathbb{Z}/26\mathbb{Z}$.*

Exercice 5.1.5 *On donne la clé "ETMONCODEESTPARFAIT", chiffrez le message "JESUISENPLEINEFORME", déchiffrez le message obtenu avec la clé "ETMONCRIPCDNJSICYQM".*

5.1.7 Enigma

Il existe quelques cas particuliers dans les chiffrements par substitution. Mais le cas le plus particulier est celui d'Enigma. C'est une machine de chiffrement utilisée par les Allemands lors de la seconde guerre mondiale. Elle est constituée dans sa

dernière version de 6 rotors, d'un brouilleur, d'un tableau de connexion et bien sûr, d'un clavier. On ne va pas détailler son fonctionnement, mais il faut garder en tête qu'il y a 10^{16} clés possibles qui correspondent à la position initiale des rotors et des brouilleurs. Chaque position donne une table de substitution différente et à chaque fois que l'on tape sur une lettre, la position des rotors et du brouilleur change. De ce fait, chaque lettre est chiffrée avec une table de substitution différente. Les figures 5.3 et 5.4 montrent des illustrations d'Enigma et de son fonctionnement.

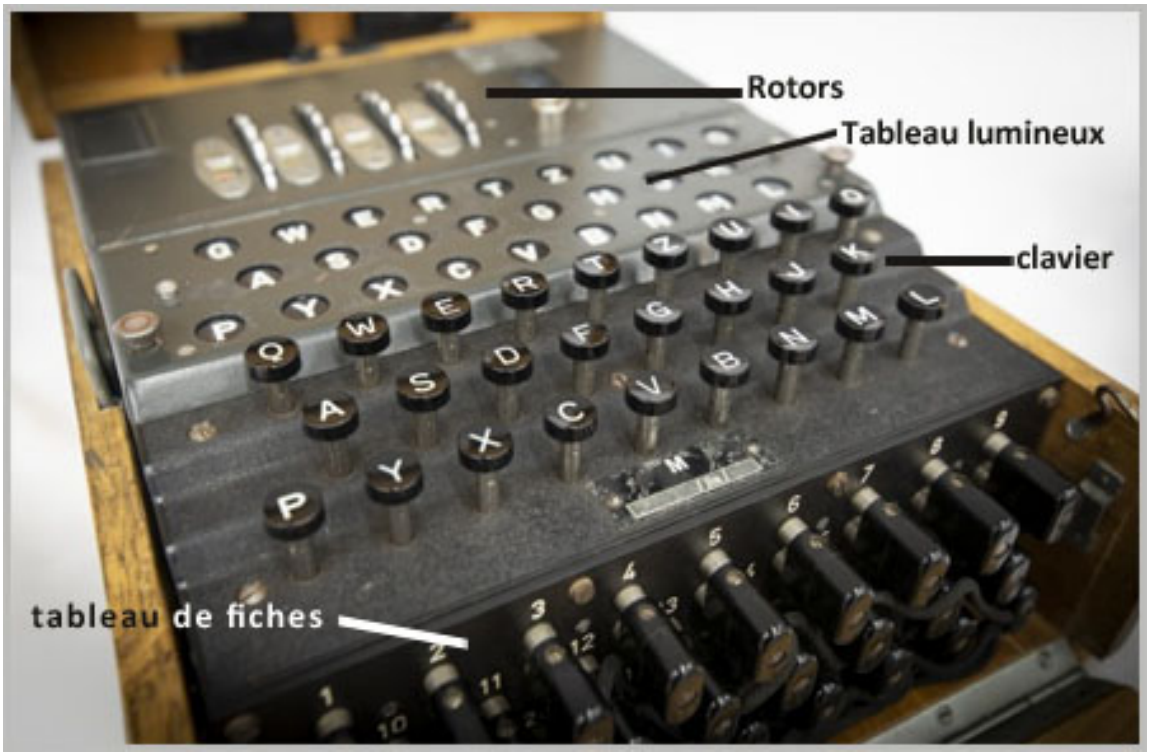

FIGURE 5.3 – La machine Énigma

5.2 Chiffrements modernes

À l'ère du numérique, les chiffrements ont, eux aussi, évolués. En effet, les ordinateurs accélérant grandement les calculs pour les recherches de clés, il a fallu s'adapter.

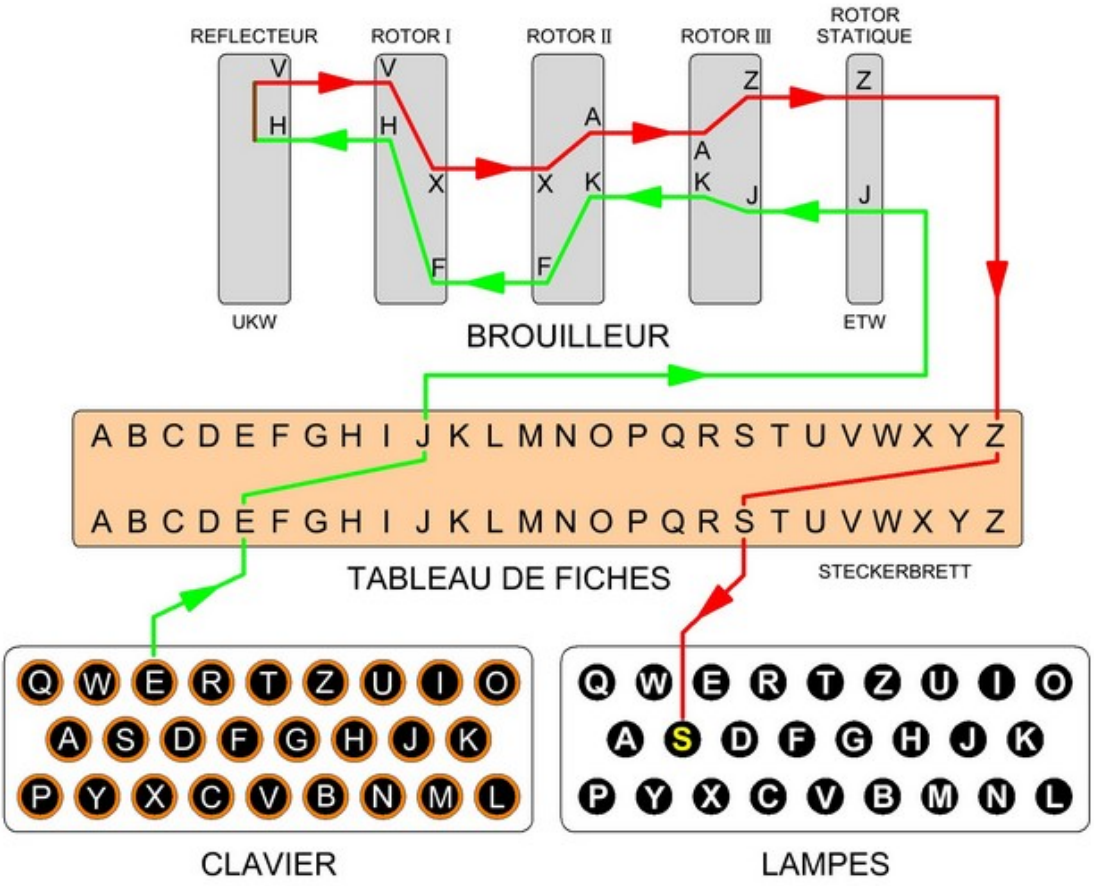

FIGURE 5.4 – Le fonctionnement d'Énigma

5.2.1 Le chiffrement de Vernam ou par masque jetable

C'est une variante du chiffrement de Vigenère, mais adaptée aux ordinateurs avec beaucoup plus de clés possibles. En effet, si on remarque que les lettres de l'alphabet peuvent se coder sur 5 bits, il est facile de comprendre comment il fonctionne. On écrit d'abord son message puis on le traduit en base 2. Notre clé sera alors une suite de bits aléatoire que l'on va venir ajouter à notre message initial. Pour déchiffrer, il suffira de retrancher la même suite de bits.

Définition 5.2.1 (Chiffrement de Vernam) *Soient $m \in (\mathbb{Z}/2\mathbb{Z})^n$ un message et $k \in (\mathbb{Z}/2\mathbb{Z})^n$ une clé alors, $c \equiv m + k[2]$. On additionne les vecteurs m et k dans $\mathbb{Z}/2\mathbb{Z}$.*

Définition 5.2.2 (Déchiffrement de Vernam) *Soient $c \in (\mathbb{Z}/2\mathbb{Z})^n$ un message chiffré et $k \in (\mathbb{Z}/2\mathbb{Z})^n$ une clé alors, $m \equiv c - k[26]$. On soustrait les vecteurs c et k dans $\mathbb{Z}/2\mathbb{Z}$.*

Exemple 5.2.1.1 *On prend $m =$ "MOI" $=> (12, 14, 8) => (01100, 01110, 01000) =>$ 011000111001000 et notre clé $k = 101010110100010$.*

Cet exemple est détaillé dans la figure 5.2.1.1

m	M	O	I
$\overline{m}^2$	01100	01110	01000
k	10101	01101	00010
$\overline{c}^2$	11001	00011	01010
c	Z	D	K

FIGURE 5.5 – Calcul d'un chiffré avec Vernam

Exercice 5.2.1 *On donne la clé "ENCODE", chiffrez le message "COUCOU", déchiffrez le message obtenu avec la clé "DECODE".*

5.2.2 Algorithme international simplifié de cryptage des données (IDEA)

L'algorithme international simplifié de cryptage des données (IDEA) [1] est un chiffrement par bloc simplifié pour les étudiants. IDEA est un chiffrement par bloc de clé symétrique qui possède les caractéristiques suivantes :

— Utilise un texte clair de longueur fixe de 16 bits et le chiffré en 4 morceaux de 4 bits chacun pour produire un texte chiffré 16 bits.
— La longueur de la clé est de 32 bits et est divisée en 8 blocs de 4 bits chacun.

Cet algorithme effectue une série de 4 tours complets identiques et 1 demi-tour. Chaque tour complet comprend une série de 14 étapes comprenant des opérations telles que :

— XOR bit à bit.
— Addition modulo 2^4.
— Multiplication modulo $2^4 + 1$.

Après 4 tours complets, le « demi-tour » final ne comprend que les 4 premières des 14 étapes précédemment effectuées dans les tours complets. La figure 5.6 montre à quoi ressemble un tour de chiffrement avec IDEA.

1. Source : Site de acervolima regardez IDEA

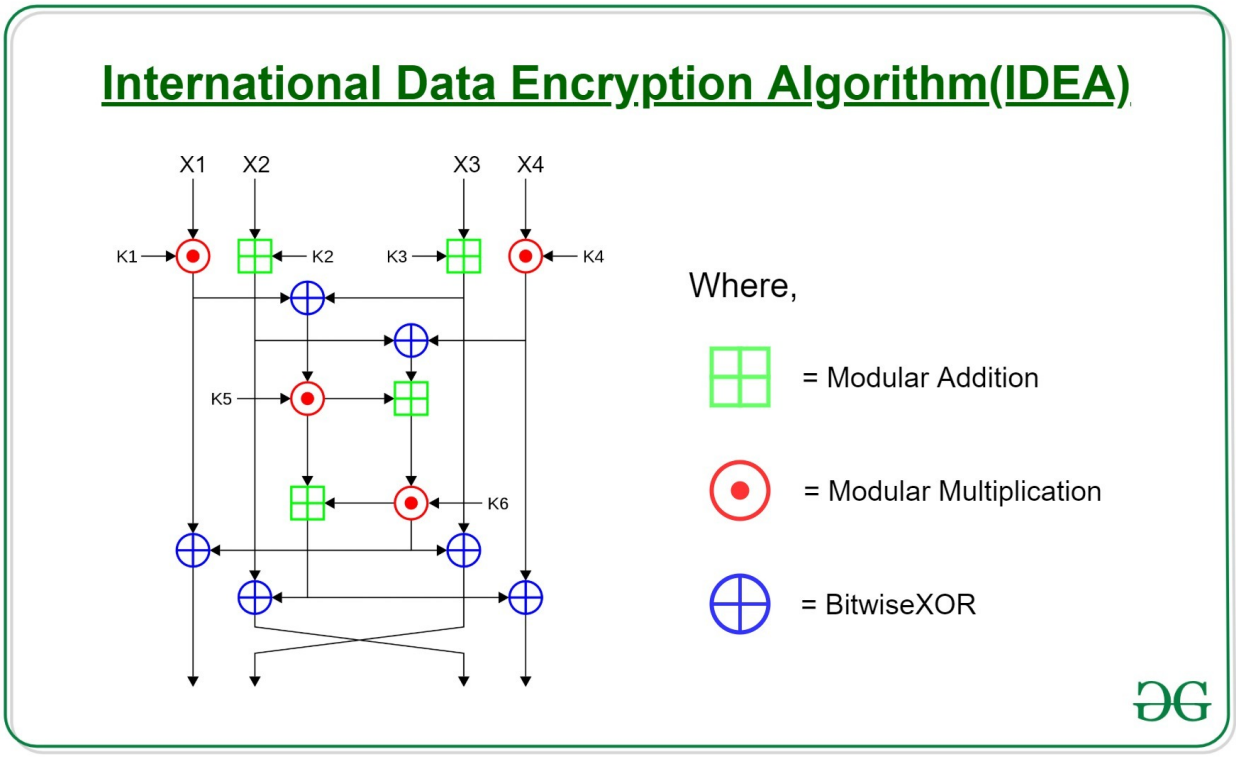

FIGURE 5.6 – Schéma d'une itération de l'IDEA

Génération des sous-clés : 4.5 tours nécessitent 28 sous-clés. La clé initiale, K de 32 bits, nous donne directement les 8 premières sous-clés par découpage. En tournant la clé K à gauche de 6 bits entre chaque groupe de 8, d'autres groupes de 8 sous-clés sont créés, ce qui implique moins d'une rotation par tour pour la clé (3 rotations).

Exemple 5.2.2.1 *Si on prend* $K = 11011100011011110011111$, *les* 28 *sous-clés seront les suivantes :*

— 1101 — 0011 — 0001 — 1101 — 1111 — 1001 — 1111
— 1100 — 1111 — 1011 — 0110 — 0011 — 1101 — 1101
— 0110 — 0101 — 1100 — 0111 — 1111 — 1100 — 0110
— 1111 — 1001 — 1111 — 0111 — 0101 — 0110 — 0111

Comme c'est un algorithme de chiffrement symétrique, la clé est la même pour chiffrer et déchiffrer. Le mieux c'est que l'algorithme de déchiffrement est iden-

tique à l'algorithme de chiffrement, il faut juste veiller à utiliser les sous-clés dans l'ordre inverse.

5.2.3 AES

Le standard actuel en terme de cryptographie symétrique. Il est donc important d'en parler même si on ne va pas rentrer dans les détails. Notez qu'il est fréquent de représenter des octets donc 8 bits en base 16.

Histoire

L'algorithme de Rijndael est un chiffrement symétrique par bloc multiple de 32 bits qui a été conçu par Joan Daemen et Vincent Rijmen, deux chercheurs belges en 1997. Dans un concours du NIST (National Institute of Standards and Technology) visant à instaurer un nouveau standard, car le `DES` (Data Encryption Standard) était devenu trop faible vis-à-vis des attaques de l'époque. Dans ce concours, la version 128 bits du chiffrement de Rijndael se placera dans les six premières places. Suite à cela, il s'impose comme le successeur du `DES` en 2000 sous le nom d'`AES` (Advanced Encryption Standard) devenant ainsi le deuxième véritable standard de la cryptographie.

Si le chiffrement de Rijndael est si important, c'est qu'il possède de nombreuses propriétés intéressantes :

— Résistance à toutes les attaques connues à l'époque.
— Il est possible d'implémenter l'`AES` aussi bien sous forme logicielle que matérielle (câblée).
— Rapidité du code sur la plus grande variété de plateformes (logicielles et matérielles) possible.
— Simplicité dans la conception.
— Besoins en ressources et mémoire très faibles.
— Il n'est pas basé sur un schéma de Feistel.

Principe

L'`AES` est un chiffrement par bloc composé de quatre fonctions majeures :

— `SubBytes` (S-box)
— `ShiftRows`

— `MixColumns`
— `AddRoundKey`

Ces fonctions sont imbriquées de façon à créer un schéma de chiffrement itérable. Dans le standard, le nombre d'itérations est 10,12 ou 14 selon la taille de la clé qui est de 128, 192 ou 256 bits.

Un algorithme de cadencement de clé calcule à partir de la clé K une suite de sous-clés de tour comportant toutes 16 octets qui seront utilisées pour le `AddRoundKey` .

Le texte à chiffrer est découpé en blocs de 16 octets et, pour chaque bloc clair, on effectue les opérations suivantes pour produire un bloc chiffré de même longueur :

— On initialise un tableau 4×4 avec 16 octets de texte clair.
— On applique successivement sur ce tableau les opérations suivantes :
  1. `AddRoundKey`
  2. On effectue "nombre de tours total"-1 tours comportant les 4 étapes : `SubBytes` , `ShiftRows` , `MixColumns` , `AddRoundKey` dans cet ordre.
  3. Un dernier tour ne comporte plus que 3 étapes : `SubBytes` , `ShiftRows` , `AddRoundKey` .
— Le contenu actuel du tableau donne les 16 octets du texte chiffré.

La figure 5.7 montre le schéma d'une itération de l'`AES`.

5.3 Exercices

Les exercices nommés sont des compléments aux cours, gardez donc une trace de ces exercices, car cela vous sera utile dans la suite.

Exercice 5.3.1 *Dire si les fonctions de chiffrement ci-dessous sont utilisables, justifiez.*

1. $y \equiv 3x + 7\ [26]$
2. $y \equiv 18x + 9\ [26]$
3. $y \equiv 23x + 12\ [26]$
4. $y \equiv 18x + 24\ [26]$
5. $y \equiv 26x + 26\ [26]$
6. $y \equiv 26x + 9\ [26]$
7. $y \equiv 23x + 15\ [26]$
8. $y \equiv 10x + 5\ [26]$
9. $y \equiv 22x + 23\ [26]$
10. $y \equiv 9x + 20\ [26]$
11. $y \equiv 9x + 24\ [26]$
12. $y \equiv 7x + 25\ [26]$

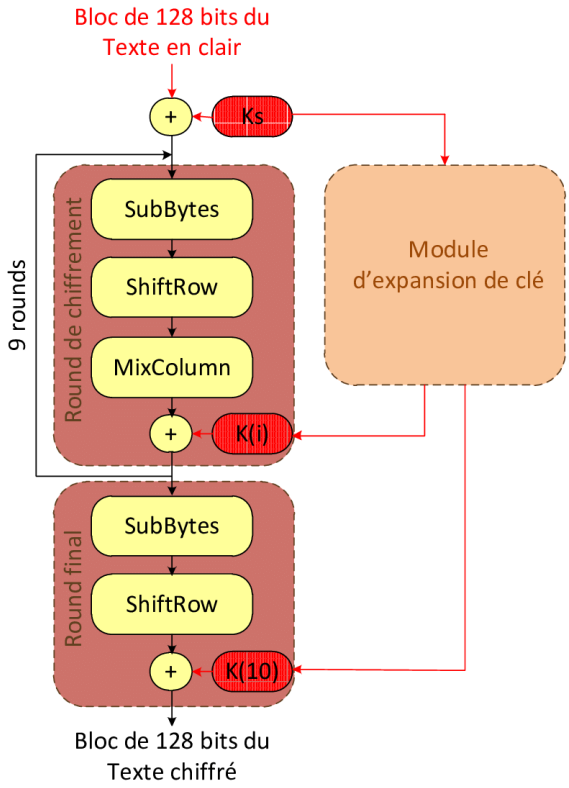

FIGURE 5.7 – Schéma d'une itération de l'AES

13. $y \equiv 14x + 11\ [26]$
14. $y \equiv 15x + 24\ [26]$
15. $y \equiv 8x + 9\ [26]$
16. $y \equiv 23x + 20\ [26]$
17. $y \equiv 6x + 2\ [26]$
18. $y \equiv 26x + 13\ [26]$
19. $y \equiv 2x + 17\ [26]$
20. $y \equiv 23x + 10\ [26]$
21. $y \equiv 21x + 24\ [26]$

Exercice 5.3.2 *Écrire les fonctions de déchiffrement des fonctions suivantes :*

1. $y \equiv 3x + 8\ [26]$
2. $y \equiv 24x + 25\ [26]$
3. $y \equiv 15x + 6\ [26]$
4. $y \equiv 14x + 19\ [26]$
5. $y \equiv 20x + 16\ [26]$
6. $y \equiv 1x + 9\ [26]$
7. $y \equiv 24x + 2\ [26]$
8. $y \equiv 11x + 16\ [26]$
9. $y \equiv 22x + 25\ [26]$
10. $y \equiv 3x + 13\ [26]$
11. $y \equiv 22x + 6\ [26]$
12. $y \equiv 23x + 8\ [26]$
13. $y \equiv 16x + 25\ [26]$
14. $y \equiv 16x + 11\ [26]$
15. $y \equiv 17x + 9\ [26]$
16. $y \equiv 18x + 21\ [26]$
17. $y \equiv 19x + 11\ [26]$
18. $y \equiv 26x + 17\ [26]$
19. $y \equiv 4x + 18\ [26]$
20. $y \equiv 7x + 18\ [26]$
21. $y \equiv 1x + 26\ [26]$

Exercice 5.3.3 *En utilisant le schéma de chiffrement de IDEA, chiffrez le message* $m_1 = 1001110010101100$ *et* $m_2 = 0000000000000000$ *avec la clé* $K = 11011100011011110011111101011001$.

Exercice 5.3.4 *Chiffrez le message "LACRYPTOCRIGOLO" avec les méthodes suivantes :*

1. $y \equiv 12x + 17\ [26]$
2. $y \equiv 25x + 15\ [26]$
3. $y \equiv 5x + 4\ [26]$
4. *Vigenère avec les clés :* $RUSOOEHJTQVLTRZ$
5. $FMMZOYUOKJSUYOE$
6. $OXTADOQTUXTMMCV$
7. *Vernam avec les clés :*
8. 00100111100110110100111001111001000100010111 1
9. 00111001001001001100010100100100000000011011
10. 00111010100001111110101110101110100000000011

Exercice 5.3.5 *Sherlock a transmis un message à Watson en morse, mais une partie de celui-ci est chiffré, aidez-le à résoudre l'énigme et sauver le monde. Le message :*

Cher Watson,

Je suis en bien mauvaise compagnie ici.
Je ne puis donc point intervenir moi-même pour contrecarrer le plan du professeur.
Celui-ci est malin et il a chiffré le code de désarmement des missiles en plusieurs parties.

Le premier morceau de code est "XSTWDV". Je l'ai trouvé près d'un livre de mathématiques datant de 1553 *et d'une photo de lui avec son nom au dos : Moriarty*

J'espère que cela va vous aider Watson.

Le deuxième morceau était sur une feuille avec écrit "10010001000001010001001001011" *et une étrange citation "Pour cacher quelque chose, il suffit bien souvent de ne pas chercher à le cacher, la clé se trouve dans la nullité.".*

J'espère que cette énigme ne vous arrêtera pas mon cher Watson. Après tout, je vous ai bien formé.

Bien, parlons du dernier message, il est écrit "EFJKDYLLZ" à côté d'un magazine dont le titre était "Lorsque l'on maigrit, on s'...", je n'ai pas eu le temps de lire le reste, c'est là que je me suis fait capturer.

Je compte sur vous Watson, le monde a besoin de vous.

Amicalement,

Holmes

PS : Un détail me revient, les deux dernières lettres du code secret sont "OM".

Exercice 5.3.6 *Le message suivant a été chiffré avec un chiffrement monoalphabétique, déchiffrez-le en utilisant la méthode de la récurrence des lettres :*

OTJAXJJFKFOFMTJQTNXQHZTJXUMMGTJUXRKGTPJTJRHFJOTQTNCFWWG
TRTUMQPUMTYMTNCFWWGTAHGPUTRTMCXQTQTJPKJMFMPMFXURXUX-

HOAC HKTMFDPTUTJMAXPGMHUMAHJFRAXJJFKOTHNXUQFMFXUDPTOTM-TYMTJX FMHJJTEOXUZOTJJHSHUMJHGHKTJJXUMOTJFUSTUMTPGJQTOHNG-BAMHU HOBJTNTJMPUTRTMCXQTATGRTMMHUMQTQTNGBAMTGOTJRTJ-JHZTJNXQTJ

La fréquence d'apparition des lettres en français est dans la figure 5.3.6.

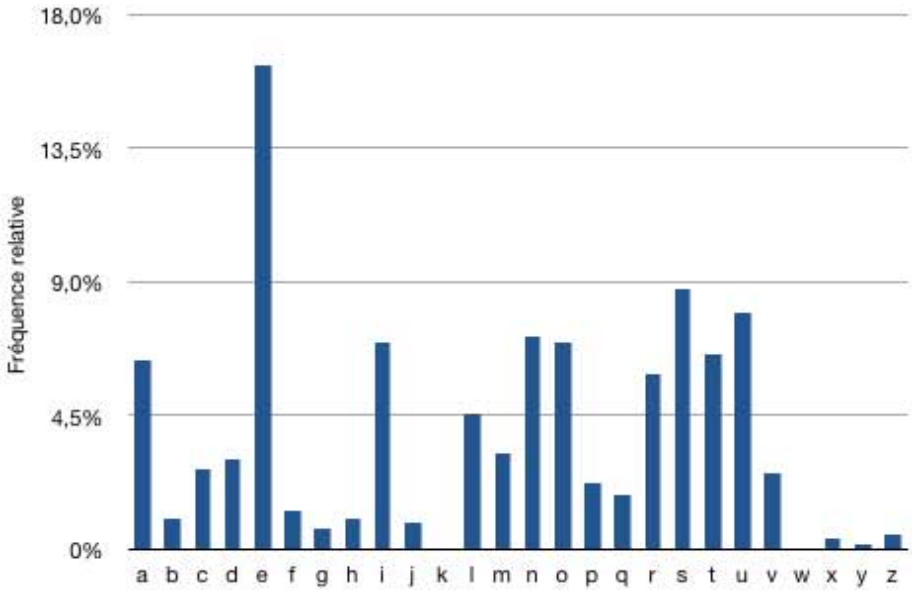

FIGURE 5.8 – Fréquence d'apparition des lettres en français

Exercice 5.3.7 *On a chiffré un message à l'aide du chiffrement de Vigenère. Ci-dessous figure le chiffré. Retrouver le clair.*

DQPQFLHHFUFIOFDMTFHWGNHYWGVMNRDWTKEPFUMIOGPITGQX JUSP-VUJYJFHTBTOITJDPFWUWEGVTFCXBSQXKFUFVJCUHTNHWBX DMFPWTSKVTP-WUGJDOITNHWBADRUEOSVGVRVUDYYRRXFCXBEG FSVNHYSU

Exercice 5.3.8 (Sur ordinateur) *Implémenter les fonctions de chiffrement et déchiffrement suivantes :*

1. *Affine.*
2. *Vernam.*
3. *Vigenère.*
4. *IDEA.*
5. *Hill pour des matrices de taille 2 et 3.*

Chapitre 6

Cryptographie asymétrique

Dans ce chapitre, nous allons parler de cryptographie asymétrique ou cryptographie à clef publique. Elle est très différente de ce que nous avons vu précédemment. En effet, celle-ci se base sur une paire de clés (p, k) avec p une clé publique et k une clé secrète. La clé publique sert au chiffrement et la clé secrète (ou privée) sert au déchiffrement. La représentante internationale des fonctions asymétriques est RSA. Le concept est assez récent et est apparu pour la première fois dans les années 1976. Tous les systèmes de chiffrements asymétriques doivent reposer sur des problèmes difficiles pour ne pas être cassés. Cette caste de problèmes particuliers est problème $NP - dur$ ou $NP - Hard$.

6.1 Les fonctions les plus connues

Faisons un petit tour des fonctions les connues que sont RSA, Diffie-Hellman et Merkle-Hellman.

6.1.1 Merkle-Hellman

C'est un des premiers systèmes de chiffrement asymétrique reposant sur le problème du sac à dos. Il a été défini par Ralph Merkle et Martin Hellman en 1978 et cassé en 1982 par Adi Shamir.

Forme des clés

Pour générer la paire de clés nécessaire, on a besoin d'une suite supercroissante.

Définition 6.1.1 (Suite supercroissante) *Une suite supercroissante est une suite croissante de n nombres telle que chaque terme est plus grand que la somme de tous les termes précédents.*

On génère donc les clés de la façon suivante :

Définition 6.1.2 (Génération de clés pour Merkle-Hellman) *Pour générer une clé de chiffrement pour Merkle-Hellman, on effectue les étapes suivantes :*

1. *Soit $S = \{s_1, \ldots, s_n\}$ une suite supercroissante et $N \in \mathbb{N}$ un entier tel que :*

$$N > \sum_{i=1}^{n} s_i$$

2. *On prend $P < N$ un entier tel que $PGCD(P, N) = 1$.*
3. *On calcule $B = \{b_1, \ldots, b_n\}$ avec $b_i \equiv P \times a_i \ [N]$ pour tout i de 1 à n.*

La clé publique est B et la clé privée est (N, P, S).

Exemple 6.1.1.1 *On prend $S = \{5, 7, 27, 81, 243, 729, 2187\}$ et $N = 6569$. Alors, $P = 17$ convient pour la suite car $PGCD(17, 6569) = 1$. Ensuite, on calcule :*

$$B = \{85, 119, 459, 1377, 4131, 5824, 4334\}$$

. Enfin, la clé publique est :

$$\{85, 119, 459, 1377, 4131, 5824, 4334\}$$

et la clé privée est :

$$(6569, 17, \{5, 7, 27, 81, 243, 729, 2187\})$$

Maintenant que nous disposons de nos clés, intéressons-nous au chiffrement et déchiffrement.

Chiffrement et déchiffrement

Définition 6.1.3 (Chiffrement de Merkle-Hellman) *On écrit le message m que l'on veut chiffrer en binaire. Le message que l'on peut chiffrer en binaire ne doit pas être plus long que n la taille de la clé publique B. Ainsi, le chiffré est :*

$$c = \sum_{i=1}^{n} \overline{m}_i^2 \times b_i$$

Exemple 6.1.1.2 *En prenant la clé publique* $\{85, 119, 459, 1377, 4131, 5824, 4334\}$, 1011010 *se chiffre en* $85 + 459 + 1377 + 5824 = 7745$.

Maintenant, il ne reste plus qu'à déchiffrer et cela se fait de la façon suivante :

Définition 6.1.4 (Déchiffrement de Merkle-Hellman) *Soit* c *le message chiffré et* (N, P, S) *la clé privée. On pose* $p \equiv P^{-1} \times c\ [N]$. *Ensuite, on résout le problème du sac à dos suivant :*

$$c = \sum_{i=1}^{n} \overline{m}_i^2 s_i$$

Cette instance du sac à dos est facile, car S *est ce qu'on appelle un sac mou à cause de sa structure supercroissante.*

Alors c'est très bien, mais comment on résout le problème du sac à dos ? Eh bien, on ne peut pas le faire en temps raisonnable sauf si le sac à une structure particulière comme ici. Mais déjà, définissons le problème du sac à dos.

Définition 6.1.5 (Problème du sac à dos) *Soit* $S = \{o_1, \dots o_n\}$ *un sac avec des objets de poids respectifs* o_i *et* p *un poids maximum. La question est la suivante, quels sont les objets que je peux mettre dans mon sac pour que celui-ci pèse exactement* k *?*

On est content, mais ça ne nous dit pas comment on peut résoudre le problème du sac à dos quand on a un sac mou ? La réponse est simple, on va utiliser un algorithme de type glouton comme décrit ci-dessous.

Définition 6.1.6 (Algorithme glouton) *Soit un sac* $S = \{o_1, \dots o_n\}$ *et* k *le poids visé. On va décomposer* k *en soustrayant le plus grand* o_i *possible à chaque fois jusqu'à obtenir* 0. *Les* o_i *ainsi utilisés nous donnent la solution au problème.*

Exemple 6.1.1.3 *On prend la clé privée* $(6569, 17, \{5, 7, 27, 81, 243, 729, 2187\})$ *et le message chiffré* 7745. *On calcule* $p \equiv 17^{-1} \times 7745 \equiv 4637 \times 7745 \equiv 842\ [N]$. *Puis on utilise notre algorithme glouton :*

$$\begin{aligned} 842 - \underline{729} &= 113 \\ 113 - \underline{81} &= 32 \\ 32 - \underline{27} &= 5 \\ 5 - \underline{5} &= 0 \end{aligned}$$

On regarde notre sac en notant ce que l'on a utilisé : $\{\underline{5}, 7, \underline{27}, \underline{81}, 243, \underline{729}, 2187\}$ *donc le message est :* 1011010

Il est important de noter que les algorithmes de chiffrement asymétrique fondés sur le sac à dos ont tous été cassés à ce jour.

6.1.2 RSA

La superstar des fonctions de chiffrement asymétrique n'est autre que RSA qui se base sur le problème de factorisation des entiers. Cet algorithme a été décrit en 1977 par Ronald Rivest, Adi Shamir et Leonard Adleman. Il est très utilisé de nos jours par exemple pour sécuriser des transactions pour le commerce en ligne. Il existe de nombreuses variantes de RSA, mais nous allons parler de la version de base.

Forme des clés

Pour générer des clés RSA, on a besoin de l'indicatrice d'Euler.

Définition 6.1.7 (Indicatrice d'Euler) *C'est une fonction φ qui à tout entier $n \in \mathbb{N}^*$ lui associe le nombre d'entiers compris entre 1 et n et premiers avec n.*

Exemple 6.1.2.1 *Par exemple si on prend 12, les nombres premiers avec 12 sont : 1, 5, 7 et 11. Ainsi, $\varphi(12) = 4$*

Il existe une formule pour la calculer :

Proposition 6.1.1 ($\varphi(n)$) *Soit n un entier alors, on peut écrire sa décomposition en facteur premier $n = \prod_{i=1}^{r} p_i^{k_i}$ avec p_i des entiers premiers et k_i des entiers. Alors :*

$$\varphi(n) = \prod_{i=1}^{r} (p_i - 1) \times p_i^{k_i - 1}$$

Exemple 6.1.2.2 *Par exemple si on prend $n = 12$ alors $n = 2^2 \times 3$. Ainsi :*

$$\varphi(12) = ((2-1) \times 2^{2-1}) \times ((3-1) \times 3^{1-1} = 2 \times 2 = 4$$

De plus, on peut généraliser le petit théorème de Fermat avec le théorème d'Euler.

Théorème 6.1.1 (Théorème d'Euler) *Pour tout entier $n \geq 2$ et tout entier a premier avec n (autrement dit : inversible mod n),*

$$a^{\varphi(n)} \equiv 1 \ [n]$$

On peut maintenant générer nos clés.

Définition 6.1.8 (Génération de clés RSA) *Pour générer une clé RSA, on a les étapes suivantes :*

1. *On tire deux entiers premiers p et q distincts.*
2. *On calcule $n = p \times q$ et $\varphi(n) = (p-1) \times (q-1)$.*
3. *On tire e un entier aléatoire premier avec $\varphi(n)$.*
4. *on calcule $d \equiv e^{-1}\ [\varphi(n)]$.*

Ainsi, la clé publique est (n, e) et la clé privée est (n, p, q, d).

Exemple 6.1.2.3 *On génère une clé RSA :*

1. *On tire deux entiers $p = 11$ et $q = 7$ premiers distincts.*
2. *On calcule $n = p \times q = 77$ et $\varphi(n) = (p-1) \times (q-1) = 60$.*
3. *On tire $e = 13$ un entier aléatoire premier avec $\varphi(n)$.*
4. *on calcule $d \equiv e^{-1} \equiv 37\ [\varphi(n)]$.*

Ainsi, la clé publique est $(77, 13)$ et la clé privée est $(77, 7, 11, 37)$.

Maintenant que nous disposons de nos clés, intéressons-nous au chiffrement et déchiffrement.

Chiffrement et déchiffrement

On chiffre un message de la manière suivante avec RSA :

Définition 6.1.9 (Chiffrement RSA) *Soit (n, e) la clé publique pour RSA et m un entier plus petit que n. Alors le chiffré de m est $c \equiv m^e\ [n]$.*

Exemple 6.1.2.4 *Si on prend la clé publique $(n = 77, e = 13)$ et que le message est $m = 42$ alors, $c \equiv 42^{13} \equiv 14\ [77]$.*

On déchiffre un message de la manière suivante avec RSA :

Définition 6.1.10 (Déchiffrement RSA) *Soit (n, p, q, d) la clé privée et c un message chiffré alors le message en clair est : $m \equiv c^d\ [n]$*

Exemple 6.1.2.5 *Si on prend la clé privée $(77, 7, 11, 37)$ et le message chiffré $c = 14$ alors, $m \equiv 14^{37} \equiv 42\ [77]$.*

Le meilleur moyen de casser RSA est de résoudre le problème de factorisation des entiers. C'est un problème $NP-Hard$ et le meilleur algorithme de factorisation nommé NFS (Crible algébrique) a besoin de 268 jours pour factoriser un entier de taille 155 bits alors que de nos jours les entiers RSA sont de taille au moins 256 bits.

6.2 Exercices

Les exercices nommés sont des compléments aux cours, gardez donc une trace de ces exercices, car cela vous sera utile dans la suite.

Exercice 6.2.1 *Dites si les suites suivantes sont supercroissantes, justifiez :*

1. $[8, 14, 47, 141, 423, 1269, 3807, 11421, 34263]$
2. $[73, 144, 356, 359, 452, 530, 537, 567, 3569]$
3. $[4, 15, 41, 123, 369, 1107, 3321, 9963, 29889]$
4. $[8, 12, 43, 129, 387, 1161, 3483, 10449, 31347]$
5. $[2, 12, 31, 93, 279, 837, 2511, 7533, 22599]$
6. $[2, 15, 37, 111, 333, 999, 2997, 8991, 26973]$
7. $[84, 247, 287, 364, 379, 589, 681, 979, 5700]$
8. $[1, 12, 29, 87, 261, 783, 2349, 7047, 21141]$
9. $[5, 14, 41, 123, 369, 1107, 3321, 9963, 29889]$
10. $[141, 219, 328, 339, 484, 499, 721, 828, 1674]$
11. $[6, 12, 39, 117, 351, 1053, 3159, 9477, 28431]$

Exercice 6.2.2

1. *Expliquer comment construire une suite supercroissante et en donner une.*
2. *Écrire en pseudo code une fonction qui prend en entrée deux entiers et qui sort une suite supercroissante commençant par ces deux entiers.*

Exercice 6.2.3 *Soit S la suite suivante :* $[3, 19, 47, 141, 423, 1269, 3807, 11421, 34263, 102789]$.

1. *Montrez que S est supercroissante.*
2. *Montrez que $N = 308375$ est plus grand que la somme de tous les éléments de S.*
3. *Montrez que $PGCD(103, N) = 1$*
4. *En déduire une clé privée pour le chiffrement de Merkle-Hellman.*
5. *En utilisant la clé privée, calculez la clé publique.*
6. *Chiffrez les messages "CR", "YP" et "TO".*
7. *Déchiffrez les messages "474214" et "531997"*

Exercice 6.2.4 (Sur ordinateur) *Implémenter le chiffrement de Merkle-Hellman.*

Exercice 6.2.5 *Dire si les nombres suivants sont premiers :*

1. 11
2. 26
3. 29
4. 83
5. 89
6. 254
7. 257
8. 767
9. 769
10. 2306
11. 2309
12. 6923

Exercice 6.2.6 (Nombre premiers jumeaux et cousins) *On dit de deux nombres premiers qu'ils sont jumeaux si et seulement si leur différence est égale à* 2.
On dit de deux nombres premiers qu'ils sont cousins si et seulement si leur différence est égale à 4.

1. *Trouvez les nombres premiers jumeaux entre* 2 *et* 105.
2. *Trouvez les nombres premiers cousins entre* 2 *et* 105.
3. *Lesquels sont les plus fréquents ?*

Exercice 6.2.7

1. *Construisez une paire de clés RSA avec* $p = 101$ *et* $q = 103$ *en ayant le* e *le plus petit possible.*
2. *Chiffrez les messages "HE", "LP" et "ME".*
3. *Déchiffrez les messages* 6366 *et "*9158*".*

Exercice 6.2.8 (Sur ordinateur) *Implémenter le chiffrement RSA.*

Chapitre 7

Signatures et fonction de hachage

Dans ce chapitre, nous traiterons du principe de signature au travers de l'exemple de la signature RSA et de fonction de hachage. La signature, comme son nom l'indique, permet d'assurer que la personne qui a envoyé le message est bien celle qu'elle prétend être. Une fonction de hachage permet de hacher un message sans qu'on puisse revenir en arrière. Cela sert notamment à s'assurer de l'intégrité d'un message.

7.1 Signature RSA

Comme dit plus haut, la signature permet à la fois de s'assurer de l'intégrité d'un message et à la fois de s'assurer que c'est la bonne personne qui l'a signé.

Alors, voyons comment Alice peut envoyer un message signé à Bob :

Définition 7.1.1 (Signature RSA) *Soient (e_A, d_A, n_A) la clé RSA d'Alice, (e_B, d_B, n_B) la clé RSA de Bob et m_A le message que veut envoyer Alice. Elle va procéder de la manière suivante :*
Elle calcule $c_1 \equiv m_A^{e_B}\ [n_B]$, $s_A \equiv m_A^{d_A}\ [n_A]$ et $c_2 \equiv s_A^{e_B}\ [n_B]$. Puis envoie le couple (c_1, c_2) à Bob.

Une fois le message parvenu à Bob, il va vérifier de la façon suivante :

Définition 7.1.2 (Vérification signature RSA) *Soient (e_A, d_A, n_A) la clé RSA d'Alice, (e_B, d_B, n_B) la clé RSA de Bob et c_A, s_A le message qu'a envoyé Alice. Bob va procéder de la manière suivante :*
Il calcule $m_A \equiv c_1^{d_B}\ [n_B]$ pour lire le message et $c_2^{d_B} \equiv s_A\ [n_B]$ puis il vérifie que $s_A^{e_A} \equiv m_A\ [n_A]$.

On peut vérifier l'intégrité et l'origine du message, car si c_1 ou c_1 a été modifié, les deux messages à la fin du processus de modification ne seront pas égaux. De ce fait, on saura que l'un des deux a été modifié ou qu'Alice n'est pas l'envoyeur.

7.2 Fonction de hachage

S'il ne s'agit que de vérifier l'intégrité d'un message ou de vérifier un mot de passe, on ne va pas utiliser d'algorithme de signature qui sont trop coûteux. On va utiliser ce que l'on appelle une fonction de hachage. Une fonction de hachage va prendre en entrée un objet de la taille que l'on veut et va en sortir une empreinte de taille fixée à l'avance. Ce sont donc des fonctions surjectives. De ce fait, si on prend h une fonction de hachage, il est possible de trouver x et x' avec $x \neq x'$ tel que $h(x) = h(x)$. On appelle cela une collision et la solidité des fonctions de hachage dépendent de la difficulté à trouver des collisions. Il existe un cas particulier de fonctions de hachage sans collisions, ce sont les fonctions de hachage parfaites.

Définition 7.2.1 (Fonction de hachage parfaite) *Soit h une fonction de hachage de A dans B, h est dite parfaite pour A si elle est injective.*

La figure 7.1 nous donne un exemple de haché avec md5.

Cependant, ce qui nous intéresse, ce sont les fonctions de hachage cryptographiques

7.2.1 Les propriétés d'une fonction de hachage cryptographique.

Les fonctions de hachage cryptographiques ont les propriétés suivantes :

Propriété 7.2.1 (Résistance à la préimage) *Soit h une fonction de hachage cryptographique. Pour toute valeur de haché y, il est difficile de trouver x tel que $h(x) = y$.*

Propriété 7.2.2 (Résistance à la seconde préimage) *Soit h une fonction de hachage cryptographique. Pour tout x, il est difficile de trouver $y \neq x$ tel que $h(x) = h(y)$.*

Propriété 7.2.3 (Résistance aux collisions) *Soit h une fonction de hachage cryptographique. Il est difficile de trouver $y \neq x$ tel que $h(x) = h(y)$.*

Remarque 7.2.1.1 *Un algorithme qui permet de trouver une préimage peut aussi trouver une collision. En pratique, on veut que les fonctions de hachage cryptographiques soient résistantes aux collisions.*

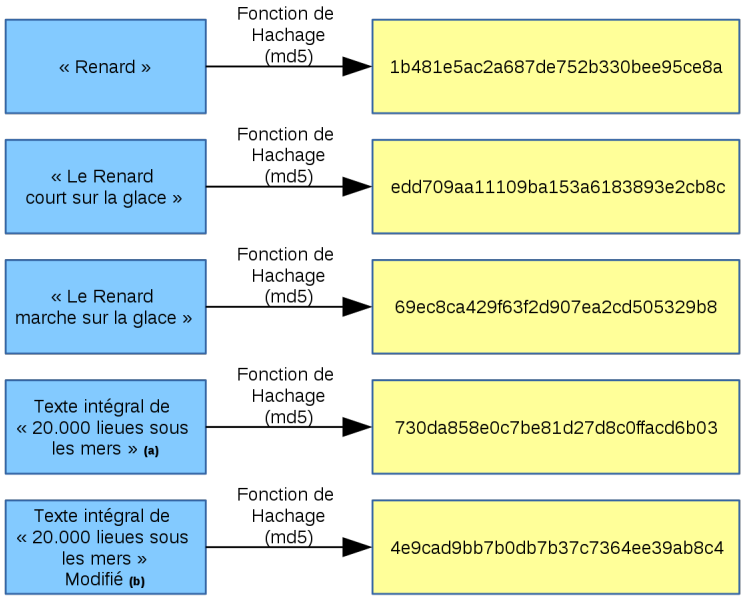

FIGURE 7.1 – Exemple de haché avec md5.

Parfois, on demande une propriété étonnante :

Propriété 7.2.4 (Indistingabilité avec un oracle aléatoire) *Soit h une fonction de hachage cryptographique et O un oracle aléatoire. Alors h et O sont indistinguables*

Autrement dit, la sortie de la fonction de hachage doit ressembler à de l'aléatoire, on ne doit pas pouvoir prédire le résultat d'un hachage.

7.2.2 Liste non exhaustive des fonctions de hachage cryptographiques

La figure 7.2.2 donne une liste non exhaustive des fonctions de hachage cryptographiques.

7.3 Exercices

Exercice 7.3.1 *1. Construisez une paire de clés RSA avec $p = 127$ et $q = 139$ en ayant le e le plus petit possible pour Alice et une paire de clés RSA avec $p = 113$ et*

Nom	Taille de la sortie en bit	Date de création	Date de mort
MD4	128	1990	1990
MD5	128	1991	1996
MD6	$0 - 512$	2008	
SHA-1	160	1995	2005
SHA-2	$224 - 512$	2002	$2008 - 2016$
SHA-3	arbitraire	2012	
Whirlpool	512	2000	
HAVAL	$128 - 256$	1994	2004

FIGURE 7.2 – Fonctions de hachage cryptographiques

$q = 149$ *en ayant le* e *le plus petit possible pour Bob.*

2. *Chiffrez et signez les messages "HE", "LP" et "ME" avec la clé de Bob.*
3. *Déchiffrez les messages* $(4286, 13489)$ *et "*$(15541, 14240)$*" avec la clé d'Alice et vérifiez que c'est bien Bob qui les a envoyés.*

Exercice 7.3.2 (Sur ordinateur) *Implémentez le chiffrement RSA signé ainsi que sa vérification.*

Exercice 7.3.3 *Soient un entier* $n > 1$ *et* $h : \{0,1\}^{2n} \to \{0,1\}^n$ *une fonction de hachage. On considère* $H : \{0,1\}^{4n} \to \{0,1\}^n$ *définie par* $H(x1||x2) = h(h(x1)||h(x2))$*, si* $x1$*,* $x2 \in \{0,1\}^{2n}$*. Montrer que si* h *est résistante aux collisions,* H *l'est aussi.*

Exercice 7.3.4 (L'attaque des anniversaires) *Soit* h *une fonction de hachage dont la longueur de sortie est* 256 *bits. Combien de message doit-on hacher pour être sûr à* 50% *d'avoir une collision ?*

Exercice 7.3.5 (MD5, Machine) *Le but de l'exercice est de vous faire implémenter la fonction de hachage MD5. On commence par implémenter quelques fonctions :*

— *Implémenter une fonction* $text\_to\_bin$ *qui prend en entrée une chaîne de caractère et qui renvoie le message en binaire.*
— *Implémenter une fonction* bin_to_hex *qui prend en entrée une chaîne de bits et qui la retourne en hexadécimale.*
— *Implémenter une fonction* bin_to_dec *qui prend en entrée une chaîne de bits et qui la retourne en base* 10*.*

- *Implémenter une fonction $shift$ qui prend en entrée une variable k et une suite de bits s. En sortie, elle renvoie la suite de bits s ayant subi une rotation de k bits vers la gauche. Sur le schéma, c'est noté $<<<_s$.*
- *Implémenter une fonction add_32 qui prend en entrée deux entiers, qui les additionne modulo 32 et renvoie le résultat. Sur le schéma, c'est noté $[+]$.*
- *Implémenter une fonction xor qui prend en entrée deux chaines de bits et qui renvoie le "xor" de ces deux chaines. On rappelle que $0 \oplus 0 = 0$, $1 \oplus 1 = 0$ et $1 \oplus 0 = 1$. Sur le schéma, c'est noté $\oplus$.*
- *Implémenter une fonction and qui prend en entrée deux chaines de bits et qui renvoie le "et" de ces deux chaines. On rappelle que $0 \wedge 0 = 0$, $1 \wedge 1 = 0$ et $1 \wedge 0 = 0$. Sur le schéma, c'est noté $\wedge$.*
- *Implémenter une fonction or qui prend en entrée deux chaines de bits et qui renvoie le "ou" de ces deux chaines. On rappelle que $0 \vee 0 = 0$, $1 \vee 1 = 1$ et $1 \vee 0 = 1$. Sur le schéma, c'est noté $\vee$.*
- *Implémenter une fonction not qui prend en entrée une chaine de bits et qui renvoie la négation de cette chaine. On rappelle que $\neg 0 = 1$ et $\neg 1 = 0$. Sur le schéma, c'est noté $\neg$.*
- *Implémenter une fonction $bourrage$ qui prend en entrée le message. Elle vérifie si le message fait un multiple de 512 bits. Si oui, elle renvoie le message initial. Sinon, on rajoute des 1 à la fin jusqu'à ce que le message mesure un multiple de 512 bits et on le renvoie.*
- *Implémenter une fonction $paquet$ qui prend en entrée un message, qui le découpe en paquet de 32 bits et renvoie ces paquets.*
- *Implémenter une fonction F qui prend en entrée quatre blocs de 8 bits A, B, C et D ainsi qu'un entier i entre 1 et 4. Si $i = 1$ alors la fonction renvoie $(B \wedge C) \vee (\neg B \wedge D)$. Si $i = 2$ alors la fonction renvoie $B \vee D \vee (\neg D \wedge C)$. Si $i = 3$ alors la fonction renvoie $B \oplus C \oplus D$. Si $i = 4$ alors la fonction renvoie $C \oplus (B \vee \neg D)$.*
- *Une fonction K qui prend en entrée un entier i et qui renvoie $\lfloor |2^{32} \times sin(i+1)| \rfloor$ en base 2 sur 32 bits. C'est cette fonction qui va générer nos valeurs K_i.*
- *Une fonction M qui prend en entrée un entier i et le message en bit. Elle revoit la i-ème chaine de 32 bits du message. C'est cette fonction qui va générer nos valeurs M_i.*
- *Écrire une fonction $MD5\_round$ qui prend en entrée des blocs de 32 bits, un entier i (pour le numéro du tour) et qui retourne une suite de bits suivant le schéma de la figure 7.3.*
- *Écrire la fonction $MD5$ qui prend en entrée un message et qui retourne le haché de celui-ci en utilisant 4 fois la fonction $MD5\_round$.*

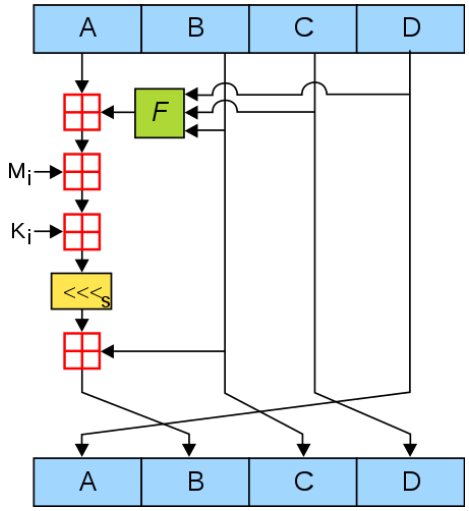

FIGURE 7.3 – Un tour de MD5.

Chapitre 8

Protocoles d'échange de clés

La cryptographie asymétrique, c'est super, mais très coûteux et lent (voir Figure 8.1), de ce fait, on s'en sert pour échanger des clés de cryptographie symétrique afin de communiquer plus rapidement. Alors pour ce faire, on peut simplement utiliser la cryptographie asymétrique, mais aussi faire ce que l'on appelle un protocole d'échange de clés. Le plus connu et le plus utilisé est le protocole de Diffie-Hellman que nous allons détailler.

8.1 Diffie-Hellman

L'objectif du protocole, c'est de faire en s se mettent d'accord sur une valeur secrète, sans qu'une personne extérieure ne puisse la connaître. La figure 8.2 bonne illustration du principe.

Définition 8.1.1 (Protocole DHKE) *Le protocole se déroule comme suit :*

1. *Alice et Bob ont choisi un nombre premier p et un nombre $0 < g < p$ qui sont publiques.*

	DVD 4,7 G.B		Blu-Ray 25 GB	
Schémas	chiffrer	déchiffrer	chiffrer	déchiffrer
RSA 2048	22 min	24 h	115 min	130 h
RSA 1024	21 min	10 h	111 min	53 h
AES CTR	20 sec	20 sec	105 sec	105 sec

FIGURE 8.1 – Coût en temps symétrique/asymétrique.

2. *Alice choisit un nombre a au hasard et envoie à Bob le nombre $A \equiv g^a\ [p]$.*
3. *Bob choisit un nombre b au hasard et envoie à Alice le nombre $B \equiv g^b\ [p]$.*
4. *Alice et Bob calculent alors leur clé commune $K \equiv A^b \equiv B^a \equiv g^{ab}\ [n]$.*

Théorème 8.1.1 ((admis)) *Soit p un nombre premier, le groupe $(\mathbb{Z}/p\mathbb{Z})^\times$ est cyclique d'ordre $p-1$ et tout élément $g \in (\mathbb{Z}/p\mathbb{Z})^*$ est générateur de ce groupe.*

Remarque 8.1.0.1 *De manière moins formelle ça veut dire que si p est premier alors quel que soit le $0 < g < p$, on peut écrire le groupe engendré par g noté $< g >$ de la manière suivante : $< g >= \{g^0\ [p], g^1\ [p], \ldots, g^{p-2}\ [p]\}$. Et c'est cyclique, car $g^{p-1} = 1$ (théorème 3.1.2), le premier élément du groupe, on tourne en rond d'où la dénomination cyclique.*

On écrit souvent le protocole de la manière décrite dans la figure 8.3

Exemple 8.1.0.1
1. *Alice et Bob ont choisi un nombre premier $p = 101$ et un nombre $g = 11$.*
2. *Alice choisit un nombre $a = 10$ au hasard et envoie à Bob le nombre $A \equiv g^a \equiv 11^{10} \equiv 17\ [101]$.*
3. *Bob choisit un nombre $b = 23$ au hasard et envoie à Alice le nombre $B \equiv 11^{23} \equiv 51\ [101]$.*
4. *Alice et Bob calculent alors leur clé commune $K \equiv A^b \equiv B^a \equiv g^{ab} \equiv 65\ [101]$.*

Une fois que l'on a une clé commune, on peut communiquer par chiffrement symétrique avec cette clé.

8.2 Problème du logarithme discret

La sécurité de ce protocole réside dans la difficulté du problème du logarithme discret. En effet, pour qu'un attaquant retrouve g^{ab} à partir de g^a et g^b, il doit élever l'un ou l'autre à la puissance b ou a. Cependant, trouver a (resp. b) de g^a (resp. g^b) est un problème NP-hard connu sous le nom de logarithme discret.

Définition 8.2.1 (Problème du logarithme discret) *Soit p un nombre premier, g un entier tel que $g < p$ et a un entier non nul. Alors, le problème du logarithme discret est le suivant : Sachant g et g^a modulo p, trouver a.*

Théorème 8.2.1 (Shoup, 1997 (Admis)) *Dans un groupe multiplicatif de n éléments, la résolution du problème du logarithme discret nécessite au moins $\sqrt{n}$ multiplications.*

Remarque 8.2.0.1 *On peut par exemple chercher des collisions avec le paradoxe des anniversaires que nous donne ce résultat.*

Il existe de nombreuses méthodes pour résoudre ce problème. On peut citer par exemple la brute force : on teste tous les a possibles. Et le plus utilisé, l'algorithme de Shanks : baby-step giant-step.

Définition 8.2.2 (Algorithme de Shanks) *Il permet de résoudre le problème du logarithme en effectuant environ $2\sqrt{n}$ multiplications dans le groupe $G =< g >$. On va procéder de la manière suivante :*

1. *On pose $m = \lceil \sqrt{n} \rceil$.*
2. *On pose ℓ la puissance visée telle que $y = g^{\ell}$.*
3. *On pose la division Euclidienne $\ell = mk + r$ avec $0 \leq r \leq m$.*
4. *On sait que $0 \leq k \leq m - 1$ car c'est comme ça que nous avons choisi m.*
5. *$y = g^{\ell}$ peut donc se réécrire $(g^m)^k = yg^{-r}$.*
6. *On fait les pas de géant et on stocke dans un tableau $PG = [(g^m)^0, \ldots, (g^m)^{m-1}]$.*
7. *Ensuite, on fait les pas de bébé, on calcule : $yg^{-0}, yg^{-1}, \ldots$ jusqu'à obtenir un élément yg^{-r} égal à un élément k du tableau.*
8. *On a donc : $(g^m)^k = yg^{-r}$ et on connait m, k et r donc, on en déduit $\ell = mk + r$*

Remarque 8.2.0.2 *Il y a plusieurs inconvénients à cette méthode :*
- *Il faut de la place en mémoire pour stoker la table PG qui peut être grande. En effet, la taille du tableau est environ $\sqrt{n}$.*
- *C'est tout de même un algorithme exponentiel donc il est long sur les grandes instances (c'est normal, car le logarithme discret est un problème NP-dur).*
- *Il faut connaitre un minimum d'informations sur le groupe G notamment son ordre et l'élément générateur utilisé. On peut tout de même supposer qu'on dispose de ces informations par le principe de Kerkcoff.*

Exercice 8.2.1 (Cours, machine) *Implémenter l'algorithme de Shanks.*

8.3 Autres algorithmes d'échanges de clés

Il existe de nombreux algorithmes qui font de l'échange de clés. En voici une petite liste :

— DHKE authentifié. — aPAKE — ECDHKE
— RSA — OPAQUE — MIME

8.4 Exercices

Exercice 8.4.1 *Mettez-vous par deux et jouer le protocole d'échange de clés Diffie-Hellman avec* $p = 547$ *et* $g = 42$.

Exercice 8.4.2 (Table de logarithme) *Écrire la table du logarithme discret de* 2 *pour* $p = 5, 7, 11, 13$ *et la table du logarithme discret de* 3 *pour* $p = 5, 7, 11, 13$.

Exercice 8.4.3 *Répondez aux questions suivantes :*

. *Vous savez que* $p = 19$ *et* $g = 7$, *retrouvez le* x *tel que* $7^x = 11$ $[19]$.
. *Vous savez que* $p = 17$ *et* $g = 3$, *retrouvez le* x *tel que* $3^x = 8$ $[17]$.
. *Vous savez que* $p = 29$ *et* $g = 17$, *retrouvez le* x *tel que* $17^x = 17$ $[29]$.
. *Vous savez que* $p = 41$ *et* $g = 22$, *retrouvez le* x *tel que* $22^x = 21$ $[41]$.
. *Vous savez que* $p = 53$ *et* $g = 11$, *retrouvez le* x *tel que* $11^x = 16$ $[53]$.
. *Vous savez que* $p = 67$ *et* $g = 16$, *retrouvez le* x *tel que* $16^x = 47$ $[67]$.
. *Vous savez que* $p = 79$ *et* $g = 30$, *retrouvez le* x *tel que* $30^x = 35$ $[79]$.
. *Vous savez que* $p = 97$ *et* $g = 47$, *retrouvez le* x *tel que* $47^x = 75$ $[97]$.
. *Vous savez que* $p = 37$ *et* $g = 15$, *retrouvez le* x *tel que* $15^x = 9$ $[37]$.
. *Vous savez que* $p = 59$ *et* $g = 32$, *retrouvez le* x *tel que* $32^x = 49$ $[59]$.

Exercice 8.4.4 *Répondez aux questions suivantes :*

. *Montrez que* 3 *est générateur d'un groupe cyclique dans* $\mathbb{Z}/22\mathbb{Z}$ *et calculez son cardinal.*
. *Montrez que* 7 *est générateur d'un groupe cyclique dans* $\mathbb{Z}/8\mathbb{Z}$ *et calculez son cardinal.*
. *Montrez que* 4 *est générateur d'un groupe cyclique dans* $\mathbb{Z}/13\mathbb{Z}$ *et calculez son cardinal.*
. *Montrez que* 13 *est générateur d'un groupe cyclique dans* $\mathbb{Z}/14\mathbb{Z}$ *et calculez son cardinal.*
. *Montrez que* 22 *est générateur d'un groupe cyclique dans* $\mathbb{Z}/23\mathbb{Z}$ *et calculez son cardinal.*
. *Montrez que* 2 *est générateur d'un groupe cyclique dans* $\mathbb{Z}/5\mathbb{Z}$ *et calculez son cardinal.*

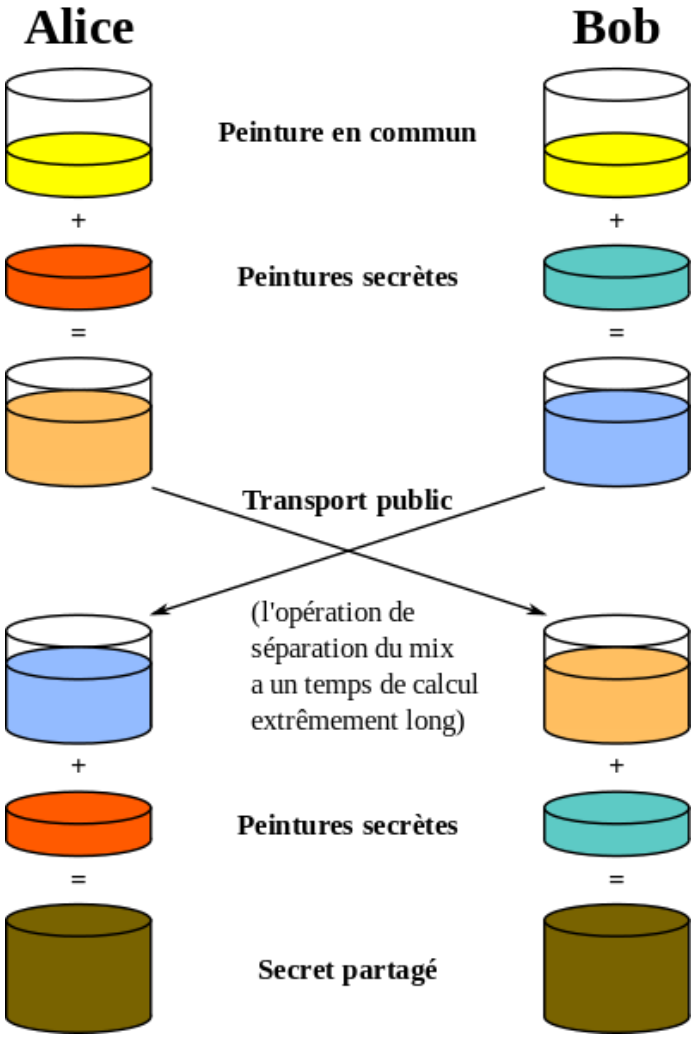

FIGURE 8.2 – Diffie-Hellman pour les nuls.

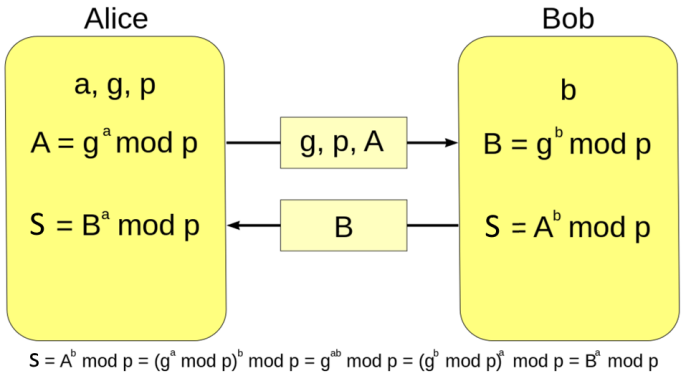

FIGURE 8.3 – Protocole Diffie-Hellman Key Exchange.

Chapitre 9

Preuves à divulgation nulle de connaissances (ZKP)

Les zero-knowledge proofs (ZKP), ou preuves à divulgation nulle de connaissances, sont des protocoles permettant à une personne de prouver à une autre personne qu'elle connait un secret sans jamais le dévoiler. Il n'existe pas de protocole ZKP universel. En effet, chaque cas particulier nécessite une mise en situation unique et le protocole est développé en fonction de ces paramètres. Nous allons étudier plusieurs protocoles pour comprendre comment ça marche.

9.1 La grotte d'Ali-Baba

On considère Alice et Bob devant la grotte d'Ali Baba. Alice connaît le mot de passe pour ouvrir la porte. Celle-ci bloque le passage de deux couloirs A et B. Bob souhaite passer la porte, mais Alice ne lui fait pas confiance et refuse de lui donner le mot de passe. Bob doute alors du fait qu'elle connaisse bien le mot magique. Le fonctionnement du protocole ZKP associé à cette situation serait qu'Alice prouve à Bob qu'elle a le mot de passe en ouvrant la porte sans pour autant lui dire. Le protocole va se dérouler comme suit :

1. Alice va donc au fond de la grotte, attendant les instructions de Bob comme dans la figure 9.1.
2. Bob sélectionne alors l'un des couloirs par lequel Alice doit pouvoir ressortir comme dans la figure 9.2.
3. Alice, ressort par le couloir sélectionné par Bob comme dans la figure 9.3.

Ce protocole réduit à 50% les chances qu'elle ait choisi au premier essai, le chemin correct choisi plus tard par Bob. La répétition de ce schéma à plusieurs reprises sert alors à déterminer qu'Alice connaît vraiment le mot magique pour ouvrir la porte, mais à aucun moment elle ne l'a dit à Bob.

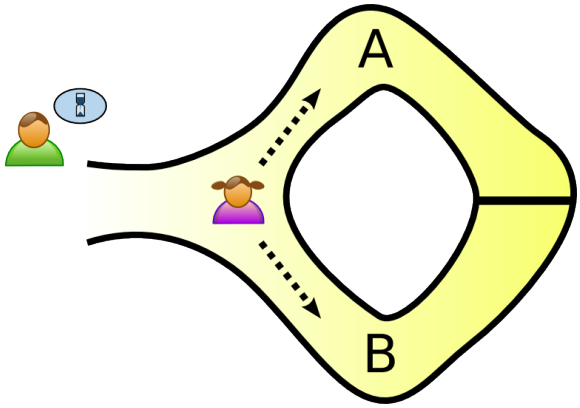

FIGURE 9.1 – Alice s'engage dans un tunnel.

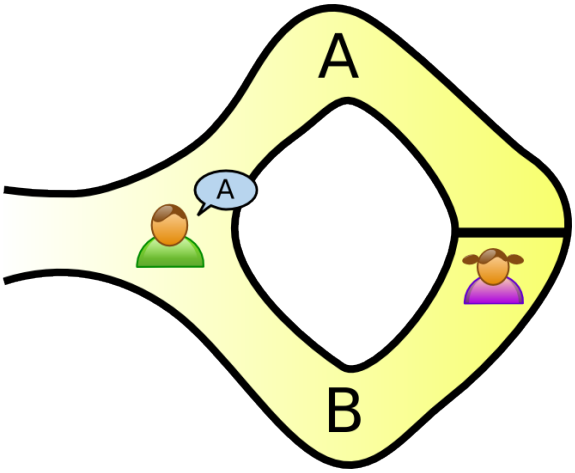

FIGURE 9.2 – Bob lui demande de sortir par un tunnel.

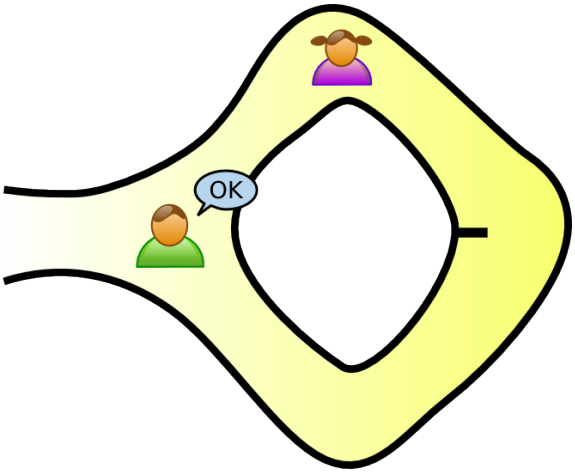

FIGURE 9.3 – Alice réussi le défi.

9.2 Les bonbons d'Halloween

De retour de leur récolte d'Halloween, Alice et Bob souhaitent connaître s'ils ont obtenu le même nombre de bonbons sans pour autant révéler à l'autre le nombre exact qu'ils possèdent.

Pour se faire, Bob va créer des boites pour chaque possibilité de récolte et y inscrire un nombre à l'intérieur. Il les ferme à clé et jette les clés des boites contenant un nombre erroné comme montrer dans la figure 9.4.

FIGURE 9.4 – Bob crée n boites avec des possibilités de récolte.

Alice, quant à elle, va mettre dans toutes les boites le signe "-" sur un bout de papier, sauf pour une boite où elle mettra un "+" (celle qui contient son nombre exact de bonbons) comme montrer dans la figure 9.5.

Par la suite, Bob va récupérer la seule boite à laquelle il a accès, c'est-à-dire celle qui contient le nombre de bonbons qu'il possède et va prendre connaissance

FIGURE 9.5 – Alice met ses bouts de papiers dans les coffres.

du signe associé. Si c'est un "+", ils ont le même nombre. Dans le cas contraire, leurs récoltes sont différentes. Dans les deux cas aucun des deux n'a eu accès à la récolte de l'autre puisque Bob n'a pas les clés des autres boites et ne peut donc pas chercher celle qui contient le "+". À votre avis, sur l'exemple, Bob et Alice ont-ils le même nombre de bonbons ?

9.3 La daltonienne et les boules colorées

Il nous est déjà arrivé de ne pas pouvoir différencier un objet d'un autre alors qu'une autre personne y arrive. En appliquant le protocole suivant, vous saurez si cette personne est capable de différencier deux objets en apparence identique sans qu'elle vous révèle la façon dont elle procède.

Soit A et B deux cartes identiques de couleur différente [1], une daltonienne Alice et une personne discernant les couleurs Bob. Les deux dos cartes sont parfaitement indistinguables pour Alice. L'objectif est de prouver à Alice que Bob est capable de distinguer les cartes sans que jamais leurs couleurs (rouge ou bleu) ne soient révélées.

Alice place les cartes derrière son dos, puis en prend une et la montre à Bob. Elle la remet dans son dos puis choisit de montrer une des cartes, soit la même, soit l'autre avec une probabilité de 50%. Alice demande donc à Bob si elle a changé de carte. Le processus est répété autant de fois qu'il le faut. Si le protocole est exécuté suffisamment de fois et si Bob donne toujours une réponse correcte, Alice sera convaincue avec une très forte probabilité que son partenaire est effectivement ca-

1. L'exemple provient de Oded Goldreich.

pable de différentier les cartes par leur couleur sans pour autant avoir appris la couleur de celle-ci.

9.4 La grille de Sudoku

C'est à vous de jouer pour celle-ci. Alice veut prouver à Bob qu'elle connaît la solution d'une grille standard de Sudoku. À la fin de la procédure, Bob a la preuve qu'Alice a la solution de la grille de Sudoku sans la connaître. Quel protocole mettriez-vous en place ?

Chapitre 10

La taille des clés dans le standard

Dans ce chapitre, on donne la liste des algorithmes de chiffrement utilisés aujourd'hui ainsi que la taille des clés associées. La taille des clés nous permet de se protéger des attaques connues et sont mises à jour tous les ans. Pour plus de détails, consultez le site de keylength.

Algorithme	Sécurité Minimale	Sécurité Intermédiaire	Forte Sécurité
3-DES	168	X	X
AES	128	256	X
HMAC	80	96	112
DSA	1024	2048	X
DH	2048	X	X
ECC	192	224	521
RSA	1024	2048	4096

Table 10.1 – Taille des clés

Chapitre 11

Les preuves importantes

Parce qu'en mathématiques, tout en dehors des axiomes repose sur des preuves, car ce qui peut être énoncé sans preuves peut être réfuté sans preuves.

Preuve de la proposition 2.1.1

Preuve 11.0.1 (Proposition 2.1.1) *On va prouver chacun des points de la proposition :*

- *Soient $a, b, c \in \mathbb{Z}$ tels que $c|a$ et $c|b$. Alors, il existe $k, k' \in \mathbb{Z}$ tel que $a = k \times c$ et $b = k' \times c$. Donc, pour tout $u, v \in \mathbb{Z}$, $au = \underbrace{ku}_{K} \times c$ et $bv = \underbrace{k'v}_{K'} \times c$. Ainsi, $au + bv = (K + K') \times c$. Donc, $c|au + bv$.* ■
- *Soient $a, b \in \mathbb{Z}$ tels que $b|a$ et $a|b$. Alors, il existe $k, k' \in \mathbb{Z}$ tel que $a = k \times b$ et $b = k' \times a$. Donc, $a = kk' \times a$ et $b = kk' \times b$. De ce fait, $kk' = 1$ donc $k = k' = 1$ ou $k = k' = -1$. Ainsi, $a = b$ ou $a = -b$.*■
- *Soient $a, b, c \in \mathbb{Z}$ tels que $c|b$ et $b|a$. Alors, il existe $k, k' \in \mathbb{Z}$ tel que $a = k \times b$ et $b = k' \times c$. On peut écrire que $a = kk' \times c$. Donc $c|a$.*■
- *Pour tout $a \in \mathbb{Z}$ $a = 1 \times a$ donc $a|a$.*■
- *Pour tout $a \in \mathbb{Z}$ $a = 1 \times a$ donc $1|a$.*■
- *Pour tout $a \in \mathbb{Z}$ $0 = 0 \times a$ donc $a|0$.*■

Preuve de la proposition 2.1.2

Preuve 11.0.2 (Proposition 2.1.2) *Soit $n \in \mathbb{Z}$ un nombre non premier. Comme n n'est pas premier, on peut l'écrire $p \times q$. Supposons par l'absurde que $p > \sqrt{n}$ et $q > \sqrt{n}$ alors,*

$p \times q > \sqrt{n}^2 = n$. Donc, par l'absurde, $p \leq \sqrt{n}$ ou $q \leq \sqrt{n}$.■

Preuve de la proposition 3.1.1

Preuve 11.0.3 (Proposition 3.1.1) *On va montrer les 3 propriétés de la proposition :*

- *Soient $a, n \in \mathbb{Z}$ avec $a < n$. Alors, la division euclidienne de a par n est $a = 0 \times n + a$. Donc $a \equiv a\,[n]$.*■
- *Soient $a, b, n \in \mathbb{Z}$. Si $a \equiv b\,[n]$ alors le reste de la division euclidienne de a par n est le même que le reste de la division euclidienne de b par n. Donc $b \equiv a\,[n]$.*■
- *Soient $a, b, c, n \in \mathbb{Z}$. Si $a \equiv b\,[n]$ alors le reste de la division euclidienne de a par n est le même que le reste de la division euclidienne de b par n. De plus, si $b \equiv c\,[n]$ alors le reste de la division euclidienne de c par n est le même que le reste de la division euclidienne de b par n. Donc le reste de la division euclidienne de a par n est le même que le reste de la division euclidienne de c par n. Ainsi, $c \equiv a\,[n]$.*■

Preuve de la proposition 3.1.2

Preuve 11.0.4 (Proposition 3.1.2) *Soit $a, n \in \mathbb{Z}$ tel que $a < n$ et $PGCD(a, n) = 1$. Alors, il existe un couple $(u, v) \in \mathbb{Z}^2$ tel que $au + nv = 1$. Donc, $au + \underbrace{n}_{0} v \equiv 1\,[n]$ donc $au \equiv 1\,[n]$. Ainsi, a est inversible modulo n et son inverse est u qui est obtenu par une relation de Bézout.*■

Preuve de la proposition 3.1.3

Preuve 11.0.5 (Proposition 3.1.3) *Soit p un nombre premier. Alors, pour tout $a \in \mathbb{Z}$ tel que $a < p$, $PGCD(a, p) = 1$. Donc a est inversible en utilisant la proposition 3.1.2. Ainsi, si p est premier, tout élément non nul de $\mathbb{Z}/n\mathbb{Z}$ est inversible.*■

Preuve du théorème 2.1.3

Preuve 11.0.6 (Théorème 2.1.3) *Soit d le PGCD de a et b et soit k un nombre entier non multiple de d.*

Supposons qu'il existe deux entiers p et q tels que $ap + bq = k$. Par définition d divise a et d divise b donc d divise la combinaison linéaire $ap + bq$. Or, $ap + bq = k$ donc d divise k,

ce qui est contraire à la donnée, donc p et q n'existent pas et l'équation n'a pas de solutions entières. ■

Preuve du théorème 3.1.1

Preuve 11.0.7 (Théorème 3.1.1) *Soit $a, n \in \mathbb{Z}$ tel que $a < n$ et $PGCD(a, n) = 1$. Alors, il existe un couple $(u, v) \in \mathbb{Z}^2$ tel que $au + nv = 1$. Donc, $au + \underbrace{n}_{0} v \equiv 1\,[n]$ donc $au \equiv 1\,[n]$. Ainsi, a est inversible modulo n et son inverse est u qui est obtenu par une relation de Bézout. Donc $PGCD(a, n) = 1 \rightarrow a$ inversible. Supposons que a soit inversible modulo n alors, il existe u tel que $au \equiv 1\,[n]$. Ainsi, on peut écrire la division euclidienne de au par n : $au = qn + 1$ et donc, $au + \underbrace{v}_{-q} n = 1$. De ce fait, avec le même argument que pour le théorème 2.1.3, $PGCD(a, n) = 1$. Donc $PGCD(a, n) = 1 \leftrightarrow a$ inversible*■

Preuve du théorème 3.1.2

Preuve 11.0.8 (Théorème 3.1.2) *Supposons que $a \in \mathbb{Z}/p\mathbb{Z}$ n'est pas divisible par p et notons N le produit $a \times 2a \times 3a \cdots \times (p-1)a$, r_k le reste de la division euclidienne de ka par p, pour tout entier k de 1 à $p-1$.*

Alors, on peut faire 3 observations :

- *En réordonnant les facteurs, $N = 1 \times 2 \times \cdots \times (p-1) \times a^{p-1} = (p-1)!a^{p-1}$.*
- *$N \equiv r_1 \times r_2 \times \cdots \times r_{p-1}$ mod p. En effet, si dans un produit on remplace un facteur par un entier qui lui est congru modulo p alors le nouveau produit est congru modulo p à l'ancien. En remplaçant dans N, un par un, chaque ka par r_k, le résultat est donc congru à N modulo p.*
- *$r_1 \times r_2 \times \cdots \times r_{p-1} = (p-1)!$ car $(r_1, r_2, \ldots, r_{p-1})$ est une permutation de $(1, 2, \ldots, p-1)$. En effet, $0 \leq rk \leq p-1$ et aucun r_k n'est nul, car (d'après le lemme d'Euclide) aucun ka n'est divisible par p; De plus, les r_k sont distincts deux à deux, car si $r_i = r_j$ alors $(i-j)a$ est divisible par p, donc (à nouveau par le lemme d'Euclide) $i - j$ aussi, donc (comme $-p < i - j < p$) $i = j$.*

Des trois points précédents on déduit : $(p-1)!a^{p-1} \equiv (p-1)!$ mod p, autrement dit $(p-1)!(a^{p-1}-1)$ est divisible par p. Par application répétée du lemme d'Euclide, on obtient la conclusion : $a^{p-1} - 1$ est divisible par p donc, $a^{p-1} \equiv 1\,[p]$.■

Preuve 11.0.9 (Théorème 6.1.1) *Soient $n > 0$ et a un entier premier avec n. Notons $\overline{a}$ la classe de a un entier inversible modulo n. Prenons la fonction bijective $(\mathbb{Z}/n\mathbb{Z})^{\times} \rightarrow$*

$(\mathbb{Z}/n\mathbb{Z})^\times$ qui à x associe $\overline{a}x$. On écrit le produit de tous les éléments inversibles de $\mathbb{Z}/n\mathbb{Z}$: $P = \prod_{x\in(\mathbb{Z}/n\mathbb{Z})^\times} x$. *Par bijection, on peut écrire :*

$$\begin{aligned} P &= \prod_{x\in(\mathbb{Z}/n\mathbb{Z})^\times} (\overline{a}x) \\ P &= P \times (\overline{a})^{card\left((\mathbb{Z}/n\mathbb{Z})^\times\right)} \\ P &= P \times (\overline{a})^{\varphi(n)} \\ 1 &= (\overline{a})^{\varphi(n)} \end{aligned}$$

Ainsi, pour tout a inversible modulo n, $1 \equiv a^{\varphi(n)}\ [n]$. ■

Chapitre 12

Sources supplémentaires

Dans ce chapitre, vous trouverez la liste des sources utiles que vous pouvez consulter pour approfondir vos connaissances ou seulement par curiosité.

Les sites web utiles :

1. Le site rapidtables pour la conversion des entiers dans différentes bases.
2. Le site keylength pour consulter la taille des clés recommandées.
3. Le site dcode pour faire rapidement et facilement de la crypto.

Les cours et exercices en ligne :

1. Le cours et les TD de Cristine Bachoc de Codes et Cryptologie.
2. Le cours et les TD de Guillem Castagnos de Cours d'arithmétique et cryptographie.
3. Le site de vulgarisation bibmath.

Pour ce qui est des livres :
— Cours de cryptographie de Gilles Zemor.
— La cryptologie au cœur du numérique de Jacques Stern.
— Exercices et problèmes de cryptographie de Damien Vergnaud.
— Initiation à la cryptographie de Gilles Dubertret.
— 25 énigmes ludiques pour s'initier à la cryptographie de Malika More et Pascal Lafourcade.
— Introduction to Modern Cryptography : Principles and Protocols de Jonathan Katz et Yehuda Lindell.

Pour aller plus loin dans les protocoles ZKP, regardez le protocole de Schnorr et protocole de Guillou-Quisquater.

Pour aller plus loin sur la stéganographie, regardez le papier de Ljupce Nikolov : Stéganographie Détection de messages cachés.