**RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE**
**MINISTRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE**

**UNIVERSITÉ ABDERRAHMAN MIRA-BEJAIA-ALGERIE**
**FACULTÉ DES SCIENCES EXACTES**
**DÉPARTEMENT DE MATHÉMATIQUES**


***Mémoire présenté pour l'obtention du diplôme de Master en Mathématiques***

**OPTION**
Analyse Mathématiques

**THÈME**

# *Introduction à la théorie des nombres et la cryptographie*

**PAR**
BENHARRAT Badri

Devant le jury :

| | | | | |
|---|---|---|---|---|
| *M.KHELOUFI* | *Arezki* | *Professeur* | *U.A.M.BEJAIA* | **Président** |
| *M.MEKERRI* | *Toufik* | *MAA* | *U.A.M.BEJAIA* | **Examinateur** |
| *M.AISSAOUI* | *Said* | *MCB* | *U.A.M.BEJAIA* | **Encadrant** |

**Soutenu publiquement le** :25 - 06 - 2023

# Remerciements

Je tient d'abord à exprimer mes plus vifs remerciements et ma profonde gratitude à mon encadrant **M. AISSAOUI** pour l'honneur qu'il m' a fait de m'encadrer.
Sa qualité de son encadrement, sa compétence, sa rigueur scientifique et sa clairvoyance m'a beaucoup aidé. je le remercier aussi pour son aide à la réalisation de ce travail.

Je remercie également les membres du jury, **M. KHELOUFI** , **M. MEKERRI** pour l'honneur qu'ils m'ont fait d'accepter l'évaluation de ce modeste travail.

Je remercie tous les enseignants du département de mathématiques qui ont contribué à ma formation.

# Dédicace

C'est avec un grand plaisir que je dédie ce modeste travail :

Á mes très chers parents, que Dieu les préserve et les accorde bonne santé et longue vie.

Á mes frères et soeurs.

Á tous mes ami(e)s et à tous ceux qui m'aiment.

Á tous mes enseignants et toute la promotion mathématique 2022-2023.

Á M. AISSAOUI pour son aide et ses précieux conseilles.

Á tous les étudiants qui, j'espère, pourront en tirer bénéfice ne serait-ce qu'un peu de ce travail

# Table des matières

# Introduction

La sécurité de l'information est devenue un enjeu crucial dans notre société de plus en plus connectée. Que ce soit pour les transactions bancaires en ligne, les communications confidentielles ou même la protection des données personnelles, la nécessité de garantir la confidentialité et l'intégrité des informations échangées est primordiale. La cryptographie, l'art de sécuriser les communications, joue un rôle central dans cet objectif. Cependant, la cryptographie ne repose pas sur des principes magiques, mais plutôt sur des fondements mathématiques solides, dont la théorie des nombres constitue l'un des piliers essentiels.

Ce mémoire se veut une introduction à la théorie des nombres et à la cryptographie, en mettant en évidence les liens étroits qui existent entre ces deux domaines. La théorie des nombres, l'une des branches les plus anciennes des mathématiques, est un vaste domaine consacré à l'étude des propriétés des nombres entiers.avec des résultats fascinants tels que le théorème de Fermat, le dernier théorème de Fermat ou encore le théorème des nombres premiers.

La cryptographie, quant à elle, est l'ensemble des techniques permettant de protéger l'information en la transformant de manière sécurisée, de sorte qu'elle ne puisse être comprise que par les personnes autorisées. La cryptographie moderne repose sur des algorithmes sophistiqués et des méthodes mathématiques avancées, dont certains sont directement inspirés par la théorie des nombres. Par exemple, les systèmes de chiffrement à clé publique, tels que le célèbre RSA, se basent sur des concepts tels que les nombres premiers et les congruences.

Dans ce mémoire, nous rappelons les fondamentaux de la théorie des nombres, tels que les nombres premiers, les congruences, les résidus quadratiques, les groupes et les anneaux. Nous étudierons également comment ces résultats sont utilisés en cryptographie, en mettant en lumière des algorithmes classiques tels que le chiffrement de César, le chiffrement de Vigenère et le chiffrement à clé publique RSA.

Ce mémoire se compose de trois chapitres :
Le premier chapitre est consacré à la présentation de quelques notions indispensables à la théorie des nombres : groupes, anneaux, division euclidien, congruences, et surtout les nombres premiers et à la fin par quelque tests de primalié qu'on a besoin pour la cryptographie. Dans le deuxième chapitre nous étudions la cryptographie et son importance dans le passé jusqu'à aujourd'hui, ses types ,et on termine le chapitre par le chiffrement RSA qui vraiment pratique et très indispensable dans le monde entiers. Pour le troisième chapitre et le dernier nous exposerons quelques programmes

des tests de primalités et de chiffrement RSA, en utilisant le langage C++.

CHAPITRE 1

# Introduction à la théorie des nombres

L'introduction á la théorie des nombres est une branche des mathématiques qui étudie les propriétés des nombres entiers et des structures algébriques qui leur sont associées, tels que les nombres premiers, les congruences et les divisions. C'est l'une des plus anciennes branches des mathématiques, remontant à l'Antiquité, et elle continue d'être un domaine actif de recherche.

La théorie des nombres aborde des questions fondamentales telles que la distribution des nombres premiers, la factorisation des entiers en produits de nombres premiers, les congruences et les résidus quadratiques. Elle explore également des concepts plus avancés tels que les formes quadratiques, les courbes elliptiques.

La recherche en théorie des nombres a conduit à la découverte de nombreux résultats surprenants et profonds, tels que le théorème des nombres premiers, qui montre la distribution asymptotique des nombres premiers, et le dernier théorème de Fermat, qui a été résolu après des siècles de recherches approfondées.

La théorie des nombres a également des applications pratiques, notamment en cryptographie, où elle est utilisée pour concevoir des systèmes de cryptage sûrs et résistants aux attaques. Elle joue également un rôle dans d'autres domaines des mathématiques, tels que l'algèbre et la géométrie. Dans ce chapitre, nous tenterons de rappeler quelques résultats et propriétés fondamentales.

## 1.1 Rappels de quelques résultats de l'arithmétique

### 1.1.1 Structures de groupes et d'anneaux

**Definition 1.1.1** *On appelle groupe un ensemble $G$ muni d'une loi interne $\times$ telle que :*

1. *la loi $\times$ est associative :pour tous $x, y, z$ de $G$, on a $x \times (y \times z) = (x \times y) \times z$.*
2. *il existe un élément neutre $e$ : pour tout $x$ de $G$, $x \times e = e \times x = x$.*
3. *tout élément possède un symétrique : pour tout $x$ de $G$, il existe $y$ de $G$ avec $x \times y = y \times x = e$.*

**Remarque 1.1.1** *Si la loi $\times$ est commutative, on parle de groupe commutatif (ou abélien).*
*On peut démontrer qu'un groupe admet un unique élément neutre et qu'un élément $x$ admet un unique symétrique et on note $x^{-1}$.*

**Exemple 1.1.1** *$\mathbb{Z}$, muni de $+$, est un groupe. $\mathbb{R}^*$, muni de $\cdot$ (la multiplication usuelle), est un groupe. $\mathbb{N}$, muni de $+$, n'est pas un groupe : 3 n'admet pas de symétrique. L'ensemble des bijections d'un ensemble $X$, muni de la composition des fonctions, est un groupe : on l'appelle groupe des permutations de $X$.*

**Definition 1.1.2** *Soit $A$ un ensemble muni de deux lois de composition interne $+$ et $\times$. On dit que $(A, +, \times)$ est un anneau si :*

1. *$(A, +)$ est un groupe commutatif.*
2. *$\times$ est associative.*
3. *$\times$ est distributive par rapport à l'addition.*

**Remarque 1.1.2** *Lorsque la loi $\times$ est commutative, on dit que l'anneau est commutatif.*

**Exemple 1.1.2** *$(\mathbb{Z}, +, \times)$, $(\mathbb{R}, +, \times)$, $(\mathbb{C}, +, \times)$ : ce sont des anneaux commutatifs.*
*$(\mathcal{M}_n(\mathbb{R}), +, \times)$ est un anneau mais il n'est pas commutatif.*

## 1.1.2 Divisibilité et congruences dans l'anneau des entiers relatifs $\mathbb{Z}$

### Divisibilité dans l'anneau des entiers relatifs $\mathbb{Z}$

**Definition 1.1.3** *On dit que l'entier $d$ est un diviseur de l'entier $n$ et on écrit $d|n$ s'il existe un entier $q$ tel que $n = qd$. On dit aussi que $n$ est un multiple de $d$. Les diviseurs propres de $n$ sont les diviseurs autres que 1 et $n$.*

**Théorème 1.1.1** *Soient $d, n, m, a$ et $b$ des entiers relatifs, la divisibilité dans $\mathbb{Z}$ a des propriétés suivantes :*

1. *$n|n$ (réflexivité)*
2. *$d|n$ et $n|m$ implique $d|m$ (transitivité)*
3. *$d|n$ et $d|m$ implique $d|(an + bm)$ (linéarité)*
4. *$d|n$ implique $ad|an$ (multiplication)*

5. $1|n$
6. $0|n$ *implique* $n = 0$
7. $n|0$
8. $d|n$ *et* $d \neq 0$ *implique* $\frac{n}{d}/n$

**Proposition 1.1.1** *1. n est divisible par* 2 *si et seulement s'il se termine par* 0, 2, 4, 6, 8.

2. *n est divisible par* 3 *si et seulement si la somme de ses chiffres est divisible par* 3.
3. *n est divisible par 4 si et seulement si ses deux derniers chiffres forment un multiple de* 4. *(ex :* 256632*).*
4. *n est divisible par* 5 *si et seulement s'il se termine par* 0 *ou* 5.
5. *n est divisible par* 8 *si et seulement si ses* 3 *derniers chiffres forment un multiple de* 8 *(ex :* 176024*).*
6. *n est divisible par* 9 *si et seulement si la somme de ses chiffres est un multiple de* 9 *(ex :* 37521 *car* $3+7+5+2+1 = 18 = 2 \times 9$*).*
7. *n est divisible par 11 si et seulement si la différence (*1*er chiffre +* 3*ème chiffre +* 5*ème chiffre+...)-(*2*ème chiffre+*4*ème chiffre +* 6*ème chiffre+...) est divisible par* 11*. Par exemple,* 1485 *est divisible par* 11*, car* $(1+8)-(4+5) = 0$ *est divisible par* 11 .

**Plus grand coommun diviseur (PGCD)**

**Definition 1.1.4** *Soient a, b, d des entiers relatifs, on dit que d est un diviseur commun de a et b si d divise a et b.*

**Exemple 1.1.3** *1.* 2 *est un diviseur commun de* 4 *et* 6.

2. 5 *est un diviseur commun de* 25 *et* 45.

**Théorème 1.1.2** *Soient a et b deux entiers relatifs, il existe un diviseur commun d de a et b de la forme* $d = ak + bk$ *avec k et k' entiers relatifs . De plus tout diviseur commun de a et b divise d.*

**Démonstration 1.1.1** *(1) Premièrement, on suppose* $a \geq 0$ *et* $b \geq 0$ *, on utilise le raisonnement par récurrence sur n où* $n = a + b$*. Si* $n = 0$ *alors* $a = b = 0$ *et on prend* $d = 0$ *avec* $k = k = 0$*.*
*On suppose que le théorème est vrai jusqu'à (n-1).*
*Par symétrie on peut supposer que* $a \leq b$*, si* $b = 0$ *on prend* $d = a$*,* $k = 1$ *et* $k = 0$*. Si* $b \geq 1$*,*
*on applique le théorème à* $(a - b)$ *et b, comme* $(a - b) + b = a = (n - b) \leq (n - 1)$ *alors d'après l'hypothèse de récurrence, il existe d le diviseur commun de* $(a-b)$ *et b de la forme* $d = (a-b)k+bk$*.*
*Ce nombre d divise aussi a. Donc d est un diviseur commun de a et b qui s'écrit sous la forme* $d = ak + b(k - k)$*. Par linéarité, on montre que tout diviseur commun de a et b divise d .*
*(2) Si* $a < 0$ *ou* $b < 0$ *(ou les deux), on applique le résultat sur /a/ et /b/ et donc il existe d le diviseur commun de /a/ et /b/de la forme :*

$$d = /a/k + /b/k$$

*si a<0 alors* $|a|k = a(-k)$ *de même pour b (si b<0 alors* $|b|k = b(-k)$*) et donc dans tous les cas d est une combinaison linéaire de a et b.*

**Definition 1.1.5** *Le nombre d du théorème précèdent est appelé* ***plus grand diviseur commun*** *de a et b noté* $a \wedge b$ *ou (a, b).*

**division Euclidienne**

**Théorème 1.1.3** *Soient* $(a, b) \in \mathbb{Z}^2$ *avec* $b \neq 0$*. Il existe un unique couple* $(q, r) \in \mathbb{Z}^2$ *tels que*

$$\begin{cases} a = bq + r \\ 0 \leq r < |b|. \end{cases}$$

*q s'appelle le quotient et r s'appelle le reste de la division.*

**Théorème 1.1.4** ***Algorithme d'Euclide*** *Soit a et b deux entiers relatifs avec* $b \neq 0$*. Soit r le reste dans la division euclidienne de a par b. Alors* $a \wedge b = b \wedge r$*.*

**Exemple 1.1.4** *On calcule* $102 \wedge 30$
*On a :* $102 = 30 \times 3 + 12$*, alors* $102 \wedge 30 = 30 \wedge 12$*,*
*on a* $30 = 12 \times 2 + 6$*, alors* $30 \wedge 12 = 12 \wedge 6 = 6$ *.*
*D'où* $(102 \wedge 30) = 6$*.*

### 1.1.3 congruences dans l'anneau des entiers relatifs $\mathbb{Z}$

**Definition 1.1.6** *Soient* $a, b, m$ *des entiers relatifs avec* $m \geq 1$*, on dit que a est congrue à b modulo m et on écrit* $a \equiv b \mod m$*.*
*si m divise* $a - b$ *alors m est appelé le module de congruence . la congruence est équivalente à la divisibilité :*

$$\boxed{a \equiv b \mod m \iff m|(a-b).}$$

*En particulier* $a \equiv 0 \mod m$ *si et seulement si* $m|a$ *ainsi* $20 \equiv 0 \mod 5$ *si et seulement si* $a - b \equiv 0 \mod m$*.*
*Si* $m \not|(a-b)$ *alors on écrit* $a \not\equiv b \mod m$

**Exemple 1.1.5**

1. $19 \equiv -1 \mod 20$, $7 \equiv 2 \mod 5$ , $20 \equiv 0 \mod 5$
2. *n est pair si et seulement si* $n \equiv 0 \mod 2$
3. *n est impair si et seulement si* $n \equiv 1 \mod 2$

**Proposition 1.1.2** *la congruence est une relation d'équivalence,on a :*

1. $a \equiv a \mod m$ *(réflexivité)*
2. $a \equiv b \mod m$ *implique* $b \equiv a \mod m$*(symétrie)*
3. $a \equiv b \mod m$ *et* $b \equiv c \mod m$ *alors* $a \equiv c \mod m$ *(transitivité)*

***Démonstration 1.1.2*** *[1]*

1. $m|0$ , $a - a = 0 = 0.m$
2. *si* $m|(a - b)$ *alors* $m|(b - a)$
3. *si* $m|(a - b)$ *et* $m|(b - c)$ *alors* $m|(a - b) + (b - c) = a - c$

# 1.2 Nombres premiers

## 1.2.1 Définitions et propriétés

**Definition 1.2.1** *Soit $n$ un entier naturel. $n$ est un* ***nombre premier*** *s'il admet exactement deux diviseurs dans $\mathbb{N}$ : 1 et lui-même.*

**Théorème 1.2.1** *Soit $n \in \mathbb{N}$ , $n \geq 2$. Si $n$ n'est pas premier, il admet au moins un diviseur premier : son plus petit diviseur dans $\mathbb{N}$ autre que 1 est premier.*

**Démonstration 1.2.1** *On va faire un raisonnement par l'absurde. Si $n$ n'est pas premier donc $n$ admet au moins un diviseur strict (distinct de $n$) strictement plus grand que 1. Soit $p$ le plus petit de ces diviseurs. On a $1 < p < n$. Supposons que $p$ ne soit pas un nombre premier, alors il existe $d \in \mathbb{N}$ tel que :$1 < d < p$ et $d$ divise $p$. Alors $d$ divise $n$, ce qui est impossible. $p$ est donc un nombre premier.*

**Théorème 1.2.2** *L'ensemble des nombres premiers est un ensemble infini.*

**Démonstration 1.2.2** *Raisonnement par l'absurde : Euclide avait supposé que l'ensemble des nombres premiers est égal $p_1, p_2, \ldots, p_n$ et il avait considéré l'entier $m = p_1 p_2 \ldots p_n + 1$ Alors cet entier est premier ou bien il possède un diviseur premier. Or aucun des nombres premiers $p_1, p_2, \ldots, p_n$ ne peut diviser $m$ ; donc il y a une contradiction. Il en déduit que l'ensemble des nombres premiers est infini.*

**Corollaire 1.2.1** *Si $n$ n'est divisible par aucun entier $p$ premier tel que $2 \leq p \leq \sqrt{n}$, alors $n$ est premier.*

**Démonstration 1.2.3** *Démonstration par contraposée : « si $P$ vraie alors $Q$ vraie » équivaut à « si $Q$ faux alors $P$ faux ». On va supposer que $n$ n'est pas premier et on va démontrer qu'alors il admet un diviseur premier inférieur ou égal à $\sqrt{n}$ . Si $n$ n'est pas premier, d'après le théorème précédent, il admet un diviseur premier $p$ qui est son plus petit diviseur , $1 < p < n$. Alors $n = p \times q$, $q$ est aussi un diviseur de $n$ donc $p \leq q$ et $p^2 \leq pq$ soit$p^2 \leq n$ ou encore $p \leq \sqrt{n}$.*

**Exemple 1.2.1** *Le nombre* 127 *est-il premier ? Comme* $\sqrt{127} \simeq 11,27$ *, il nous suffit de vérifier que* 127 *n'est divisible par aucun des nombres* $2, 3, 5, 7$ *et* 11*. Les caractères de divisibilité montrent que* 127 *n'est pas divisible par*2 *ou par*3 *ou par* 5*. Pour*7 *et* 11 *on effectue les divisions euclidiennes :*

1. $127 = 187 + 1$*, le reste de la division de* 127 *par* 7 *est* 1*.* 127 *n'est donc pas divisible par* 7*.*

2. $127 = 11 \times 11 + 6$*, le reste de la division de* 127 *par* 11 *est* 6*.* 127 *n'est donc pas divisible par* 11*.*

*On en conclut que* 127 *est un nombre premier.*

**Théorème 1.2.3** ***Théorème fondamental de l'arithmétique*** *Tout entier* $n \geq 2$ *s'écrit de manière unique* $n = p_1^{\alpha_1} \cdots p_r^{\alpha_r}$ *où* $p_1 < p_2 < \cdots < p_r$ *sont des nombres premiers et* $\alpha_1, \ldots, \alpha_k$ *sont dans* $\mathbb{N}^*$*. On dit que* $n = p_1^{\alpha_1} \cdots p_r^{\alpha_r}$ *est la décomposition en produit de facteurs premiers de* $n$*.*

**Corollaire 1.2.2** *Soient* $a$ *et* $b$ *deux entiers naturels non nuls et* $p$ *un nombre premier. Si* $p$ *divise* $ab$*, alors* $p$ *divise* $a$ *ou* $p$ *divise* $b$*.*

**Démonstration 1.2.4** *Démonstration par disjonction des cas*

1. *Si* $p$ *divise* $a$*, alors la propriété est vraie*
2. *Si* $p$ *ne divise pas* $a$*, alors* $p$ *est premier avec* $a$ *puisque* $p$ *admet pour seuls diviseurs 1 et* $p$*, et d'après le théorème de Gauss* $p$ *divise* $b$*.*

**Remarque 1.2.1** *On note :*

$$\mathcal{P} = \{1, 3, 5, 7, 11, \ldots\}, \text{ (l'ensemble des nombres premiers).}$$

**Definition 1.2.2** *On définit* $\mathcal{P}$ *l'ensemble des nombres entiers positifs premiers. Pour* $x \geq 1$ *réel, soit* $\pi$ *l'application défine de* $\mathbb{R}$ *dans* $\mathbb{N}$ *par :*

$$\forall x \in \mathbb{R},\ \pi(x) := Card\{p \in \mathcal{P} : 1 \leq p \leq x\}.$$

**Théorème 1.2.4**

$$\pi(x) \sim \frac{x}{logx} [\sim\text{: signifie Asymptotiquement lorsque } x \to \infty].$$

**Historique** ;
- Conjecturé par Gauss en 1792.
-Tchebychev a montré en en 1851 que si x est assez grand on a :

$$0,92\tfrac{x}{logx} \leq \pi(x) \leq 1,11\tfrac{x}{logx}$$

-Démontré indépendamment par Hadamard et La Vallée Poussin en 1896 à l'aide de méthodes d'analyse complexe.

en 1977 Euler avait introduit la fonction zéta

**Definition 1.2.3** *la fonction zéta de Riemann :*

$$\zeta(s) = \sum_{n=1}^{\infty} \frac{1}{n^s} \ tq : s \in \mathbb{R}$$

**Théorème 1.2.5 (Euler)**

$$\zeta(s) = \sum_{n=1}^{\infty} \frac{1}{n^s} = \prod_{p \in \mathcal{P}} \frac{1}{1 - \frac{1}{p^s}}, \ s > 1.$$

le produit est pris sur tous les nombres premiers.

**Démonstration 1.2.5** *La factorisation unique des entiers $n \geq 1$, et la convergence justifient le calcul formel :*

$$\begin{aligned} \zeta(s) &= \sum_{r_2, r_3, r_5, \ldots \geq 0} \frac{1}{(2^{r_2} \times 3^{r_3} \times 5^{r_5} \ldots)^s} \\ &= \prod_{p \in \mathcal{P}} (\sum_{r \geq 0} \frac{1}{p^{rs}}) \\ &= \sum_{n=1}^{\infty} \frac{1}{n^s} = \prod_{p \in \mathcal{P}} \frac{1}{1 - \frac{1}{p^s}}, \ s > 1. \end{aligned}$$

**Definition 1.2.4 (Symbole de Legendre)** *Soit $p$ un nombre premier et $a$ un entier naturel premier avec $p$. Le symbole $(\frac{-}{p})$ est défini par : $(\frac{a}{p}) = 1$ si l'équation $x^2 \equiv a \mod p$ possède une solution dans $\mathbb{N}$. Dans le cas contraire on a : $(\frac{a}{p}) = -1$ autrement dit :*

$$(\frac{a}{p}) = \begin{cases} 1 \ si \ x^2 \equiv a \mod p \ possède \ une \ solution \ dans \ \mathbb{N} \\ \\ -1 \ si x^2 \equiv a \mod p \ ne \ possède \ pas \ de \ solutions \ dans \ \mathbb{N} \end{cases}$$

**Exemple 1.2.2** $(\frac{1}{p}) = 1, \ (\frac{m^2}{p}) = 1, \ (\frac{2}{11}) = -1, (\frac{5}{11}) = 1$

**Remarque 1.2.2** *On prolonge le symbole $(\frac{-}{p})$ par zéro sur $\mathbb{N}$. Ce symbole a été défini par Legendre*[1] *pour les nombres premiers impairs et par Kronecker*[2] *pour le nombre 2*

1. Adrien-Marie Legendre, né le 18 septembre 1752 à Paris et mort le 9 janvier 1833 dans la même ville, est un mathématicien français.

2. Leopold Kronecker (7 décembre 1823 - 29 décembre 1891) est un mathématicien et logicien allemand. Persuadé que l'arithmétique et l'analyse doivent être fondées sur les « nombres entiers », il est célèbre pour la citation suivante : « Dieu a fait les nombres entiers, tout le reste est l'œuvre de l'Homme

**Remarque 1.2.3** *[4] Le symbole de Legendre vérifie de plus :*

1. $(\frac{a}{p}) \equiv a^{\frac{(p-1)}{2}} \mod p$, *pour* $a \in \mathbb{N}$.
2. $(\frac{ab}{p}) = (\frac{a}{p})(\frac{b}{p})$ *pour* $a$ *et* $b$ *quelconques dans* $\mathbb{N}$.

**Proposition 1.2.1** *On dit que* $x \in \mathbb{Z}$ *est* ***résidu quadratique*** *modulo* $n$ *s'il existe* $y \in \mathbb{Z}$ *tel que* $y^2 \equiv x\ [n]$. *Si* $p$ *est un entier premier,*
*En utilisant les résidus quadratiques On définit le symbole de Legendre* $\left(\frac{x}{p}\right)$ *par*

$$\left(\frac{x}{p}\right) = \begin{cases} 0 \\ \quad si\ x\ est\ divisible\ par\ p \\ 1 \\ \quad si\ x\ n'est\ pas\ divisible\ par\ :p\ et\ x\ est\ un\ résidu\ quadratique\ modulo\ p \\ -1 \\ \quad si\ x\ n'est\ pas\ un\ résidu\ quadratique\ modulo\ p. \end{cases}$$

**Definition 1.2.5 (indicatrice d'Euler)** *La fonction indicatrice d'Euler,* $\phi$, *est définie pour tout* $n \in \mathbb{N}^\star$ *par*

$$\phi(n) = card\{1 \leq k \leq n;\ k\ est\ premier\ avec\, n\}.$$

*c-à-d :* $\phi$ *est la fonction de l'ensemble des entiers strictement positifs dans lui-même, qui à* $n$ *associe le nombre d'entiers positifs inférieurs à* $n$ *et premiers avec* $n$.

**Exemple 1.2.3** *Par exemple,* $\phi(8) = 4$ *car les quatres nombres* $1, 3, 5$ *et* $7$ *sont premiers avec* $8$. *Par ailleurs* $\phi(1) = 1$ *c'est le seul nombre qui est égal à son indicatrice d'Euler.*

**Théorème 1.2.6 (Restes chinois)** *soient* $m_1$, ..., $m_n$ *des entiers* $\geq 2$ *deux à deux premiers entre eux, et* $a_1, \ldots, a_n$ *des entiers. Le système de congruence :*

$$\begin{cases} x & \equiv & a_1 & \mod m_1 \\ \vdots & \vdots & \vdots & \\ x & \equiv & a_n & \mod m_n \end{cases}$$

*admet exactement* ***une unique solution*** *modulo le produit* $M = m_1 \times \cdots \times m_n$ *donnée par la formule :*

$$x = a_1 M_1 y_1 + \cdots + a_n M_n y_N$$

*où* $M_i = M/m_i$ *et* $y_i \equiv M_i^{-1} \mod m_i$ *pour* $i$ *compris entre* $1$ *et* $n$.

**Exemple 1.2.4** *[2] : Une bande de* $17$ *pirates s'est emparée d'un butin composé de pièces d'or d'égale valeur. Ils décident de se les partager également, et de donner le reste au cuisinier chinois. Celui-ci recevrait alors* $3$ *pièces. Mais les pirates se querellent, et six d'entre eux sont tués. Le*

*cuisinier recevrait alors* 4 *pièces. Dans un naufrage ultérieur, seuls le butin, six pirates et le cuisinier sont sauvés, et le partage donnerait alors* 5 *pièces d'or à ce dernier. Quelle est la fortune minimale que peut espérer le cuisinier quand il décide d'empoisonner le reste des pirates ?*

*Si* $x$ *est ce nombre,* $x$ *est le plus petit entier positif tel que :*

$$\begin{cases} x & \equiv & 3 & \mod 17 \\ x & \equiv & 4 & \mod 11 \\ x & \equiv & 5 & \mod 6 \end{cases}$$

*On applique le théorème chinois : on a* $M = 17 \times 11 \times 6 = 1122$*,* $M_1 = 66$*,* $M_2 = 102$*,* $M_3 = 187$*. L'inversion de chaque* $M_i$ *modulo* $m_i$ *(par l'algorithme d'Euclide) donne* $y_1 = 8$*,* $y_2 = 4$*,* $y_3 = 1$*.*

*On obtient donc :* $x \equiv 3 \times 66 \times 8 + 4 \times 102 \times 4 + 5 \times 187 \times 1 \mod 1122 \equiv 785 \mod 1122$*. Le gain minimal est de 785 pièces d'or, voila qui est particulièrement motivant !*

## 1.3 Test de primalité

Il est très important de savoir si un nombre donné est premier ou pas ; et sinon de détérminer sa décomposition en produit d'éléments premiers. Si on arrive à déterminer la primalité d'un entier dans un temps donné, alors sa factorisation peut prendre énormément plus de temps. On peut toujours essayer de chercher parmi les nombres inférieurs à ce nombre ses diviseurs, essayer la méthode du crible d'Eratosthène, le test de Fermat, celui d'Euler ou bien d'autres. Cependant ces algorithmes restent incapables de déterminer, en un temps raisonnable, la factorisation d'un grand nombre

### 1.3.1 Division et crible d'eratosthène[4]

Pour tester la primalité d'un entier il suffit de parcourir tous les entiers entre 2 et $n-1$, et tester si ces entiers divisent n ou non. Bien sûr, il est facile d'améliorer cet algorithme : si n n'est pas premier, l'un de ces diviseurs est plus petit que $\sqrt{n}$ il suffit de tester les entiers entre 2 et $\sqrt{n}$
Dans le même ordre d'idées, citons le crible d'Erathostène, qui permet de mettre la main sur tous les premiers entre 2 et $n$.
Á titre d'exemple pour déterminer tous les entiers premiers plus petits que 100, on procède comme suit :
on écrit tous les entiers qui vont de 2 à 100 (rappelons que 1 n'est pas premier). Le premier entier écrit est 2. Il est premier : on l'entoure, et on barre tous ses multiples. Le premier entier non barré après 2 est 3 : il est premier, et on barre tous ses multiples. Le premier entier non barré après 3 est 5 : il est premier et on barre tous ses multiples. Et on procède comme ceci jusqu'à épuiser tous les entiers.... Ceux qui ne sont pas barrés sont exactement les premiers !
Voici un exemple pour déterminer tous les premiers de 1 à 40 :

| | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|
| 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 |
| 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 |

TABLE 1.1: étape 1

| | 2 | 3 | | 5 | | 7 | | 9 | |
|---|---|---|---|---|---|---|---|---|---|
| 11 | | 13 | | 15 | | 17 | | 19 | |
| 21 | | 23 | | 25 | | 27 | | 29 | |
| 31 | | 33 | | 35 | | 37 | | 39 | |

TABLE 1.2: étape 2

| | 2 | 3 | | 5 | | 7 | | | |
|---|---|---|---|---|---|---|---|---|---|
| 11 | | 13 | | | | 17 | | 19 | |
| | | 23 | | 25 | | | | 29 | |
| 31 | | | | 35 | | 37 | | | |

TABLE 1.3: étape 3

| | 2 | 3 | | 5 | | 7 | | | |
|---|---|---|---|---|---|---|---|---|---|
| 11 | | 13 | | | | 17 | | 19 | |
| | | 23 | | | | | | 29 | |
| 31 | | | | | | 37 | | | |

TABLE 1.4: étape 4

Les divisions et le crible d'Erathotène sont assez efficaces pour de petits entiers. Mais dès que ces entiers dépassent 50 chiffres, ils deviennent inutilisables ; ainsi il faut totalement changer de méthode.

### 1.3.2 Test de Fermat

En rapport avec son efficacité, c'est le test de primalité, i.e. le critère de composition, le plus simple. Il repose directement sur le petit théorème de Fermat

**Théorème 1.3.1 (petit théorème de Fermat)** *Soit p un nombre premier et a un entier positif inférieur strictement à p, alors on a :* $a^{p-1} \equiv 1 \mod p$.

**Démonstration 1.3.1** *parmi les démonstrations on choisi la démonstration par récurrence : Soit p un nombre premier et P(a) la propriété :*$a^p \equiv a \mod p$

**Initialisation** : *Pour* $a = 0$, $0^p = 0 \equiv 0 \mod p$

**Hérédité** : *On suppose P(a) est vraie, c'est à dire* $a^p \equiv a^{p-1} \mod p$ *et on montre que* $(a+1)^p \equiv (a+1) \mod p$. *En appliquant P(a) on obtient :* $(a+1)^p \equiv (a+1) \mod p$.

**Conclusion** : *Pour tout* $a \in \mathbb{N}$ *: P(a) est vraie.*

**Definition 1.3.1** *Soit p un entier* $\geq 2$. *On appelle témoin de Fermat pour a, tout entier a premier avec p,* $a^{p-1} \not\equiv 1 \mod p$

On prend un entier a au hasard et on calcule $a^{p-1} \mod p$ si $[a^{p-1} \not\equiv 1 \mod p\ ]$ alors $p$ n'et pas premier

**Remarque 1.3.1** *Le petit théorème de Fermat peut servir de test de primarité (certains disent de nos jours primalité, le franglais ça fait tendance...), mais ce n'est pas un critère. Il n'est qu'une condition nécessaire pour tout candidat au statut de nombre premier.*

**Exemple 1.3.1** *On dit que* 341 *est un entier naturel pseudo-premier.*

### 1.3.3 Test d'Euler

**Definition 1.3.2** ***Indicatrice d'Euler*** *: Soit n un entier positif supérieur à 2, a un entier premier avec n et* $\phi(n)$ *l'ordre du groupe* $\mathbb{Z}_n^\star$ *; alors on a :*

$$\phi(n) = (p_1 - 1)p_1^{r_1 - 1}...(p_s - 1)p_s^{r_s - 1} \text{ si } n = p_1^{r_1}...p_s^{r_s} \text{ et } a^{\phi(n)} \equiv 1 \mod n$$

**Théorème 1.3.2** *Soit n un entier sans facteurs carrés et a un entier positif inférieur strictement à n ; alors on a :*

$$\forall k \in \mathbb{Z} : \ a^{k\phi(n)+1} \equiv a \mod n.$$

**Definition 1.3.3** *:Soit $n$ un entier impair $\geq 1$. On appelle témoin d'Euler pour $n$, tout entier $a$ tel que :*

$$1 < a < n,\ pgcd(a,n) = 1 \text{ et } (\frac{a}{n}) \not\equiv a^{\frac{n-1}{2}} \mod n$$

On prend un entier a au hasard, et on calcule $a^{\frac{n-1}{2}} \mod n$.
si $a^{\frac{n-1}{2}} \not\equiv (\frac{a}{n}) \mod n$ alors $n$ n'est pas premier.

**Théorème 1.3.3 (Solovay [3] et Strassen [4])** *Soit $n$ un entier impair $\geq 3$ tel que l'on ait :* $(\frac{a}{n}) \equiv a^{\frac{n-1}{2}} \mod n$ *pour tout entier $a$ premier avec $n$ alors, $n$ est premier.*

**Démonstration 1.3.2** *[5]*

### 1.3.4 Test de primalité de Lehmer

Grâce au test de Fermat, d'autre variété de tests a été mise au point. Dans ce test, on suppose donnée une décomposition en facteurs premiers de $p-1$
**Critère de Lehmer** [5] :
Soit $n$ un entier impair $\geq 3$. Les conditions suivantes sont équivalentes :

1. $n$ est premier.
2. Il existe un entier $a$ tel que :

$$a^{n-1} \equiv 1 \mod n \text{ et } a^{\frac{n-1}{q}} \not\equiv 1 \mod n, \text{pour tout diviseur premier } q \text{ de } n-1$$

**Démonstration 1.3.3** *[5]*

**Exemple 1.3.2** *Ce critère, utilise avec $a = 5$, permet de démontrer que $3 \times 2^{3189} + 1$ est premier. Cet entier possède neuf cent soixante et un chiffres décimaux*

**Corollaire 1.3.1** *Soit $n$ un entier impair $\geq 3$. Les conditions suivantes sont équivalentes :*

1. *$n$ est premier.*
2. *Il existe un entier $a$ tel que : $a^{\frac{n-1}{2}} \equiv -1 \mod n$ et $a^{\frac{n-1}{q}} \not\equiv 1 \mod n$ pour tout diviseur premier impair $q$ de $n-1$*

**Proposition 1.3.1** *Soit $n$ un entier $\geq 2$. Supposons que pour tout diviseur premier $q$ de $n-1$, il existe un entier $a$, qui dépend de $q$, tel que l'on ait :*
$a^{n-1} \equiv 1 \mod n$ *et* $a^{\frac{n-1}{q}} \not\equiv 1 \mod n$
*Alors, $n$ est premier.*

5. Derrick Henry Lehmer est un mathématicien américain, spécialiste de théorie des nombres connu pour ses tests de primalité, né le 23 février 1905 à Berkeley (Californie) où il est mort 22 mai 1991. Il a aussi posé |e problème qui porte son nom (Problème de Lehmer) : si $n \equiv 1 \mod \phi(n)$, n est-il nécessairement premier ?

**Exemple 1.3.3** *On peut démontrer que :* $3 \times 2^{2816} + 1$ *est premier,en utilisant ce critère avec les couples (q ; a) = (2 ; 7) et (3 ; 2), ou bien la proposition, avec a = 13.*

**Proposition 1.3.2** ***Test de Pepin***[6] *Soit n un entier* $\geq 1$ *et* $F_n = 2^{2^n} + 1$ *On a l'équivalence :*

$$F_n \text{ est premier} \iff 3^{\frac{F_n-1}{2}} \equiv -1 \mod F_n$$

### 1.3.5 Test de Lucas-Lehmer

Soit $n \in \mathbb{N}^*$. Le nombre de Mersenne[7] d'indice $n$ est le nombre $M_n = 2^n - 1$. Généralement, ces nombres sont considérés lorsque $n$ est un entier premier afin de tenter de fabriquer de grands nombres premiers. En effet, si $M_n$ est premier, alors $n$ est premier. Réciproquement, il est faux que $n$ premier entraîne $M_n$ premier et d'ailleurs on ne sait même pas si c'est vrai pour une infinité de nombres premiers.
Pour tester si $M_p$ est premier, le test de Lucas[8]-Lehmer :

**Théorème 1.3.4 (critère de Lucas-Lehmer)** *Soit p un nombre premier impair. Soit* $(S_n)_{n\geq 0}$ *la suite définie par* $S_0 = 4$ *et pour tout* $n \geq 0$, $S_{n+1} = S_n^2 - 2$. *Alors* $M_p$ *est premier si et seulement si* $M_p | S_{p-2}$.

**Exemple 1.3.4** $M_3 = 7$ *est premier, en utilisant le test de Luca-Lehmer. On a* $s_0 = 4$, *on calcule :* $s_3 - 2 = s_1 = 422 = 140 \mod 7$

6. Né à Cluses, en Haute-Savoie, il devient jésuite en 1846 et, de 1850 à 1856 puis de 1862 à 1871, il est professeur de mathématiques dans divers collèges jésuites. Il est nommé professeur de droit canonique en 1873. Il est mort à Lyon à l'âge de 77 ans. Ses travaux mathématiques sont centrés sur la théorie des nombres.

7. Marin Mersenne (1588-1648), connu également sous son patronyme latinisé Marinus Mersenius, est un religieux français de l'ordre des Minimes, érudit, physicien, mathématicien et philosophe

8. François Édouard Anatole Lucas (1842-1891) est un mathématicien français.
Biographie

CHAPITRE 2

# La cryptographie

## 2.1 Introduction à la cryptographie

### 2.1.1 Définition de la cryptographie et de son importance dans la sécurité des communications

**c'est quoi la cryptographie ?**

La cryptographie est l'art de chiffrer, coder les messages au plus précis (l'art et la science d'obscurcir les messages afin que personne, sauf le destinataire prévu, puisse les lire). qui est devenue aujourd'hui une science à part entière.
Au croisement des mathématiques, de l'informatique, et parfois même de la physique, elle permet ce dont les civilisations ont besoin depuis qu'elles existent : le maintien du secret afin d' éviter une guerre, protéger un peuple, il est parfois nécessaire de cacher des choses.

**L'usage de la cryptographie[3]**

La cryptographie est traditionnellement utilisée pour dissimuler des messages aux yeux de certains utilisateurs. Cette utilisation a aujourd'hui un intérêt d'autant plus grand que les communications via internet circulent dans des infrastructures dont on ne peut garantir la fiabilité et la confidentialité. Désormais, la cryptographie sert non seulement à préserver la confidentialité des données mais aussi à garantir leur intégrité et leur authenticité.

1. La confidentialité : consiste à rendre l'information intelligible ( clair, compris) à d'autres personnes que les acteurs de la transaction.
2. L'intégrité : vérifier l'intégrité des données consiste à déterminer si les données n'ont pas été altérées ( modifiées) durant la communication.
3. L'authentification : consiste à assurer l'identité d'un utilisateur, c.-à-d de garantir à chacun des correspondants que son partenaire est bien celui qu'il croit être un contrôle d'accès peut permettre l'accès à des ressources uniquement aux personnes autorisées.
4. La non répudiation de l'information est la garantie qu'aucun des correspondants ne pourra nier la transaction.

### 2.1.2 Concepts de base

Les concepts de base de la cryptographie comprennent :

(1) **Chiffrement** : C'est le processus de conversion des données en un format illisible appelé « texte chiffré ». Le chiffrement utilise une clé de chiffrement pour effectuer des transformations mathématiques sur les données d'origine, les rendant ainsi inintelligibles pour quiconque ne possède pas la clé appropriée.

(2) **Clé de chiffrement** : Une clé de chiffrement est un paramètre utilisé par l'algorithme de chiffrement pour effectuer des opérations de chiffrement et de déchiffrement.

(3) **Signature numérique** : Une signature numérique est un mécanisme utilisé pour garantir l'authenticité et l'intégrité d'un message ou d'un document électronique. Elle utilise une clé privée pour générer une empreinte numérique unique du message, qui peut être vérifiée à l'aide de la clé publique correspondante. Si la vérification de la signature est réussie, cela garantit que le message n'a pas été modifié depuis sa signature et que l'expéditeur est authentique.

(4) **Hachage** : Le hachage est une fonction mathématique qui prend en entrée des données de taille variable et produit une sortie de taille fixe appelée « haché » ou « empreinte ». Cette empreinte est généralement utilisée pour vérifier l'intégrité des données. Même une petite modification des données d'origine produira une empreinte totalement différente. Les fonctions de hachage sont utilisées dans de nombreux protocoles cryptographiques, tels que les signatures numériques et les certificats numériques.

### 2.1.3 Historique de la cryptographie, en mettant en évidence les premières méthodes utilisées dans l'Antiquité

Depuis des temps très reculés dans l'histoire, les messages secrets étaient utilisés pour plusieurs raisons et surtout pour des raisons diplomatiques ou militaires. Ces messages secrets sont des messages qu'on écrit tout d'abord d'une façon naturelle,

**Scytale**

Sparte vers -450 AJC, (principe des codes de permutation)

**Code de Jules César**
vers -50 AJC, (principe des codes de substitution, $n = n + 3$) cryptanalysé par les arabes (9e),développé (ajout de blancs, mauvaise orthographe, et qui a coûté la vie à Marie Stuart (fin 16e)
**Code de Vigenére 1586**
Premier chiffre polyalphabétique, invulnérable à l'analyse statistique avec un nombre immense de clef, il resta négligé pendant deux siècles lui préférant des chiffres de substitution homophonique : l'exemple le plus remarquable est le grand chiffre de Louis XIV (17e) déchiffré seulement à la fin du 19e.
**Codes à répertoires**
Très anciens, utilisés intensivement jusqu'au d´début du 20-ème siècle.
**Enigma : début XXe**
Utilisée par l'armée allemande durant la seconde guerre mondiale, décryptée par les polonais grâce a une répétition récurrence, puis par Alan Turing via la recherche de mots probables.

### 2.1.4 Principe de Kerckhoffs

Jusqu'au milieu du XXe siècle, la sécurité d'un chiffre reposait sur le secret de son fonctionnement. Le problème est que dés que ce secret est éventé, il faut changer entièrement le cryptosystème ce qui est complexe et coûteux.
**Principe de Kerckhoffs** [1] :
La sécurité d'un cryptosystème ne repose pas sur le secret du cryptosystème mais seulement sur la clef du cryptosystème qui est un paramètre facile à transmettre secrètement et 'a changer, de taille réduite (actuellement de 64 à 2048 bits).

1. Système doit être indéchiffrable.
2. La force ne doit pas résider dans l'algorithme de chiffrement (ou la machine)
3. La clé doit être simple à mémoriser, sans notes écrites, et facile à changer.
4. Le système doit être portatif avec un seul opérateur.
5. D'usage facile (pas de stress).
6. Applicable au télégraphe.

**codes modernes** :
On peut distinguer deux grandes familles de codes classiques :

1. Codes à clefs secrètes :qui mêlent codes de permutation et codes de substitution. Les exemples les plus célèbres sont DES(Data Encryption Standard), et AES(Advanced Encryption Standard).

1. Kerckhoffs est né à Nuth aux Pays-Bas et fut baptisé Jean-Guillaume-Hubert-Victor-François-Alexandre-Auguste Kerckhoffs von Nieuwenhoff. Il raccourcit son nom par la suite et entama des études à l'université de Liège, où il obtient le grade de Docteur en Lettres. Après une période où il enseigna en France et dans les Pays-Bas, il devint professeur d'allemand à l'École des Hautes Études Commerciales et à l'École Arago.

2. codes à clefs publiques :qui reposent sur la notion mathématique de fonctions ‘a sens unique. Citons dans cette catégorie les codes RSA et El Gamal.

Suivant le principe de Kerckhoffs, ces cryptosystèmes sont connus de tous, leur sécurité reposant sur l'existence de clés au coeur du chiffre.

### 2.1.5 Histoire de la cryptographie pendant les 2 guerres mondiales

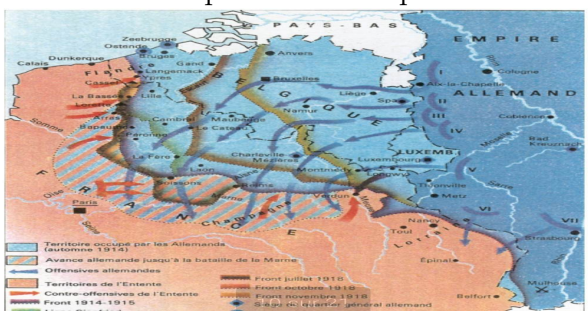
**première guerre mondiale**

La cryptographie devient une arme. Peut-on faire quelque chose ?

1. On transmet en clair (armée russe) : progrès ? (ou sous le stress)
2. Analyse de trafic (doigté de l'opérateur) (expéditeur/ destinataire/ date/longueur/préambule)
3. Gestion des clés 1914 : changement trimestriel 1918 : quotidien

**Exemples (cryptanalyse, ses succès)** :

1. Nov. 1916, Arthur Zimmermann, ministre des affaires étrangères 9 janvier, réunion au château de Pless. Guerre navale totale dès le 1 février 1917, mais il faudrait éviter l'entrée en guerre des Etats-Unis qui vit sous la présidence Wilson.



[8]

2. w. Wilson : (1916) ( Nous ne sommes pas en guerre,grâce à moi).


"All the News That's Fit to Print."

The New York Times.

THE WEATHER

NEW YORK, FRIDAY, NOVEMBER 10, 1916.—TWENTY-TWO PAGES.

ONE CENT

WITH 272 ELECTORAL VOTES, WILSON WINS; GETS CALIFORNIA, NORTH DAKOTA, NEW MEXICO

GERMANY FAVORS LEAGUE OF PEACE WITH CONDITION.

ASQUITH EXPOUNDS ESSENTIALS OF P...

WON'T CONCEDE WILSON VICTORY WITHOUT RECOUNT

Both Sides Should Be Anxious to Remove All Doubt. Wilson Says.

FOUR STATES IN QUESTION

M'CORMICK JOYFUL

WON'T CONCEDE WILSON VICTORY WIHOUT RECOUNT

SHOW SUMMARY

WILSON AHEAD IN CALIFORNIA BY OVER 3,100

Republican Chairman Concedes State on Face of the Returns.

30 PRECINCTS ARE MISSING

FULL RECOUNT IS CERTAIN

THE ELECTORAL VOTE.

WILSON HUGHES

CALIFORNIA DECIDES

Result Was in Doubt Till Republicans Gave It Up at 11:25 P. M.

MINNESOTA STILL CLOSE

Hughes's Lead Going Up There; Wilson Gains in Other Close States.

CONTESTS ARE PROMISED

Charges of Fraud Are Already Under Investigation in Some Districts.

ONE CITY PLAYS BIG PART

Hughes's Lead Is Up in Minnesota; Wilson Holds N... Dakota, New Mexico

[8]

3. Le télégramme intercepté est-il authentique ? La réponse arrive le 2 mars 1917 Le 2 avril, la déclaration de guerre est adopté par le Congrès

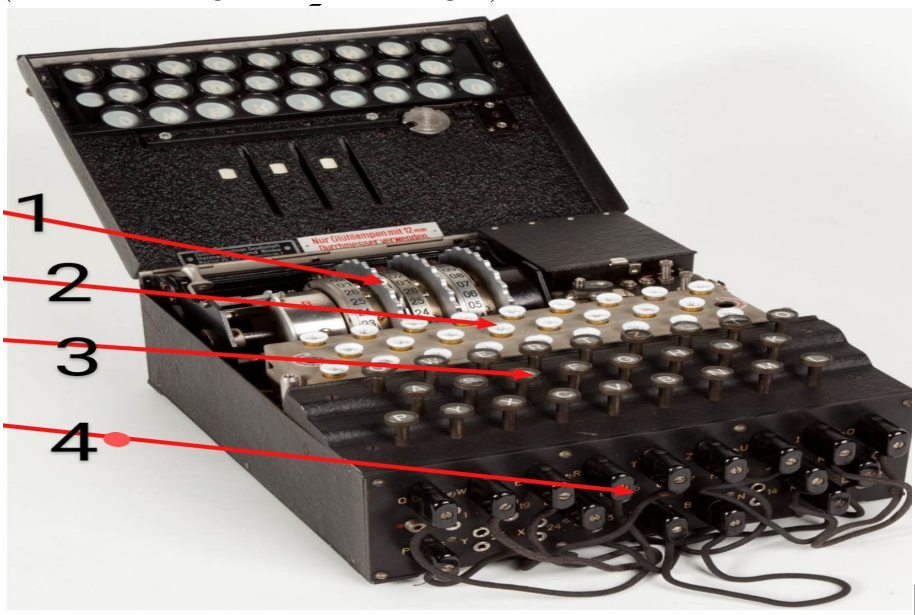

[8]

**enigma**

Machine de chiffrement des Allemands pour les relations diplomatiques puis pour l'armée. scherbius (fondé en 1918, armée : 1925) (similaire aux Etats-Unis, Hollande, Angleterre)

- Mécanique (vitesse)
- Changement facile de clés
- Chiffrement par cascades de substitutions

(casser toute régularité de la langue) • Confiance absolue en son inviolabilité.



[8]

**les composants**

- Rois rotors (substitution,26 lettres )(1)
- panneaux lumineux(2)
- Clavier(3)
- Panneau de connections frontal(4)

Il est prévu que la sécurité du système de cryptage soit préservée même si l'ennemi a une machine à sa disposition
Changement quotidien des clés sur une machine Enigma

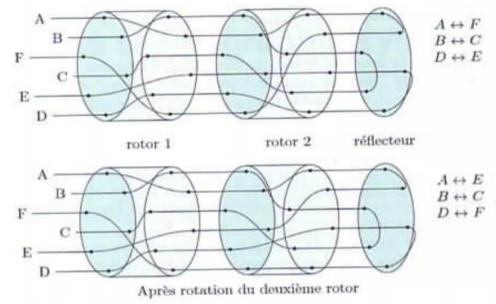
- Connections avant : A-L, P-R, T-D, B-W, K-F, O-y
- Brouilleur : $2 - 3 - 1$
- Orientation du brouilleur : Q - C - W

Nombre de clés :
$26 \times 26 \times 26 \times 6 \times 100391791500 = 10^{16} = 10000000000000000$


[8]

Et durant le même jour, clé de session (une clé par message) :
changement de position de l'orientation du brouilleur
PGHPGH -> KIVBJE

## Enigma (1925 - 1945 ...)




[8]

### Á l'attaque d'enigma

Trahison de Schmidt (8 nov. 1931) vente de documents à l'agent français Rex
France renonce ...
Pologne : Marian Rejewski (1905-1980)

- Construire une réplique de la machine (en partie depuis la machine commerciale)
- Déchiffrement via l'émission de la clé de session en double (PGHPGH -> KIVBJE)

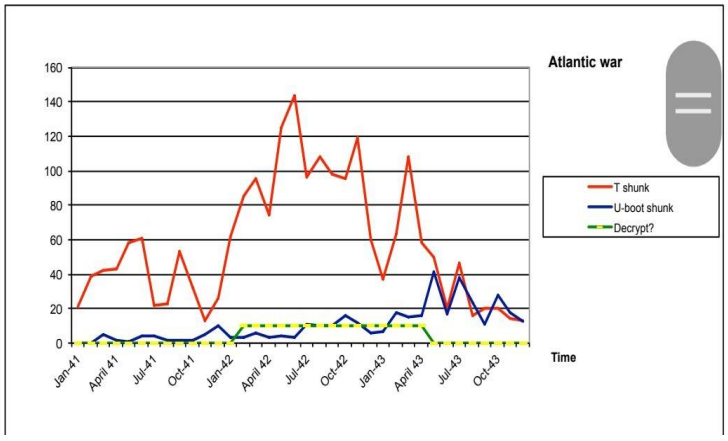
Lien entre P -> K et P -> B (+ 3 mouvements)
24 juillet 1939 : les Polonais donnent une machine Enigma aux Français et Anglais
1 septembre 1939 : début de la 2e guerre mondiale Room 40 -> Bletchley : Plus de ressources

1. Les trois lettres clef ne sont pas toujours aléatoires (clavier)
2. Le rotor ne peut pas être à la même place deux jours de suite
3. Connections : pas entre lettres consécutives (5-> T)

**déchiffrement** :
Déchiffrer Enigma Succès si l'on peut déchiffrer ...

Déchiffrer Enigma

Succès si l'on peut déchiffrer...




[8]

## 2.2 Types de cryptographie

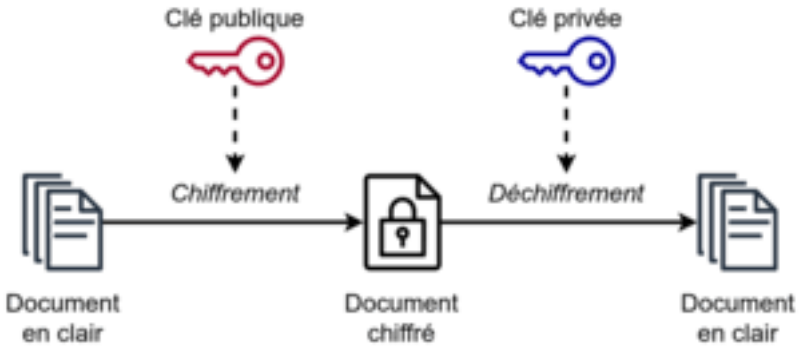
### 2.2.1 Clé secrète[7]

La cryptographie à clé secrète ou cryptographie symétrique c'est la plus ancienne, consiste à utiliser la même clé et le même algorithme pour le chiffrement et le déchiffrement entre deux personnes en communication qui sont les seuls à connaître cette. Le chiffrement symétrique se divise en deux parties : chiffrement par bloc (block ciphers) et chiffrement par flot (stream ciphers) .

1. Chiffrement par flot : dans un crypto système par flots, le cryptage des messages se fait caractère par caractère ou bit par bit, au moyen de substitutions générées aléatoirement, la taille de la clé est donc égale à la taille du message .
2. Chiffrement par bloc : dans un algorithme de chiffrement par bloc, chaque message clair est découpé en blocs de taille fixe de même longueur et chiffré à l'aide d'une clé unique. Ces algorithmes sont en général construits sur un modèle itératif. Il utilise une fonction F qui prend une clé secrète k et un message M de n bits. La fonction F est itérée un certain nombre de fois (nombre de tours). Lors de chaque tour, la clé k est différente et on chiffre le message qui vient d'être obtenu de l'itération précédente. Les différentes clés k(i) qui sont utilisées sont déduites de la clé secrète k.


### 2.2.2 clé publique[6]

La cryptographie à clé publique (asymétrique) consiste en l'existence d'une paire de clés de chaque côté (émetteur et récepteur) liées mathématiquement. avec les algorithmes asymétriques, les clés de chiffrement et de déchiffrement sont distinctes et ne peuvent se déduire l'une de l'autre. On peut donc rendre l'une des deux publique tandis que l'autre reste privée. C'est pourquoi on parle de chiffrement à clef publique. Si la clé publique sert au chiffrement, tout le monde peut chiffrer un message, que seul le propriétaire de clé privé pourra déchiffrer. On assure ainsi la confidentialité. Certains algorithmes permettent d'utiliser la clé privée pour chiffrer. Dans ce cas, n'importe qui pourra déchiffrer, mais seul le possesseur de clé privée peut chiffrer.




Ce système a deux utilisations majeures

1. la confidentialité des messages reçus : c'est celle qu'on vient de décrire, l'expéditeur utilise la clé publique du destinataire pour chiffrer son message. Le destinataire utilise sa clé privée pour déchiffrer le message de l'expéditeur, garantissant la confidentialité du contenu ;
2. l'authentification de l'expéditeur d'un message (pas nécessairement confidentiel) : l'expéditeur utilise sa clé privée pour chiffrer un message que n'importe qui peut déchiffrer avec la clé publique de l'expéditeur, ce qui garantit que le message a été chiffré par l'expéditeur, seul à posséder la clé privée ; c'est le mécanisme utilisé par la signature numérique pour authentifier l'auteur d'un message .

## 2.3 Cryptographie moderne

### 2.3.1 Chiffrement par décalage

**Definition 2.3.1** *Pour* $1 \leq k \leq 26$ *on définie :*
$e_k(x) = (X + k) \mod 26$
$d_k(y) = (y - k) \mod 26$
$(x, y \in \mathbb{Z}_{26})$

**Remarque 2.3.1** *Pour une cas particulier* $k = 3$ *le cryptosystème est souvent appelé le décalage de César .*

**Exemple 2.3.1** *On définit une bijection :*
$f : \{A, B, C, ..., Z\} \longrightarrow \{0, 1, 2, ..., 25\}$

*par :*
$A \leftrightarrow 1$ , $B \leftrightarrow 2$ ,..., $Z \leftrightarrow 25$.
*Ainsi par exemple :"A L A A" devient "0 11 0 0".*
***Petite exemple très pratique*** *:supposant que la clé pour le chiffrement de décalage est* $K = 11$*, supposant que le texte en clair est :*

*SALUTBADRI*

*Premièrement on convertit le message avec la correspondance indiquée : 18 0 11 20 19 1 0 3 17 8 après on ajoute 11 :*
*3 11 22 5 4 12 11 14 2 19*
*Enfin, on convertit vers les caractères alphabétiques :*

*DLWFEMLOCT*

### 2.3.2 Chiffrement par substitution

**Definition 2.3.2** *K consiste tous les possibilité de permutation de 26 symboles* $0, .., 25$ *pour chaque permutation* $\pi \in K$ *on définie :*
$e_\pi(x) = \pi(x)$
*Et on définie :*
$d_\pi(y) = \pi^{-1}(y)$
*où* $\pi^{-1}$ *est la permutation inverse de* $\pi$ .

**Exemple 2.3.2** *Voici un exemple d'équation aléatoire qui pourrait comprendre la fonction de chiffrement*

| $a$ | $b$ | $c$ | $d$ | $e$ | $f$ | $g$ | $h$ | $i$ | $j$ | $k$ | $l$ | $m$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $X$ | $N$ | $Y$ | $A$ | $H$ | $P$ | $O$ | $G$ | $Z$ | $Q$ | $W$ | $B$ | $T$ |

| $n$ | $o$ | $p$ | $q$ | $r$ | $s$ | $t$ | $u$ | $v$ | $w$ | $x$ | $y$ | $z$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $s$ | $F$ | $L$ | $R$ | $C$ | $V$ | $M$ | $U$ | $E$ | $K$ | $J$ | $D$ | $I$ |

*ainsi :*
$e_\pi(a) = X$ *et* $e_\pi(b) = N$ *etc„,.*
*La fonction de déchiffrement est la permutation inverse ,ceci est formé en écrivant la deuxième ligne en premier,puis tri par ordre alphabétique, on obtient :*

| $A$ | $B$ | $C$ | $D$ | $E$ | $F$ | $G$ | $H$ | $I$ | $J$ | $K$ | $L$ | $M$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $d$ | $l$ | $r$ | $y$ | $v$ | $o$ | $h$ | $e$ | $z$ | $x$ | $w$ | $p$ | $t$ |

| $N$ | $O$ | $P$ | $Q$ | $R$ | $S$ | $T$ | $U$ | $V$ | $W$ | $X$ | $Y$ | $Z$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $b$ | $g$ | $f$ | $j$ | $q$ | $n$ | $m$ | $u$ | $s$ | $K$ | $a$ | $c$ | $i$ |

*ainsi :*
$e_\pi(A) = d$ *et* $e_\pi(B) = l$ *etc„,.*

### 2.3.3 Chiffrement affine

**Definition 2.3.3** *:*
*On met :*
$K = \{(a, b) \in \mathbb{Z}_{26} \times \mathbb{Z}_{26} : pgcd(a, 26) = 1\}$
*pour* $k = (a, b) \in K$ *on définie :*
$e_k(x) = (ax + b) \mod 26$
$d_k(y) = a^{-1}(y - b) \mod 26$
$(x, y \in \mathbb{Z}_{26})$

**Exemple 2.3.3** *Supposant que k=(7,3),notons que :*$7^{-1} \mod 26 = 15$ *la fonction de chiffrement est :*$e_k(x) = 7x + 3$
*et la fonction de déchiffrement est :*
$d_k(y) = 15(y - 3) = 15y - 19$
*après on trouve :*
$d_k(e_k(x)) = d_k(7x + 3)$
*=15(7x+3)-19*
*=x+19-19*
$=x$
*Pour illustrer on chiffre le mot "bad".*
*Premièrement on convertit les lettres modulo* 26*.*
*On trouve :1 0 3 , ainsi on chiffre :* $7 \times 1 + 3 \mod 26 = 10$
$7 \times 0 + 3 \mod 26 = 3$
$7 \times 3 + 3 \mod 26 = 24$
*On obtient : 10 3 24 En alphabétique : KDY*

### 2.3.4 Chiffrement de Viginére

Dans le chiffrent par décalage et le chiffrement par substituions ,une fois qu'une clé est choisie ,chaque caractère alphabétique est mappé sur un caractére alphabétique unique,pour cette raison,ces cryptosysteme manoalphabétique , nous présentons maintenant un cryptosysteme qui n'est pas manoalphabétique, le bien connu chiffrement de Véginère .
En utilisant la correspondance $A \leftrightarrow 1$ , $B \leftrightarrow 2$ ,..., $Z \leftrightarrow 25$.décrire précisément ,on peut associe a chaque k une chaîne alphabétique de longueur m, appelée mot-clé . le chiffrement viginére crypte m charactère alphabétique a la fois :chaque element de tect en claire equivant m caractères alphabétique.

**Definition 2.3.4** *Soit m un entier positif ,on définie* $K = (\mathbb{Z}_{26})^m$ *,pour une clé :* $k = (k_1, ..., k_m)$*,on définie :*
$e_k(x_1, ..., x_m) = (x_1 + k_1, ..., x_m + k_m)$
*et*

$d_k(y_1, ..., y_m) = (y_1 + k_1, ..., y_m + k_m)$
*où tous les opérations sont exécutés dans* $\mathbb{Z}_{26}$

**Exemple 2.3.4** *[9] Supposant que* $m = 6$ *et le mot clé est CYPHER, cela correspond à l'équivalent numérique K=(2,8,15,7,4,17), supposant que le texte en clair est :*

*thecryptosystemisnotsecure*

*On convertit les élément du texte en clair modulo 26, puis les écrivant par groupes de six, puis ajoutons le mot clé modulo 26, comme suit :*

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 19 | 7 | 8 | 18 | 2 | 17 | 24 | 15 | 19 | 14 | 18 | 24 |
| 2 | 8 | 15 | 7 | 4 | 17 | 2 | 8 | 15 | 7 | 4 | 17 |
| 21 | 15 | 23 | 25 | 6 | 8 | 0 | 23 | 8 | 21 | 22 | 15 |

| | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 18 | 19 | 4 | 12 | 8 | 18 | 13 | 14 | 19 | 18 | 4 | 2 | 20 | 17 | 4 |
| 2 | 8 | 15 | 7 | 4 | 17 | 2 | 8 | 15 | 7 | 4 | 17 | 2 | 8 | 15 |
| 20 | 1 | 19 | 19 | 12 | 9 | 15 | 22 | 8 | 25 | 8 | 19 | 22 | 25 | 19 |

*L'equivalent alphabetique de la chaine de texte chiffré serait donc :*

*VPXZGIAXIVWPUBTTMJPWIZITWZT*

*Pour déchiffrer on peut utilise le même mot-clé ,mais on le soustrairons modulo 26 du texte chiffré au lieu d'ajouter*

### 2.3.5 Chiffrement de Hill

**Definition 2.3.5** *soit* $m \geq 3$*un entier , soit K=(*$m \times m$ *matrices inversibles sur* $\mathbb{Z}_{26}$*)*
*pour une clé k, on définie :*

$$e_k(x) = xk \text{ et } d_k(y) = yk^{-1}$$

*Où tous les opérations sont exécutés dans* $\mathbb{Z}_{26}$

**Exemple 2.3.5** *[9] Supposant que la clé est :* $K = \begin{pmatrix} 11 & 8 \\ 3 & 7 \end{pmatrix}$ *D'après les calculs ,on a :*

$K^{-1} = \begin{pmatrix} 7 & 18 \\ 23 & 11 \end{pmatrix}$
*Supposant qu'on veut chiffrer le texte :july,on a deux éléments de texte à chiffrer (19,20) [ju] et (11,24) [ly], on calcule comme suit :*
$(9, 20)\begin{pmatrix} 11 & 8 \\ 3 & 7 \end{pmatrix} = (99 + 60, 72 + 140) = (159, 112) = (3, 4)$

*et*
$(11,24)\begin{pmatrix}11 & 8\\ 3 & 7\end{pmatrix} = (121+72, 88+68) = (11,21)$
*Ainsi ,le cryptage de july est DELW,pour décrypter, on calcule :*
$(3,4)\begin{pmatrix}7 & 18\\ 23 & 11\end{pmatrix} = (9,20)$
*et*
$(11,20)\begin{pmatrix}7 & 18\\ 23 & 11\end{pmatrix} = (11,24)$
*Par conséquent ,le texte correct est obtenu .*

**Remarque 2.3.2** *Á ce point, on a montré que le déchiffrement est possible si k a un inverse, en effet pour le déchiffrement soit possible ,*

## 2.4 RSA

**Vue ensemble** :

1. Le système RSA, du nom de ses concepteurs Rivest, Shamir et Adleman est le premier système de chiffrement à clé publique robuste à avoir été inventé (en 1977).
2. Il est toujours largement utilisé lorsque l'on veut échanger des données de manière sécurisée sur Internet.
3. Le chiffrement RSA est asymétrique : il utilise une paire de clés, une clé publique pour le chiffrement et une clé privée pour le déchiffrement. Le système RSA permet également de signer des données. Celles-ci sont signées avec la clé privée et tout possesseur de la clé publique peut vérifier la signature.
4. La robustesse du système RSA repose sur le fait que l'on ne sait pas avec les moyens et savoir actuel, obtenir la clé privée à partir de la simple connaissance de la clé publique.

### 2.4.1 Principe de RSA

Le principe de l'algorithme RSA est le suivant :
Nous commençons par choisir deux grands nombres premiers p et q.

Calculons $n = p \times q$.

Nous cherchons ensuite e, un nombre entier aléatoire qui est premier avec $(p-1)(q-1)$.

Le couple (n, e) consitue la clé publique.

En utilisant l'algorithme d'Euclide, on calcule d, tel que $=e \times d = 1(\mod (p-1)*(q-1))$.
Le couple (n, d) consitue la clé privée.

Aussi bien le texte décrypté m que le texte crypté c doivent être des entiers positifs.

Puis, avant de crypter le message m, nous nous assurons que $0 \leq m < n$.

Si m est plus grand que le modulo n, le résultat c ne sera pas lié à m de façon unique.

Nous savons grâce au théorème d'Euler que pour tous les entiers m,$m^{(e\times d)} = m(\mod n)$.

Ainsi, avec $0 \leq m < n, m^{(e\times d)}(\mod n) = m$.

Pour crypter le message m, nous réalisons l'algorithme suivant :

$Ek(m) = m^e(\mod n) = c$

avec Ek( ) pour l'algorithme de cryptage.

Pour décrypter c avec la clé privée d, nous réalisons l'algorithme suivant :

$Dk(c) = c^d(\mod n) = m^{(e\times d)}(\mod n) = m^1(\mod n) = m$

avec Dk( ) pour l'algorithme de décryptage.

La paire (e, n) constitue la clé publique du cryptosystème RSA.
Tout le monde peut utiliser cette paire (e, n) pour crypter un message.

**Exemples d'utilisation RSA** :

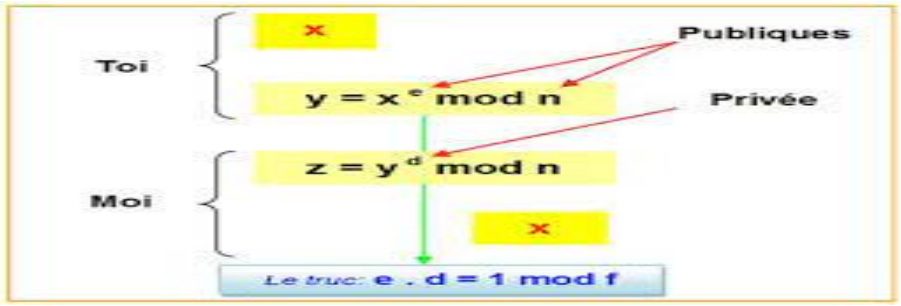
**Exemples d'utilisation RSA** :
Soit Deux petits premiers :$p = 5$ et $q = 7$

- $n = 5 \times 7 = 35$,$\phi(n) = (5-1)(7-1) = 24$
- e et d : $ed \equiv 1 \mod 24$
- $ed = 1$ : Non, trop petit
- $ed = 25$ : Ok, mais $e = d = 5$ et alors clé privé = clé publique
- $ed = 49$ : Pareil, e = d
- $ed = 73$ : 73 est premier, raté
- $ed = 97$ : 97 est premier, raté
- $ed = 121$ : 11 au carré, encore raté

- $ed = 165 : 165 = 5 \times 33$, et 5 est premier : Ok
- Clé publique $= (35, 5)$ • Clé privée $= (35, 33)$

**Bases**

1. Vous voulez me transmettre la valeur x.
2. Je vous transmets sans précaution la clé publique (e et n).
3. Vous vous en servez pour crypter le message.
4. Vous me le transmettez : y
5. Je suis le seul à pouvoir décrypter car j'ai la clé privée.




**Explication** :

1. x est la valeur que vous voulez coder ( découpez le message de façon que $x \leq n$).
2. e et $n$ sont les deux éléments de la clé publique.
3. Vous enfouissez $x$ dans un calcul qui donne y.
4. y est la valeur transmise.
5. On calcule z en utilisant ma clé privé d.
6. Après ce tour de passe-passe mathématique, il se trouve que z est égal à x.

### 2.4.2 Codage et décodage d'une message

**Explication** :
On se donne le message suivant : (*Attaquez*)
Je doit envoyer secrètement ce message à une personne X. Cette personne doit avoir une clé publique qui n'est rien d'autre que deux entiers $n_X$ et $s_X$ vérifiant les conditions suivantes :
*i*. $n_X = pq$ où $p$ et $q$ sont des nombres premiers.
*ii*. $p$ et $q$ sont gardés secrets par chacun.
*iii*. L'entier $s_x$ est premier avec l'entier $(p-1)(q-1)$.

**Transformation d'un message**

| | | | |
|---|---|---|---|
| A = 01 | K = 11 | U = 21 | 1 = 31 |
| B = 02 | L = 12 | V = 22 | A = 01 |
| B = 02 | L = 12 | V = 22 | 2 = 32 |
| C = 03 | M = 13 | W = 23 | 3 = 33 |
| D = 04 | N = 14 | X = 24 | 4 = 34 |
| E = 05 | O = 15 | Y = 25 | 5 = 35 |
| F = 06 | P = 16 | Z = 26 | 6 = 36 |
| G = 07 | Q = 17 | , = 27 | 7 = 37 |
| H = 08 | R = 18 | . = 28 | 8 = 38 |
| I = 09 | S = 19 | ? = 29 | 9 = 39 |
| J = 10 | T = 20 | 0 = 30 | ! = 40 |

**Remarque 2.4.1** *Pour désigner un vide entre deux mots on écrit le nombre* 00. *Ainsi notre message devient un nombre M : M* = 0120200117210526

**codage du message**

**Exemple** On coupe M en morceaux plus petits que $n_x$.
$n_x = 37 \times 41 = 1517$; $M = 0120200117210526 = \underbrace{01}_{M_1}\underbrace{20}_{M_2}\underbrace{02}_{M_3}\underbrace{11}_{M_4}\underbrace{72}_{M_5}\underbrace{10526}$
On travaille, dans la suite, successivement avec chaque morceau $M_1..M_5$. Le message codé devient $\overline{M}$ : $\underbrace{...}_{\overline{M_1}}\underbrace{...}_{\overline{M_2}}\underbrace{...}_{\overline{M_3}}\underbrace{...}_{\overline{M_4}}\underbrace{...}_{\overline{M_5}}$ où $\overline{M}$ est le reste de la division de $(M_i)^{s_x}$ par $n_x$ pour $i = 1, ..., 5$.[4]

**Décodage du message**

Le destinataire reçoit le message codé $\overline{M}$ comme il connaît la décomposition $n_x = pq$ et on sait que $s_x$ est premier avec $(p-1)(q-1)$ ; alors il existe un entier $t_x$ tel que : $1 \leq t_X < (p-1)(q-1)$ et $s_x t_x \equiv 1 \mod (p-1)(q-1)$ Le destinataire peut donc facilement calculer l'entier $t_X$. Personne d'autre ne peut le calculer tant que la décomposition de $n_x$ reste secrète, Pour décoder le message on calcule le reste de la division de $(\overline{M})^{t_x}$ par $n_x$ pour $i = 1, ..., 5$. Ce reste n'est rien d'autre que l'entier $M_i$ pour $i = 1, ..., 5$. Ainsi le message décodé est bien : $M = 0120200117210526 = \underbrace{01}_{M_1}\underbrace{20}_{M_2}\underbrace{01}_{M_3}\underbrace{17}_{M_4}\underbrace{2105}_{M_5}\underbrace{26}$

**Remarque**[4]

1. comme $s_x t_x \equiv 1 \mod (p-1)(q-1)$ ; alors il existe un entier k tel que $s_x t_x = 1 + k(p-1)(q-1) = 1 + \phi(nX)$.
2. L'entier $t_X$ est une clé secrète.

**Signature du message**

Une personne est identifiée par sa clé publique, et elle est parfaitement identifiée par sa clé publique et sa signature que seul lui peut la signer. Donc un message, pour plus de sécurité, doit être signé. On va décrire, comment on signe un message crypté en RSA. J'ai envoyé un message M à une personne X, que j'ai transformé à l'aide des entiers $n_X$ et $s_X$. La personne $X$ va décoder le message avec sa clé secrète $t_X$. Mais qui prouve que c'est bien moi qui a envoyé ce message ; ma clé publique est publique et n'importe qui peut l'utiliser ! Donc je doit ajouter ma signature à ce message.
Moi aussi, Badri, j'ai une clé publique $(n_A; s_A)$ et une clé secrète $t_A$. J'ajoute au message M ma signature $M^{t_A}$. Pour que X s'assure que c'est bien moi qui a envoyé le message M, il calcule : $(M^{t_A})^{s_A} \mod n_A$ ;
S'il trouve M, alors c'est bien moi. Sinon, c'est que le message ne vient pas de moi.
**Conseil d'utilisation RSA** :
Il y a de nombreuses manières de mal utiliser RSA et d'ouvrir des failles de sécurité !

1. Ne jamais utiliser de valeur n trop petite.
2. Ne jamais utiliser d'exposant e trop petit.
3. N'utiliser que des clés fortes.
4. ($p-1 et q-1$ ont un grand facteur premier).
5. Ne pas chiffrer de blocs trop courts.
6. Ne pas utiliser de n communs à plusieurs clés

CHAPITRE 3

# Programme C++ pour implémenter les algorithmes :tests de primalité et RSA

## 3.1 Tests de primalité

### 3.1.1 Test d'Euler

Un programme C++ simple pour vérifier si la racine carrée d'un nombre [selon la méthode d'euler]

```
#include<iostream>
using namespace std;

// (renvoie vrai si la racine carrée de n dans modulo p existe)
bool squareRootExists(int n, int p)
{
n = n%p;
// One by one check all numbers from 2 to p − 1
(Vérifiez un par un tous les nombres de 2 à p − 1)
for (int x=2; x<p; x++)
if ((x*x)%p == n)
return true;
```

return false ;
}

// Driver program to test
(programme pilote à tester) int main()
{
int $p = 7$;
int $n = 2$;
squareRootExists(n, p) ? cout « "Yes" : cout « "No" ;
return 0 ;
}
après la compilation on obtient : (Output :

Yes)

d'où le racine carrée existe

### 3.1.2 Test de Miller Rabin

on met l'algorithme suivant pour tester la primalité selon Miller Rabin

// C++ program Miller-Rabin primality test
#include <bits/stdc++.h>
using namespace std ;

// Utility function to do modular exponentiation.(Fonction utilitaire pour faire une exponentiation modulaire)
// It returns $(x^y)$ % p
int power(int x, unsigned int y, int p)
{ int $res = 1$ ; // Initialize result x = x % p ; // Update x if it is more than or // equal to p(Mettre à jour x s'il est supérieur ou égal à p)
while (y > 0)
{ // If y is odd, multiply x with result(Si y est impair, multiplier x par le résultat)
if ($y1$)
res = (res*x) % p ;
// y must be even now
$y = y >> 1$ ; // $y = y/2$ $x = (x * x)$ % p ; } return res ;
}

// This function is called for all k trials. It returns(Cette fonction est appelée pour tous les k essais. Il revient)
// false if n is composite and returns true if n is (si n composé) // probably prime.(probablemnt premier)
// d is an odd number such that d*2 = n-1 (d est un nombre impair)
// for some $r >= 1$(pour certaint r)
bool miillerTest(int d, int n)
{ // Pick a random number in [2..n-2](on prendre un nombre aléatoire)
// Corner cases make sure that $n > 4$
int a = 2 + rand()
// Compute $a^d$ % n
int $x = power(a, d, n)$;

if $(x == 1 || x == n - 1)$
return true;

// Keep squaring x while one of the following doesn't(Continuez à élever x au carré tant que l'un des éléments suivants ne le fait pas)
// happen
(arriver)
// (i) d does not reach $n - 1$
(d n'atteint pas) // (ii) $(x^2)$ % n is not 1
// (iii) $(x^2)$ % n is not $n - 1$
while $(d! = n - 1)$
{
$x = (x * x)$ % n;
$d/ = 2$;

if $(x == 1)$ return false;
if $(x == n - 1)$ return true;
}

// Return composite
return false;
}

// It returns false if n is composite and returns true if n
(Il renvoie faux si n est composé et renvoie vrai si n)
. // is probably prime. k is an input parameter that determines
( est probablement premier. k est un paramètre d'entrée qui détermine)

```
// accuracy level. Higher value of k indicates more accuracy.
(niveau de précision. Une valeur plus élevée de k indique une plus grande précision)
bool isPrime(int n, int k)
{
// Corner cases
if (n <= 1||n == 4) return false;
if (n <= 3) return true;

    // Find r such that n = 2^d * r + 1 for some r >= 1
int d = n - 1; while (d%2 == 0) d/ = 2;
    // Iterate given number of 'k' times
for (int i = 0; i < k; i++)


if ( !miillerTest(d, n))

return false;

    return true;
}

    // Driver program
int main()

int k = 4; // Number of iterations

    cout « "All primes smaller than 100 : |n";
for (int n = 1; n < 100; n++)
if (isPrime(n, k))
cout « n « " ";

    return 0;
}
```

Après on compile on obtient :
All primes smaller than 100 : (tout les nombres premiers plus petit que 100)
2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97

### 3.1.3 Test de Pepin

Algorithme selon Pepin :

```
#include <iostream>
#include <cmath>

using namespace std;

long double pepinTest(int n)
{ long double Fn = pow(2, pow(2, n)) + 1;
long double p = sqrt(pow(3, (Fn − 1))) + 1;
long double result = p / Fn;

return ceil(result) - result;
}
int main()
{
long double pepvalue;

for (int i = 1; i <= 15; i + +)
{
pepvalue = pepinTest(i);

if(isfinite(pepvalue))
if( pepvalue == 0) cout << "F" << i << "prime" << endl;
else cout << "F" << i << "composite" << endl;
}

return 0;
}
```

on compile on obtient :
F1 prime
F2 prime
F3 prime
d'où f1 f2 et f3 sont premiers

### 3.1.4 Test de Lucas Lehmer

En mathématiques, le test de Lucas-Lehmer est un test de primalité pour les nombres de Mersenne (se forme $2^n - 1$) dans ce algorithme (language c++) on fais par exemple les i jusqu'a 4

```
#include <stdio.h>
```

```
#include <math.h> // pow(x, exp)

    //————————————————————————————-

    char isMersenneLucasLehmer(unsigned int prime)
{
unsigned int i, termN = 4;
unsigned long mersenne;
unsigned int limit;
int res;

    mersenne = (unsigned long) pow(2, (double)prime) - 1;
if(prime%2 == 0)
{
returnprime == 2;
}
else
{
res = (int) sqrt((double) prime);
for(i = 3; i <= res; i+ = 2)
{
if(prime%i == 0)
{
return 0;
}
}

    limit = prime - 2;
for (i = 1; i <= limit; ++i)
{
termN = (termN * termN - 2)%mersenne;
}
}
returntermN == 0;
}
//————————————————————————————-

    /*
Function : findMersenneLucasLehmer()
```

```
    */
void findMersenneLucasLehmer(unsigned int limit)
{
unsigned int i, current = 0;
unsigned long mersenne, bitsInLong = 64;
(Cela ne fonctionne que jusqu'à n = 64)
    for(i = 2; i <= bitsInLong; i++)
{
if(current >= limit)
{
break;
}

    if (isMersenneLucasLehmer(i))
{
mersenne = (unsignedlong)pow(2, (double)i) - 1;
printf("current = %lu, mersenne = %lu, index = %u|n", current, mersenne, i);
++current;
}
}
}
//-------------------------------------------

    int main()
{
unsigned int limit = 8;
findMersenneLucasLehmer(limit);
return 0;
}
```

après la compilation on trouve :
current = 0, mersenne = 3, index = 2
current = 1, mersenne = 7, index = 3
current = 2, mersenne = 31, index = 5
current = 3, mersenne = 127, index = 7
current = 4, mersenne = 8191, index = 13

## 3.2 Programme de cryptage RSA

Algorithme de cryptage RSA se déroule par les étapes suivantes :
begin

1. Choisissez deux nombres premiers p et q.
2. Calculez n = p*q.
3. Calculez phi = (p-1) * (q-1).
4. Choisir un entier e tel que 1 < e < phi(n) et pgcd(e, phi(n)) = 1 ; c'est-à-dire que e et phi (n) sont premiers entre eux.
5. Calculer d comme $d \equiv e1(\mod phi(n))$ ; ici, d est l'inverse multiplicatif modulaire de e modulo phi(n).
6. Pour le chiffrement, c = me mod n, où m = message d'origine.
7. Pour le déchiffrement, m = c d mod n.

end
**Exemple**
on fais un algorithme avec p=17 et q=23.

```
#include <iostream>
#include <math.h>
using namespace std;

    // trouver pgcd
int gcd(int a, int b)
{
int t;
while (1)
{
t = a % b;
if (t == 0)
return b;
a = b;
b = t;
}
}

    int main()
{
// 2 nombres premiers aléatoires
```

double $p = 17$ ;
double $q = 23$ ;
double $n = p * q$ ; //calculer n
double track ;
double phi = (p - 1) * (q - 1) ; //calculer phi

//clé public
//e signifie chiffrer (en anglais encrypt)
double $e = 7$ ;
//pour vérifier que $1 < e < phi(n)$et $pgcd(e, phi(n)) = 1$ ; c'est-à-dire que e et phi (n) sont premiers entre eux.
while $(e < phi)$
{
track = gcd(e, phi) ;
if $(track == 1)$
break ;
else
e++ ;
}

//clé privée //d signifie déchiffrer (en anglais decrypt)
//choisir d tel qu'il vérifie d*e = 1 mod phi
double $d1 = 1/e$ ;
double $d = fmod(d1, phi)$ ;
double message = 99 ;
double c = pow(message, e) ; //chiffrer le message
double m = pow(c, d) ;

$c = fmod(c, n)$ ;
$m = fmod(m, n)$ ;

cout « "Original Message = " « message ;
cout « "|n"
« "p = " « p ;
cout « "|n"
« "q = " « q ;
cout « "|n"
« "n = pq = " « n ;
cout « "|n"
« "phi = " « phi ;

```
cout « "|n"
« "e = " « e;
cout « "|n"
« "d = " « d;
cout « "|n"
« "Encrypted message = " « c;
cout « "|n"
« "Decrypted message = " « m;

    return 0;
}
```

Après la compilation on trouve :
Original Message = 99
$p = 17$
$q = 23$
$n = pq = 391$
$phi = 352$
$e = 7$
d = 0.142857
Encrypted message = 74
Decrypted message = 99

CHAPITRE 4

# Conclusion

En conclusion, ce mémoire nous a permis de plonger dans les profondeurs de la théorie des nombres et de la cryptographie, et de comprendre l'importance cruciale de ces deux domaines dans la sécurité de l'information. Nous avons exploré les concepts fondamentaux de la théorie des nombres tels que les nombres premiers, les congruences et les résidus quadratiques, et nous avons découvert comment ces concepts sont utilisés pour concevoir des systèmes de chiffrement solides.

La cryptographie moderne repose sur des algorithmes sophistiqués, tels que le chiffrement à clé publique RSA et les systèmes de chiffrement á base de courbes elliptiques, qui tirent leur force de la théorie des nombres. Nous avons également examiné des protocoles avancés, tels que Diffie-Hellman et les protocoles de preuve de connaissance zéro, qui ouvrent de nouvelles possibilités en matière de sécurité de l'information.

Ce mémoire nous a permis de comprendre l'intérêret t de la théorie des nombres sur notre vie quotidienne, en soulignant son rôle essentiel dans la protection des communications, des transactions et des données personnelles. La cryptographie, en tant qu'outil puissant et indispensable, repose sur ces fondements mathématiques solides pour garantir la confidentialité et l'intégrité des informations sensibles.

Il est important de continuer à explorer et à approfondir ces domaines, car la sécurité de l'information continue d'évoluer face aux nouvelles menaces et aux avancées technologiques. Les chercheurs et les professionnels de la cryptographie doivent rester à la pointe des développements, en s'appuyant sur la théorie des nombres pour concevoir de nouvelles méthodes de cryptographie robustes et résistantes aux attaques. Nous espérons que ce mémoire vous a permis d'appréhender les bases de la théorie des nombres et de la cryptographie, et de comprendre leur rôle central dans la sécurisation des échanges d'informations. Que vous choisissiez de poursuivre vos études dans ce

domaine ou simplement d'approfondir vos connaissances, nous vous encourageons à continuer à explorer les merveilles de la théorie des nombres et à contribuer à la sécurité de l'information dans notre société de plus en plus connectée.

Mots clés : théorie des nombres, cryptographie, sécurité de l'information, chiffrement, nombres premiers, congruences, résidus quadratiques, chiffrement à clé publique, courbes elliptiques, protocoles de preuve de connaissance zéro.

# Bibliographie


[1] TM Apostol. Introduction to analytic number theory springer. *New York*, 1976.

[2] Renaud Dumont. Cryptographie et sécurité informatique. *cours provisoires, Université de Liège Faculté des Sciences Appliquées*, 2010, 2009.

[3] Zemor G. *Cours de cryptographie*. 2000.

[4] PARTIE II. Universite mohamed premier faculte des sciences departement de mathematiques et informatique oujda.

[5] Alain Kraus. Cours de cryptographie mm029-2009/10.

[6] Reguidel M. *Quelques rappels sur les techniques cryptographiques*. 2002.

[7] Videau M. Critère de sécurité des algorithmes de chiffrement à clé secrète , thèse de doctorat. *Université de Paris 6,(france)*, 2005.

[8] Jackes Savoy. Histoire de la cryptographie de la première guerre mondiale à internet.

[9] Douglas Robert Stinson and Maura Paterson. *Cryptography : theory and practice*. CRC press, 2018.