RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE

MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE Dr. TAHAR MOULAY SAIDA

FACULTE : TECHNOLOGIE

DEPARTEMENT : INFORMATIQUE

MÉMOIRE DE MASTER

Option : sécurité informatique et cryptographie

Étude comparative de systèmes cryptographiques

Présenté par :

Belarbi abd elkader

Benaida Ahmed

Encadré par :

Dr.Ahmed chaouki lokbani

Année Universitaire 2020-2021

ABSTRACT

Encryption is the process of changing the form of information to another form using encryption algorithms that require the presence of certain values, and these values are the key used in the encryption process, the final output of the encryption process is unreadable text. The time required for encryption and the memory space used are two essential things : these algorithms take a long time to encode or decrypt, and they differ in the amount of memory consumed. From this point of view, we looked for the ideal algorithm, among the four algorithms, Caesar, Wegner, and RSA..., which consumes little memory in more than a little time, and is more secure.

Le cryptage est le processus consistant à changer la forme des informations en une autre forme à l'aide d'algorithmes de cryptage qui nécessitent la présence de certaines valeurs, et ces valeurs sont la clé utilisée dans le processus de cryptage, la sortie finale du processus de cryptage est un texte illisible. Le temps nécessaire au chiffrement et l'espace mémoire utilisé sont deux choses essentielles : ces algorithmes sont longs à encoder ou à déchiffrer, et ils diffèrent par la quantité de mémoire consommée. De ce point de vue, nous avons recherché l'algorithme idéal, parmi les quatre algorithmes,seasar et végnere et AES et RSA, qui consomme peu de mémoire en plus d'un peu de temps, et est plus sécurisé.

.

التشفير هو عملية تغيير شكل المعلومات إلى نموذج آخر باستخدام خوارزميات التشفير التي تتطلب وجود قيم معينة ، وهذه القيم هي المفتاح المستخدم في عملية التشفير ، والمخرج النهائي لعملية التشفير هو نص غير قابل للقراءة. الوقت اللازم للتشفير ومقدار الذاكرة المستخدمة هما شيئان رئيسيان: تستغرق هذه الخوارزميات وقتاًا طويلاً للتشفير أو فك التشفير ، وتختلف في مقدار الذاكرة المستهلكة. من وجهة النظر هذه ، بحثنا عن الخوارزمية المثالية ، من بين الخوارزميات الأربع ، "seasar vegnere AES RSA"، والتي تستهلك ذاكرة أكثر بقليل من القليل من الوقت ، وهي أكثر أماناًا.

REMERCIEMENT

Avant toute chose, nous tenons à remercier Allah pour cette grâce d'être en vie et en bonne santé, et pour avoir terminé ce travail dans les meilleures conditions et ce malgré toutes les contraintes et les obstacles que nous avons rencontré.

Il est souvent difficile de remercier les gens qui vous aident à accomplir les tâches qui vous sont données, et pourtant nous nous devons exprimer l'entière gratitude que nous ressentons envers eux.

Nous tenons donc à présenter un remerciement bien distingué à notre encadrant Dr. ahmed chaouki lokbani pour son soutien, son aide, et ses conseils qui nous ont guidés durant l'élaboration de ce travail.

Nous portons toutes nos gratitudes aux enseignants de l'université de Moulay Tahar, pour leurs dévouements et leurs Assistantes tout au long de mes études universitaires.

Nous tenons aussi à exprimer nos gratitudes reconnaissance à tous les membres de jury d'avoir accepté et d'évaluer notre travail.

Enfin on remercie toutes personnes qui ont contribuées de près ou de loin à la réalisation de ce travail, ainsi qu'au bon déroulement du stage, et dont les noms ne figurent pas dans ce document

3DES : Triple DES
AES : Advanced Encryption Standard
DES : Data Encryption Standard
GSM : Global System for Mobile communications
ASCII : American Standard Code for Information Interchange
RSA :Ron Rivest, Adi Shamir, et Len Adlmen
IBM :International Business Machines
NSA :National Security Agency
RC4 :Rivest Cipher 4
XOR :Exclusive or,
NIST : National Institute of Standards and Technology
RC6 : Rivest cipher 6
AK :Add-RoundKey
MC :Mix-Columns
SB :Sub-Bytes
PGP : Pretty Good Privacy
S / MIME :Secure/Multipurpose Internet Mail Extensions
PGCD : Le plus grand dénominateur commun

TABLE DES MATIÈRES

abstract **3**

Table des matières **1**

Liste des tableaux **3**

Table des figures **4**

1 Introduction générale **1**

2 cryptographie classique **3**

2.1 Introduction 3

2.2 . Catégorie de cryptographie classique 3

2.2.1 Cryptographie par substitution 3

2.2.2 Substitution mono alphabétique 4

2.3 Le Chiffrement affine 4

2.3.1 Substitution poly alphabétique 5

2.3.2 Substitution par polygrammes 6

2.3.3 cryptographie par transposition 8

2.4 conclusion 10

cryptographie moderne **11**

2.5 Introduction 11

2.6 La cryptographie symétrique ou a clé secret 11

2.6.1 Chiffrement par block 12

2.6.2 Chiffrement à flot (stream cipher) : 12

2.7 L'algorithme DES (Data Ecryption Standard) 13

2.7.1 Etapes de l'algorithme DES 14

2.8 L'algorithme AES (cipher bloc Advenced Encryption Standard). 16

2.8.1 Fonctionnement d'AES 17

2.9 Cryptographie 3DES 19
2.9.1 Principe de chiffrement 3DES 19
2.10 Comparaison des trois chiffrements DES 3DES et AES 20
2.11 Conclusion 20
2.12 Chiffrement clé asymétrique 20
2.12.1 Chiffrements Diffie helman 21
2.12.2 Protocole de Diffie helman : 21
2.13 L'algorithme RSA (Ron Rivest, Adi Shamir, et Len Adlmen) 22
2.13.1 Principe de fonctionnement du RSA [34][35] 22

La Réalisation **25**
2.14 Introduction 25
2.15 L'environnement de développement 25
2.16 Qu'est-ce le langage Python en informatique ? 25
2.16.1 A quoi sert le langage Python ? 26
2.16.2 Qu'est-ce que Tkinter ? 26
2.17 jupyter notebook 27
2.17.1 Comment fonctionne un notebook Jupyter ? 28
2.18 Architecteur de application 29
2.19 Les algorithmes on pratique : 29
2.19.1 Les déffirants algorithme utilisée 29
2.20 Chiffrement de sesar 30
2.21 Chiffrement de vegner 30
2.22 L'algorithme AES 31
2.23 L'algorithme RSA 32
2.24 Comparaison et performance des tests des algorithmes 33
2.24.1 Le temps d'exécution 33
2.24.2 l'espace mémoire 39
2.25 conclusion 40
2.26 conclusion générale 41

Bibliographie **43**

LISTE DES TABLEAUX

TABLE **Page**

2.1 substitution mono alphabétique 4
2.2 alphabie affine 4
2.3 alphabie affine 4
2.5 déchiffrement affine 5
2.4 chiffrement affine 5
2.6 alphabie affine 7
2.7 alphabie affine 7
2.8 chéffrement hill 7
2.9 déchéffrement hill 8
2.10 dtransposition simple 9
2.11 transposition complexe par clonnes 9

2.12 temps d'éxicution de chiffrement AES vs RSA 33
2.13 temps d'éxicution de déchiffrement AES vs RSA 34
2.14 temps 2 d'éxicution de chiffrement AES vs RSA 34
2.15 temps 2 d'éxicution de déchiffrement AES vs RSA 35
2.16 temps 3 d'éxécution de chiffrement AES vs RSA 35
2.17 temps 3 d'éxécution de déchiffrementAES vs RSA 35
2.18 temps d'éxicution SESAR vs VEGNER 36
2.19 temps 2 d'éxicution SESAR vs VEGNER vs RSA 37
2.20 temps 3 d'éxicution SESAR vs VEGNER vs RSA 38
2.21 L'espace mémoir AES vs RSA 39
2.22 L'espace 2 mémoir AES vs RSA 39
2.23 L'espace mémoir SEASAR vs VEGNER 40
2.24 L'espace 2 mémoir SEASAR vs VEGNER 40

TABLE DES FIGURES

FIGURE **Page**

2.1 table de végner 6
2.2 fonction HILL "chiffrement" 7
2.3 fonction HILL (déchiffrement) 8

2.4 cryptographie symétrique 12
2.5 Principe du chiffrement par flot 13
2.6 représentation générale du DES 15
2.7 Le chiffrement AES. 16
2.8 les matrices 17
2.9 principe de chiffrement AES 18
2.10 3DES 19
2.11 chiffrement asymétrique 21

2.12 python 25
2.13 jupyter 27
2.14 architecteur générale de application 29
2.15 sesar chiffrement 30
2.16 Déchiffrement seasar 30
2.17 chiffrement végnere 31
2.18 déchiffrement végnere 31
2.19 Figure chiffrement TEXT.txt AES 32
2.20 déchiffrement RSA 33
2.21 Histogramme1 : Comparaison 1 AES et RSA 34
2.22 Histogramme2 : Comparaison 2 de déchiffrement AES et RSA 35
2.23 Histogramme3 : Comparaison 3 AES et RSA 36
2.24 Histogramme4 : Comparaison temps chiffrement et déchiffrement de César Vs Vigener 37

2.25 Histogramme5 : Comparaison temps chiffrement et déchiffrement de César Vs Vigener 38
2.26 Histogramme6 : Comparaison temps chiffrement et déchiffrement de César Vs Vigener 38

CHAPITRE 1

INTRODUCTION GÉNÉRALE

L e transfert croissant de données multimédias engendre des besoins en termes de sécurité d'informations. Pour répondre aux normes de confidentialité, plusieurs algorithmes de cryptage ont été développés ces dernières années. En raison de l'utilisation massive des données images dans divers domaines (médicale, industrielle, . . .) il est nécessaire d'utiliser et d'adapter ces techniques de sécurité à ce type de données.la cryptographie assurent les confidentialité des donnés Et leur l'intégrités. constituent le cœur de la sécurité permet des stocker les information sensibles ou de les faire véhiculer a travers un réseau peu sur de façon qu'elles ne soient intelligibles que a a un destinataire autorisé.[1] Malgré ses diverses formes, sans cesse en progrès, le fonctionnement de la cryptographie reste peu connu par le grand public mise à part quelques formes primitives englobant des algorithmes rudimentaires dans leur ensemble du fait qu'ils consistaient notamment au remplacement de caractères par d'autres. Quand aux applications modernes, avec usage des mathématiques, de la physique quantique, et des technologies les plus avancées, elles sont encore beaucoup moins connues. Ces dernières permettent de protéger le contenu d'un message en s'aidant le plus souvent de clés, et suivant que ces clés de chiffrement sont gardées secrètes ou pas, les algorithmes de chiffrement se scindent en deux principales catégories : les algorithmes à clé publique (RSA,. . .) et les algorithmes à clé privée (DES, 3DES, AES . . .). La cryptographie hybride consiste en une association des meilleures fonctionnalités des deux techniques de cryptage précédentes .Mais avec la puissance montante des ordinateurs, la cryptographie reste un domaine en plein mouvement pour suivre les nombreux progrès des cryptanalyses. Ainsi, certains algorithmes ont été remis en cause, tel que le DES qui a été remplacé par 3DES puis

récemment par AES. De plus, la sécurité de RSA est directement liée au progrès de la factorisation d'entiers qui peut être révélé un jour.[18] Ce rapport est structurer en trois chapitres : le premier chapitre donne un aperçu sur la cryptographie classique , le deuxième aborde le cryptographie moderne et quelques algorithmes de chiffrement basé sur le problème l'espace mémoire et le temps d'exécution le troisième chapitre , Vise à choisir l'algorithme idéal et quant 'à eux sont consacrés 'a l'implémentation de ces algorithmes. Enfin nous allons conclure par quelques perspectives concernant l'implémentation des algorithmes de chiffrement

CHAPITRE 2

CRYPTOGRAPHIE CLASSIQUE

2.1 Introduction

La cryptographie classique est l'étude des méthodes permettant de transmettre des données de manière confidentielle. Afin de protéger un message, L'apparition de la cryptographie remonte à la fin de l'Ancien Empire égyptien vers 2350avant Jésus Christ où l'on retrouve différentes techniques par permutation ou par substitution basées sur les hiéroglyphes égyptiens on lui applique une transformation dépendante d'un paramètre, appelé clé, afin de le rendre incompréhensible (chiffrement). Le déchiffrement est l'action inverse qui permet de reconstruire le texte en clair à partir du texte chiffré. [2] La cryptographie classique décrit la période avant les ordinateurs. Au Xème et VIIème siècle avant J.C., les Grecs ont utilisés une technique de chiffrement dite« par transposition » à des fins militaires. Cela consistait à effectuer des changements de positions de lettres dans le message.

2.2 . Catégorie de cryptographie classique

2.2.1 Cryptographie par substitution

Si bien que certains protocoles auparavant considérés comme sûrs sont devenus aujourd'hui triviaux à casser. Nous essayons ici de catégoriser les différents procédés cryptographiques historiques, dans l'objectif de montrer les avantages et failles de ceux-ci, et donc la difficulté de bâtir un algorithme suffisamment robuste. [3] Par un procédé de, chaque caractère est remplacé dans le cryptogramme par un autre caractère, mais

conserve sa place au sien de message Un des plus vieux exemples est le code de césar ou chaque lettre de l'alphabet (latin) est codée par celle figurant trois positions plus loin dans l'ordre l'alphabétique. (Figure 1) [4]

2.2.2 Substitution mono alphabétique

2.2.2.1 Substitution mono alphabétique de César

Jules César (100-44 avant J.C.) employait une substitution simple avec l'alphabet français. [5]
La plupart des méthodes utilisées durant les siècles suivants reposaient sur des substitutions mono-alphabétiques. Une substitution mono-alphabétique consiste à remplacer l'alphabet habituel par un autre alphabet, ou chaque caractère du texte en clair est remplacé par un caractère correspondant dans le texte chiffré (figure 1). [6]

K	A	C	I	M	I	A	Y	M	E	N
M	O	S	S	S	T	R	F	G	H	U

TABLE 2.1: substitution mono alphabétique

2.3 Le Chiffrement affine

Le chiffrement Affine est un type de substitution mono alphabétique, dans lequel chaque lettre d'un alphabet est mappée à son équivalent numérique, chiffré à l'aide d'une simple fonction, et reconverti en lettre. La formule utilisée signifie que chaque lettre crypté sur une autre lettre, et retour encore une fois, ce qui signifie que le chiffre est essentiellement une substitution standard chiffré avec une règle régissant quelle lettre va à quelle. Une Le chiffrement affine est créé en combinant le chiffrement additif et chiffrement multiplicatif comme suit

A	B	C	D	E	F	G	H	I	J	K	L	M	N
0	1	2	3	4	5	6	7	8	9	10	11	12	13

TABLE 2.2: alphabie affine

O	P	Q	R	S	T	U	V	W	X	Y	Z
14	15	16	17	18	19	20	21	22	23	24	25

TABLE 2.3: alphabie affine

C=(a*x.+b)
Ou a et b sont deux nombres et C est le message chiffré
Exemple :

Le message à chiffré est 'cipher', comme clé on prend (2 ,7) :
a- Chiffrement affine :
C=(2*2+7) mod26 =11 =l et c=(8*2+7)mod26=23=x
Le caractère c du message claire est chiffré en I et le i en X, ainssi de suite pour « p,h,e,r » en « L,L,P,P »

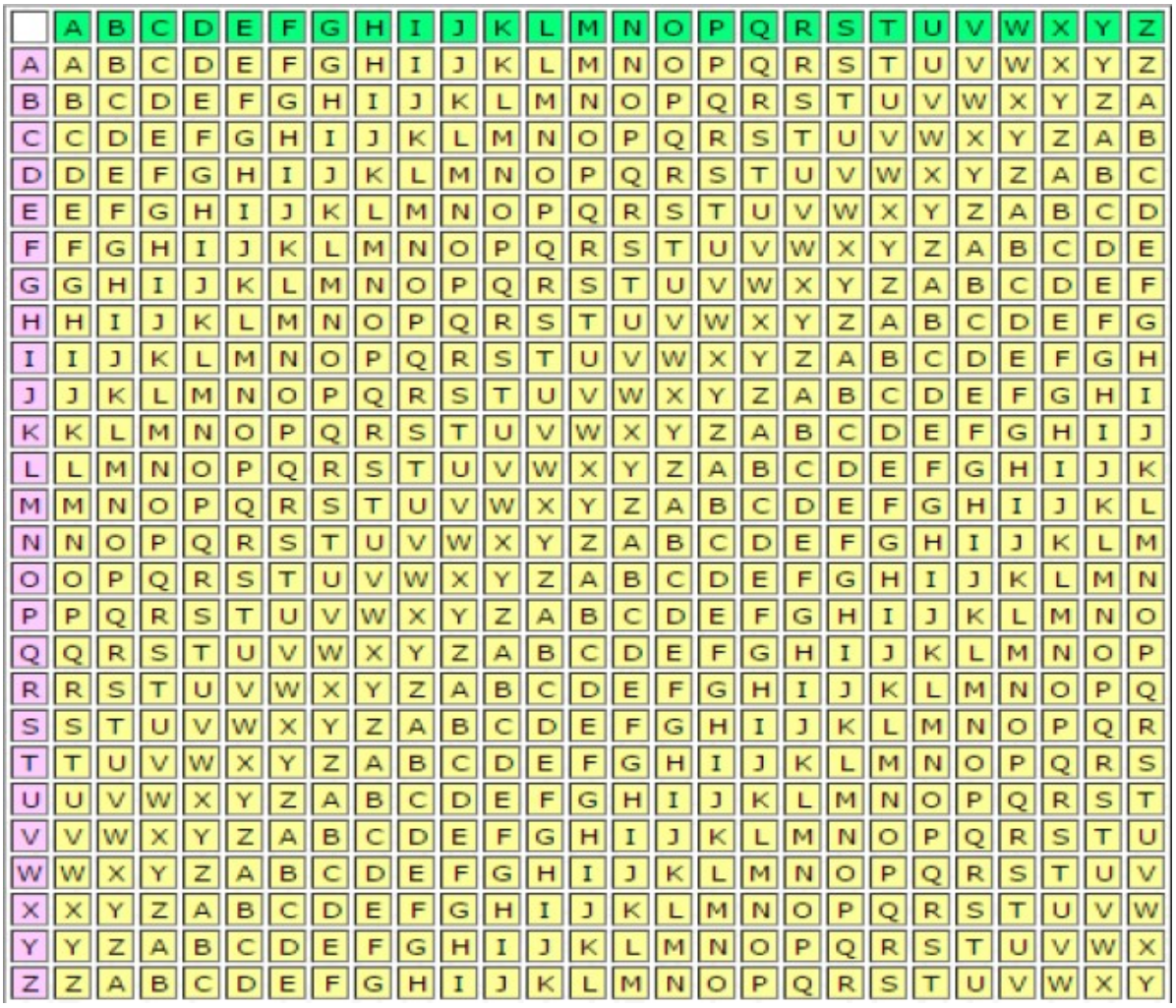
Plans texte	c	i	p	h	e	r
ck	11	23	11	11	15	15
Message chiffré	I	X	L	L	P	P

TABLE 2.4: chiffrement affine

b- Déchiffrement affine :
X=($a^{-1}(c-b)) et "a^{-1}" est l'inverse de "a"$:

2.3.1 Substitution poly alphabétique

Au XVIe siècle, Vigenère « invente » une nouvelle méthode de chiffrement en s'inspirant fortement des travaux de l'abbé allemand Jean. Vigenère a néanmoins modifié légèrement la méthode de Trithème en rendant l'utilisation de clé de chiffrement possible. [7] Les substitutions poly alphabétiques sont des combinaisons de plusieurs tables de substitutions simples, où l'on change dans chaque table la lettre, ce qui rend le message chiffré plus dur à casser sans la clé. Le principe consiste à remplacer chaque lettre du message en clair par une nouvelle lettre prise dans un ou plusieurs alphabets aléatoires associés Le chiffrement poly-alphabétique utilise deux clés : une table de chiffrement et un mot clé, qui change en général dans chaque message, comme par exemple : « Cabab » sera donc chiffré en « Cbddf»



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

FIGURE 2.1: table de végner

2.3.2 Substitution par polygrammes

Playfair Cipher a été inventé par Sir Charles Wheatstone (1802-1875) en 1854, et popularisé par Baron Lyon Playfair (1819-1898), dont j'ai immortalisé le nom pour cela algorithme. Bien que l'algorithme Playfair soit déjà dangereux pour l'utilité du monde. Les caractères du texte en claire sont chiffrés par blocs exemple, »ABA »peut être chiffré par »RTQ » tandis que « ABB » est chiffrés par « SLL«.les exemples les plus célèbres sont algorithmes de PLAYFAIRE et de HILL.

2.3.2.1 Chiffrement de Hill

La substitution est faite par groupe de lettres, elle est plus difficile pour être caser seulement par observation des fréquences d'apparition des lettres de l'alphabet. Basé

sur l'algèbre matricielle. [9]
Exemple de chiffrement :

A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	2	3	4	5	6	7	8	9	10	11	12	13	14

TABLE 2.6: alphabie affine

o	P	Q	R	S	T	U	V	W	X	Y	Z
15	16	17	18	19	20	21	22	23	24	25	26

TABLE 2.7: alphabie affine

Ahmed prend comme clef de chiffrement la matrice pour chiffrer le message « je vous D'aprèsle tableau 2.7, p1= 'j'=10 et p2= 'e'=5 ainsi de suite .....

$$\begin{pmatrix} C_k \\ C_{k+1} \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} P_k \\ P_{k+1} \end{pmatrix} \pmod{26}$$

FIGURE 2.2: fonction HILL "chiffrement"

On divise le message en bloc de deux caractères.
$C1 \rightarrow 9 * 10 + 4 * 5(mod 26) = 110(mod 26) = 6$
$C2 \rightarrow 5 * 10 + 7 * 5(mod 26) = 85(mod 26) = 7 \ldots\ldots ect$

On procède de même avec les paires de lettres suivantes, nous obtenant

LETTER	J	E	V	O	U	S
Range (ck)	10	5	22	15	21	19
Range (Pk)	6	7	20	7	5	4
Letterschifrés	F	G	X	G	E	D

TABLE 2.8: chéffrement hill

Le message chiffré sera donc" FGXGED"

2.3.2.2 Déchiffrement Hill

Pour déchiffré le message ahmed , ali doit calculer :
Pour déchiffrer le principe est le même que le chiffrement, on prend les lettres deux par deux .puis on les multiple par la matrice de déchiffrement.
Ordinairement l'inverse de la matrice :[9][10]

Exemple de de déchiffrement

$$\begin{pmatrix} P_k \\ P_{k+1} \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix}^{-1} \begin{pmatrix} C_k \\ C_{k+1} \end{pmatrix} \pmod{26}$$

FIGURE 2.3: fonction HILL (déchiffrement)

$$\begin{Bmatrix} 9 & 4 \\ 5 & 7 \end{Bmatrix} \bmod 26 = 1_{\overline{9*7-4*7}} * \begin{Bmatrix} 9 & -4 \\ -5 & 7 \end{Bmatrix} \bmod 26 = 43^{-1} \begin{Bmatrix} 9 & -4 \\ -5 & 7 \end{Bmatrix} mod 26 = 23$$

Ali peut maintenant terminer le calculé :

$$23 * \begin{Bmatrix} 7 & -4 \\ -5 & 9 \end{Bmatrix} \bmod 26 = \begin{Bmatrix} 161 & -92 \\ -115 & 207 \end{Bmatrix} \bmod 26 = \begin{Bmatrix} 5 & 12 \\ 15 & 25 \end{Bmatrix}$$

Ali utilise la nouvelle matrice pour déchiffrer le message ”FGXGED” après avoir remplacé les lettres par leur rang on calcule :

P1=5*6+12*7(mod26) = 114(mod26) = 10

P2=15*6+25*7(mod26) = 265(mod26) = 5

On procède de la sorte avec les paires de lettres suivantes, on obtient finalement [10] : Le message

Letterschifrés	F	G	X	G	E	D
Range (ck)	6	7	24	7	5	4
Range (pk)	10	5	22	15	21	19
LETTER	J	E	V	O	U	S

[h]

TABLE 2.9: déchéffrement hill

déchiffré est donc bien "je vous"

2.3.3 cryptographie par transposition

Définition

Il utilise le concept mathématique des permutations. Un chiffrement à transposition est un chiffrement dans lequel les caractères du texte en clair demeurent inchangés mais dont les positions respectives sont modifiées. [17] Pour déchiffré le texte chiffré, il suffit d'écrire verticalement celui-ci sur un morceau de papier quadrillé de la même largeur et de lire horizontalement le texte clair. Plusieurs types différents de transpositions existent.

2.3.3.1 Transposition simple par colonnes

Pour appliquer la transposition simple en colonnes, Avec le principe de la transposition de toutes les lettres du message sont présentes, mais dans un ordre différent. Il utilise le principe mathématique des permutations. On écrit le texte en clair horizontalement sur un morceau de

papier quadrillé de largeur fixe et l'on relève le texte chiffré verticalement

Exemple :
Texte chiffré = 'NBCD EFLK JIHG MNOP QRSTU VWX DTRN ZY '

N	B	C	D	E	F
l	k	j	i	h	g
M	N	O	P	Q	R
S	T	U	V	W	X
D	T	R	N	Z	Y

TABLE 2.10: dtransposition simple

Texte claire c'est :'NLMSD BKNTT CJOUR DIPVN EHQWZ FGRXY'

2.3.3.2 Transposition complexe par colonnes

Dans la transposition complexe, un mot clé secret (avec uniquement des caractères différents) est utilisé pour dériver une séquence de chiffres commençant de 1 et finissant au nombre de lettres composant le mot clé
Cette séquence est obtenue en numérotant les lettres du mot clé en partant de la gauche vers la droite et en donnant l'ordre d'apparition dans l'alphabet.
Une fois la séquence de transposition obtenue, on chiffre en écrivant d'abord le message par lignes dans un rectangle, puis on lit le texte par colonnes en suivant l'ordre déterminé par la séquence.
Exemple :
Texte clair c'est " I LOVE YOU MAMA" et lo mot clé "aymen"

$$\text{“AYMEN”} \rightarrow \begin{bmatrix} A & Y & M & E & N \\ 1 & 5 & 3 & 2 & 4 \end{bmatrix}$$

I	L	O	V	E
Y	O	U	M	A
M	A			

TABLE 2.11: transposition complexe par clonnes

Texte chiffré : "IYM VM OU EA LOA "

2.4 conclusion

la cryptographie classique utilisait des cryptosystéme basée sur les letters .
de nos jour on manipule des bits ,ou liens des caractére on passe de 26 élément a deux élément
on va voir par la suit les deux types cryptographie moderne cryptographie a clé secret et cryptographie a clé publique

2.5 Introduction

C e chapitre est consacré à la cryptographie moderne utilisée dans la transmission et le stockage sécurise de données. Après un rapide historique de la cryptographie on examinera quelques principaux systèmes cryptographiques actuellement utilisés. [20] La cryptographie moderne a commencé dans les années 1970 quand Diffue et Hellman [18] ont proposé une méthode d'échange de clefs et le développement des ordinateurs, les techniques de cryptographie ont clairement évolué, mettant à la touche ainsi les méthodes de cryptage manuel. Malgré ça, les procédés de substitution et de transposition restent toujours d'actualité mais en manipulant, cette fois-ci, des séquences de bits du fait que les ordinateurs ne manipulent que des données numériques ce qui rend les techniques de chiffrement actuelles plus sûres [17], voir même incassables pour certaines techniques, ou du moins prendraient des millions d'années avec la puissance actuelle des meilleurs supercalculateurs. D'autre part, le fait que maintenant les algorithmes ne sont plus cachés, mais au contraire sont connus de tous et leur sécurité est liée seulement aux clés utilisées. Le premier système cryptographique a clé publique a été proposé par Rivets, Shamir et Adleman (RSA).
La cryptographie est divisée en deux grands domaines : la cryptographie symétrique et la cryptographie asymétrique. Comme leurs noms l'indiquent, la différence entre ces deux domaines est une question de symétrie. En cryptographie symétrique, on suppose que tous les intervenants possèdent le même secret alors qu'en cryptographie asymétrique, un seul des intervenants possède un secret. [22]

2.6 La cryptographie symétrique ou a clé secret

Définition :
Les principes des codes symétriques commerciaux modernes du type DES (Data Encryptions Standard) ont été mis au point dans les années 1970 par IBM avec l'aide de la NSA (National Security Agency) [21]. La méthode connue pour échanger de l'information était la cryptographie à clé secrète (symétrique). Pour que deux personnes puissent échanger un message chiffré, elles devaient partager un même secret connu d'elles seules. Dans la cryptographie symétrique, les

deux utilisateurs partagent un secret commun (mot, un nombre ou une phrase). La clé étant secrète et par suite elle doit être échangée entre les intervenants par un canal confidentiel [19].
La cryptographie à clé secrète peut se classer en deux catégories :
1. les systèmes de chiffrement par bloc.
2. les systèmes de chiffrement à flot.

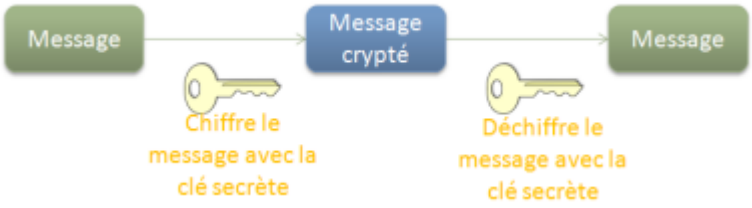


FIGURE 2.4: cryptographie symétrique

2.6.1 Chiffrement par block

Un système de chiffrement est dit par blocs s'il divise le texte clair en blocs de taille fixe (généralement 64 ou 128 bits) et chiffre un bloc à la fois avec la même clef, et renvoie des blocs de texte chiffré de taille fixe.
Les exemples les plus connus de chiffrement par blocs sont le DES (Data Encryptions Standard) qui fut adopté comme standard américain pour les communications commerciales en 1977 et qui est aujourd'hui vulnérable à la cryptanalyse. [23]

2.6.2 Chiffrement à flot (stream cipher) :

Le chiffrement par flot (ou chiffrement en continu, stream cipher) : permet de travailler sur un message de taille arbitraire, et le traiter octet par octet (voir bit par bit). Ces algorithmes sont généralement plus rapides mais moins résistants que les chiffrements par blocs [5].

Exemple :
A5/1 (chiffrement GSM) : clé=64bits (seulement 54 pour le GSM)
RC4 (WEP) : clé variable
E0 (Bluetooth) : clé=128bits en général

Un chiffrement par flot se présente sous la forme d'un générateur de nombres pseudo-aléatoires avec lequel on opère un ou exclusif (XOR) entre un bit à la sortie du générateur et un

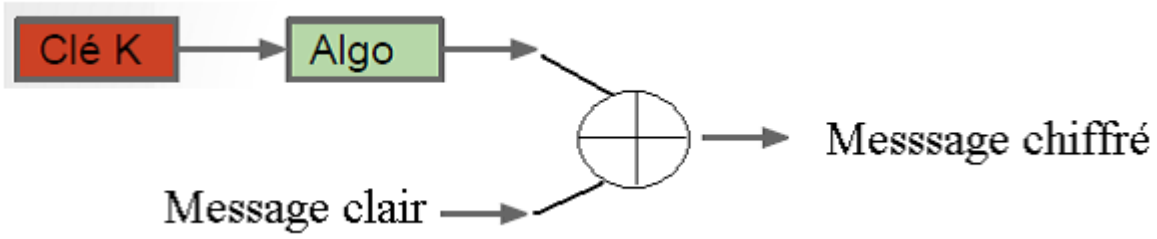


FIGURE 2.5: Principe du chiffrement par flot

bit provenant du message en clair.

Propriétés :
La séquence qui sert au chiffrement ne dépend pas du message clair, mais uniquement de la clé.
Il est possible de chiffrer des messages de tailles variables.
Le chiffrement et le déchiffrement s'effectuent de la même manière (XOR)
L'impact de la modification d'une partie du message chiffré pendant la transmission du message est limité à cette partie du message déchiffré.

Une des principales caractéristiques des algorithmes de chiffrement par flot est qu'ils permettent d'atteindre un très haut niveau de performances. Ces performances s'expriment soit en termes de vitesse de chiffrement soit en termes d'efficacité matérielle. On distingue deux principaux types d'algorithmes par flot :

Les algorithmes adaptés à une implantation logicielle, qui peuvent atteindre des vitesses de chiffrement très élevées (plusieurs Gbits/s).
Les algorithmes adaptés à une implantation matérielle, dont les implantations sont efficaces en termes de taille ou de consommation électrique. [6][7]

2.7 L'algorithme DES (Data Ecryption Standard)

L'algorithme DES transforme un bloc de 64 bits en un autre bloc de 64 bits. Il manipule des clés individuelles de 56 bits, représentées par 64 bits (avec un bit de chaque octet servant pour le contrôle de parité).
Ce système de chiffrement symétrique fait partie de la famille des chiffrements itératifs par blocs, plus particulièrement il s'agit d'un schéma de Feistel (du nom de Horst Feistel à l'origine du chiffrement Lucifer).

Le DES est basé sur les attributs fondamentaux de fonctions cryptographiques telles que substitution (confusion) et transposition (diffusion), opération OU exclusif, décalage et swap Ping. Le DES se compose de 16 tours et chaque tour utilise une seule clé ronde et que les clés sont générées à partir du générateur de la clé ronde. Le générateur de clé ronde crée le 48 bits clés pour chaque tour.[24]
D'une manière générale, on peut dire que DES fonctionne en trois étapes :

Permutation initiale et fixe d'un bloc (sans aucune incidence sur le niveau de sécurité) ;
Le résultat est soumis à 16 itérations d'une transformation, ces itérations dépendent à chaque tour d'une autre clé partielle de 48 bits. Cette clé de tour intermédiaire est calculée à partir de la clé initiale de l'utilisateur (grâce à un réseau de tables de substitution et d'opérateurs XOR). Lors de chaque tour, le bloc de 64 bits est découpé en deux blocs de 32 bits, et ces blocs sont échangés l'un avec l'autre selon un schéma de Feistel. Le bloc de 32 bits ayant le poids le plus fort (celui qui s'étend du bit 32 au bit 64) subira une transformation ;
le résultat du dernier tour est transformé par la fonction inverse de la permutation initiale.

2.7.1 Etapes de l'algorithme DES

Le message est découpé en blocs de 64 bits.
— Une permutation initiale est faite sur le bloc de 64 bits (permutation).
— Le bloc de 64 bits est découpé en deux blocs de 32 bits, et ces blocs sont échangés l'un avec l'autre selon un schéma de Feistel.
— Une extension du bloc de taille 32 bits à 48 bits est appliquée. — Une opération XOR est faite entre la clé de 56 bits et le bloc de 48 bits (Ronde).
— les 48 bits sont ensuite divisés en 8 blocs de 6 bits, chaque vecteur étant finalement traité grâce à une table d'expansion
— Des transpositions sont faites sur des blocs de 32 bits grâce à une table de substitution (S-BOx).
— L'algorithme consiste en fait en 16 itérations de cryptage, et dans chaque nouvelle itération, une nouvelle clé est utilisée. A la fin de la 16ème itération, les deux blocs de 32 bits de gauche et de droite sont réunis en un seul bloc de 64 bits (permutation inverse). [29]

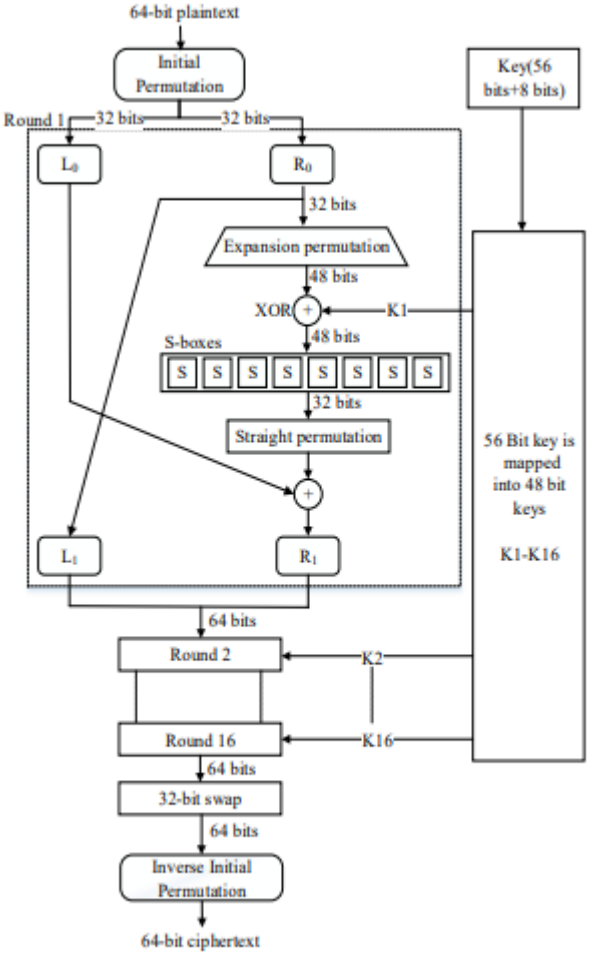


Fig 3: General Depiction of DES

FIGURE 2.6: représentation générale du DES

2.7.1.1 Diffusion

La diffusion disperse la redondance du texte clair dans le texte chiffré, par exemple deux lettres doublées ne doivent pas rester côte à côte après cryptage. [25]

Remarques :

1. Expansion et permutation permettent une diffusion rapide.

2. Le DES est un très bon algorithme, il a résisté à 30 ans de cryptanalyse. Finalement, la taille des clés étant petite, il a succombé à la cryptanalyse différentielle (attaque à texte clair choisi) et linéaire (attaque à texte clair connu).

3. La clé DES étant trop courte, cet algorithme est très vulnérable aux attaques par force brute. En 1997, la clé a été cassée en 3 semaines. En 1998 , elle a été cassée en 56 heures et en 1999, elle a été cassée en 22 heures. Le triple DES a ensuite été proposé avec de plus grandes clés, afin de prolonger la durée de vie du DES en attendant l'AES.

2.7.1.2 Confusion

La confusion gomme les relations entre le texte clair et le texte chiffré pour éviter les attaques par analyse statistique. Autrement dit un changement d'un seul bit dans le texte clair doit affecter un grand nombre de bits (idéalement tous) du texte chiffré.

2.8 L'algorithme AES (cipher bloc Advenced Encryption Standard).

L'AES (Rinjdaël) est le standard actuel utilisé pour les chiffrements symétriques. Contrairement à DES, il n'utilise pas de réseau de Feistel. La taille du bloc est de 16 octets (128 bits), alors que la taille de la clé est de trois variantes (128, 192, 256 bits).
Il a été publié en 2001 suite à un appel du NIST ayant donné lieu à cinq ans de compétition (FIPS-PUB 197) entre plusieurs algorithmes (MARS, RC6, Rinjdaël, Serpent, Twofish).[27]

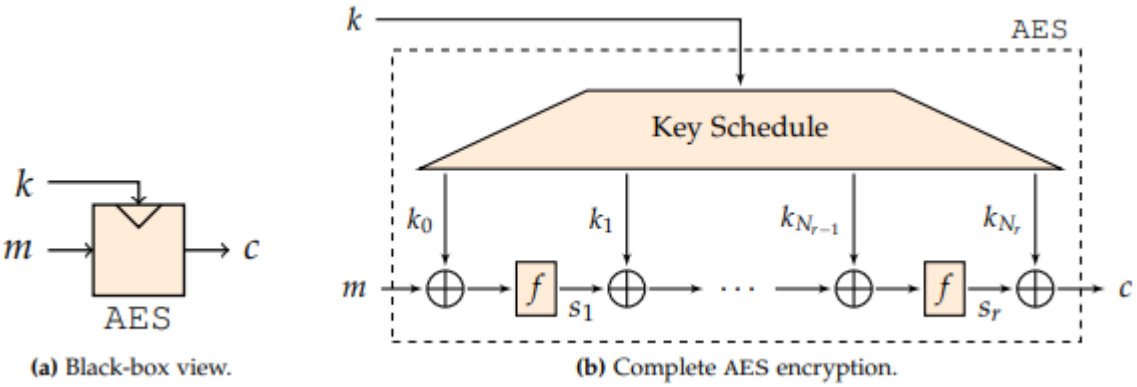


FIGURE 2.7: Le chiffrement AES.

Performances :
AES128 est 2,7 fois plus rapide que 3DES et presque aussi rapide que le DES.

2.8.0.1 Le corps A.E.S

Représentation :
Mot de 8 bit correspond à un mot de deux chiffres hexadécimaux et à des polynômes de F2 de degré 7. Les octets sont représentés en hexadécima

2.8.0.2 Le corps F2 :

L'ensemble F2= 0,1 des entiers modulo 2 est un corps (la multiplication correspond au « et logique » et l'addition à l'addition modulo 2).
Exemple : On identifie
8A =1000 1010=1x$^7 + 0x^6 + 0x^5 + 0x^4 + 1x^3 + 0x^2 + 1x^1 + 0$

Pour le corps A E S, on travaille dans le quotient F2[x] /R(x)
R(x) est le polynôme de Rinjdael (irréductible sur F2)
R(x)=(x)$^8 + (x)^7 + (x)^3 + (x)^1 + 1$
$F2[X]/R[X] = F2^8$

2.8.1 Fonctionnement d'AES

On découpe les données et les clés en octets et on les place dans des tableaux. Les données comportent td=16 octets P0, P1,..... P15 qui sont classés dans un tableau ayant 4 lignes et 4 colonnes. Le tableau est rempli colonne par colonne. De même la clé est découpée en octets (tk=16, tk=24 ou tk=32 octets) k0 ,k1 ,..... ,k tk-1 .Ces octets sont aussi classés dans un tableau de 4 lignes et Nk colonnes (Nk=4, Nk=6 ou Nk=8).

p_0	p_4	p_8	p_{12}
p_1	p_5	p_9	p_{13}
p_2	p_6	p_{10}	p_{14}
p_3	p_7	p_{11}	p_{15}

k_0	k_4	k_8	k_{12}	k_{16}	k_{20}	k_{24}	k_{28}
k_1	k_5	k_9	k_{13}	k_{17}	k_{21}	k_{25}	k_{29}
k_2	k_6	k_{10}	k_{14}	k_{18}	k_{22}	k_{26}	k_{30}
k_3	k_7	k_{11}	k_{15}	k_{19}	k_{23}	k_{27}	k_{31}

FIGURE 2.8: les matrices

Le système AES effectue plusieurs tours d'une même composition de transformations.
• Le nombre de tours
Suivant la version (la taille de la clé), ce nombre de tours noté nr, est différent. Le nombre nr est donné dans le tableau suivant.

NK	4	6	8
Nr	10	12	14

- La fonction d'étage 'g' est l'ensemble des opérations que l'on fait subir au texte qui arrive à l'étage i. Elle est la même pour tous les étages sauf le dernier.

- La fonction 'g' consiste en l'application successive de 4 opérations :
SUBBYTES
SHIFTROWS
MIXCOLUMNS
ADDROUNDKEY

Pour le dernier étage, on applique seulement les opérations :
SUBBYTES
SHIFTROWS
ADDROUNDKEY

• La clé de tour

A partir de la clé secrète K, on génère des sous clés d'étages par l'algorithme de diversification de clé, EXPANDKEY[i]
On obtient les clefs, K1, K2, . . . , Knr k^1, k^2, ..., k^{Nr}

Tableau d'itérations [28]

Il applique quatre opérations à la matrice d'état (sauf la dernière où nous omettons les Mix-Columns) :
Add-RoundKey (AK) ajoute la sous-clé 128 bits suivante à l'état. Sub-Bytes (SB) applique la même S-Box S bijective 8 bits à 8 bits 16 fois en parallèle sur chaque octet de l'état.
Shift-Rows (SR) décale la i-ème rangée vers la gauche de i positions.
Mix-Columns (MC) remplace chacune des quatre colonnes C de l'état par M × C où M est une matrice séparable à distance maximale constante 4 × 4 sur GF (28).[27][29]

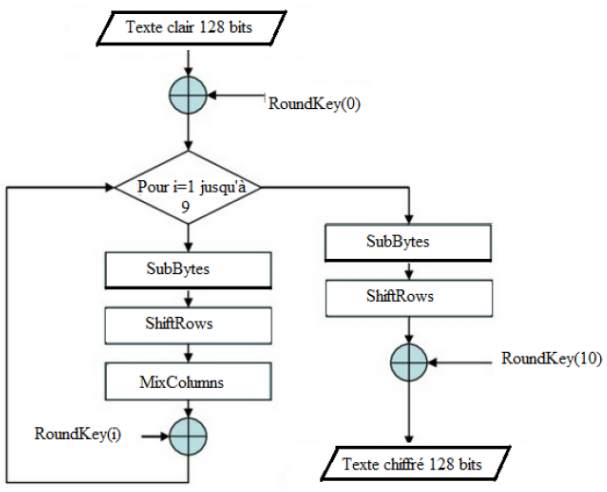


FIG. 6-2 – Principe du chiffrement AES.

FIGURE 2.9: principe de chiffrement AES

2.9 Cryptographie 3DES

En cryptographie, Triple DES est également appelé algorithme de cryptage de données triple, qui est un chiffrement par blocs. Triple Data Encryptions Standard (3DES) a été publié pour la première fois en 1998 et tire son nom ainsi parce qu'il applique le chiffrement DES[30]. Le DES a été remplacé par Triple DES ou 3DES, qui est DES avec trois clés.
Il est essentiel d'éviter d'avoir la même clé pour les étapes de chiffrement depuis la sortie. Il a deux formes, l'une nécessitant trois touches complètement différentes et les deux autres seulement deux touches complètement différentes.
Sa longueur de la clé est de 112 bits ou 168 bits et la taille du bloc est de 64 bits. En raison de la puissance de calcul croissante disponible ces jours-ci et de la faiblesse du chiffrement DES original, il a été soumis à des attaques par force brute et à diverses attaques cryptanalyse, Triple DES a été conçu pour fournir une méthode relativement simple pour augmenter la taille de la clé du DES pour se protéger contre de telles attaques [31]

2.9.1 Principe de chiffrement 3DES

La première méthode utilise trois clés pour crypter le texte en clair :
1. Tout d'abord en utilisant la clé k1.
2. Suivie par cryptage avec la clé k2.
3. Enfin un troisième cryptage est effectué avec la clé k3.
Nous effectuer l'opération C = EK3 (EK2 (EK1 (P))) pour crypter le texte en clair
P =DK3 (DK2 (DK1 (C))) pour le décryptage.
PGP et S / MIME sont des exemples de produits qui utilisent les trois touches 3DES. Même si 3DES utilise trois clés pour fournir un niveau élevé de sécurité, il a toujours un inconvénient car il faut 56 * 3 = 168 bits pour les clés, ce qui peut être difficile à faire fonctionner dans des situations pratiques. Pour cette raison, la méthode de 3DES n'utilisae que deux clé. [30]

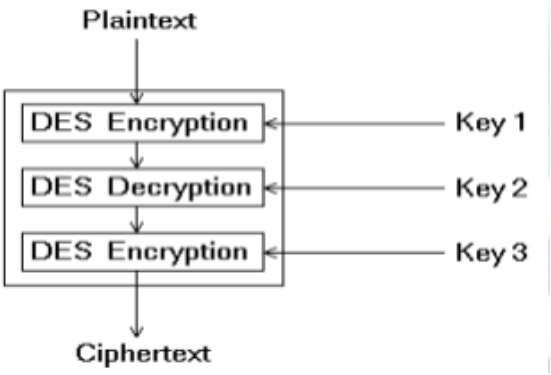


FIGURE 2.10: 3DES

2.10 Comparaison des trois chiffrements DES 3DES et AES

1.3DES utilise un cryptage identique à DES alors qu'AES en utilise un complètement différent, 3DES a une longueur plus courte et plus faible.
2. les clés de chiffrement par rapport à AES, et 3DES applique à plusieurs reprises des clés de chiffrement, contrairement à AES. 3.AES est fortement résistant aux attaques différentielles, différentielles tronquées, linéaires, interpolées et carrées, contrairement à 3DES qui est vulnérable à la cryptanalyse différentielle et linéaire et possède des tables de substitution faibles.
4. le temps nécessaire pour vérifier toutes les clés possibles à 50 milliards de clés par seconde en AES pour une clé de 128 bits est de 5 x 1021 ans, alors que 3DES avec une clé de 56 bits prendrait 400 jours.
5. 3DES utilise un bloc longueur de 64 bits, soit la moitié de la taille d'un bloc AES de 128 bits.Un autre inconvénient lors de l'utilisation de 3DES est la nécessité de changer les clés de cryptage tous les 32 Go de transfert de données pour réduire les risques de fuites.
6. l'utilisation d'AES fournit assurance supplémentaire car il est difficile de déchiffrer les données de blocs identiques. Le processus de cryptage 3DES est beaucoup plus long que AES, car la répétition du
7. le même processus de cryptage trois fois dans 3DES prend un certain temps par rapport au processus de cryptage AES qui est beaucoup plus rapide.[31]

2.11 Conclusion

dans le systeme de cryptographie a clé secrétes la fonction de déchiffrement ce déduit facilment de la fonction de chiffrement et avant tout utilisation d'un tel systéme , le couple (émitteur,déstinateur) doit ce metter d'acourd sur une clé , ce qui pose le probléme de la déstribution des clé.

2.12 Chiffrement clé asymétrique

Définition :
La cryptographie asymétrique ou cryptographie à clé publique fonctionne de façon totalement différente à la cryptographie symétrique. Chaque utilisateur possède deux clés mathématiquement liées : l'une secrète et l'autre publique. Ce qui est chiffré par l'une des deux clés n'est déchiffrable que par l'autre.[35]
Si l'on peut comparer la cryptographie symétrique à un coffre-fort auquel seules les personnes possédant la clé peuvent accéder, la cryptographie asymétrique pourrait être comparée à une boite aux lettres dans laquelle on peut déposer des informations, et seule la personne possédant la clé peut accéder au contenu de la boîte. La boite aux lettres serait la clé publique (donc accessible

à tout le monde), alors que la clé pour ouvrir la boite serait la clé privée. .[34]

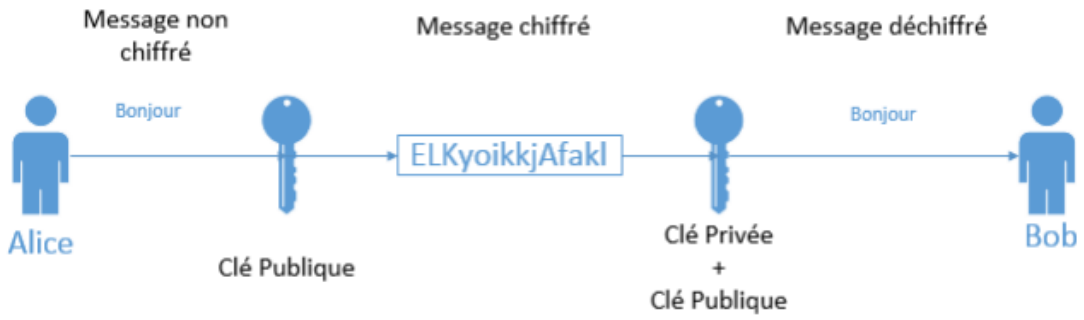


FIGURE 2.11: chiffrement asymétrique

2.12.1 Chiffrements Diffie helman

Définition :
Diffie et Hellman ont travaillé sur la cryptographie à clé publique tandis que Merkle a fait ses contributions sur la distribution de clé publique. Lorsqu'ils ont pris conscience du travail de chacun, il a décidé de travailler ensemble dans l'espoir de meilleurs résultats. Cela a conduit plus tard à la publication de leur article commun intitulé "New Directions in Cryptographie" - publié en novembre 1976. Cet article a apporté une nouvelle idée au domaine de la cryptographie ; il a décrit les concepts clés de la cryptographie à clé publique tels que la production de signatures numériques, et a donné quelques algorithmes pour la mise en œuvre L'idée de la cryptographie à clé publique est née à la suite de défis majeurs. Le premier d'entre eux était le problème de la distribution des clés : si deux personnes qui ne se sont jamais rencontrées auparavant doivent communiquer à l'aide de systèmes numériques. [36]

2.12.2 Protocole de Diffie helman :

Ce protocole permet un échange de clés pour des systèmes de chiffrements asymétriques où aucun secret n'est partagé à aucun moment entre l'émetteur et le récepteur. L'opération de déchiffrement doit être contrôlée par une clé gardée secrète. Cette dernière ne peut être déduite de la clé publique. Pour cela des fonctions à sens unique doivent être utilisées. Parmi ces fonctions, nous citons le logarithme discret [36]

2.13 L'algorithme RSA (Ron Rivest, Adi Shamir, et Len Adlmen)

L'algorithme RSA est sans doute le plus utilisé des systèmes à clé publique actuellement ; il a été présenté en 1977 par Ron Rivest, Adi Shamir, et Len Adlmen. Il nécessite des clés d'au moins 1024 bits pour obtenir une sécurisation satisfaisante.

2.13.1 Principe de fonctionnement du RSA [34][35]

Etape 1 : Création des clés
le récepteur choisit au hasard deux nombres premiers p et q et on calcule n = p.q . Il choisit au hasard e tel que :

$$\begin{cases} 1 < e < \varphi n = (p-1)(q-1) \\ \quad pgcd(e, \varphi n) = 1 \end{cases}$$

φn,le nombre d'entiers inférieurs à n, premiers avec n Le récepteur calcule l'entier d pour inverser la fonction de chiffrement tel que :

$$\begin{cases} 1 < d < (\varphi n) \\ 1 mod(\varphi n) \end{cases}$$

La clé publique du récepteur est (n,e) et sa clé secrète est (n,d).
Etape 2 : Chiffrement du message
L'émetteur récupère la clé publique (n,e) du récepteur et souhaite lui envoyer la version cryptée d'un texte en clair, représenté par la donnée d'un entier m tel que :
$0 \leq m < n$

L'émetteur calcule : C=m$^e mod n$, Etape 3 : Déchiffrement du message
Lorsque le récepteur reçoit C, il calcule C^d, $et récupère ainsi le message puisque$:
$m = c^d mod n$

Ce qu'est en rouge sont des nombres privée du récepteur.
Le principe de l'algorithme RSA repose sur le fait qu'il est très difficile et très long de factoriser un très grand nombre en deux facteurs premiers, pour cela il faut procéder de la manière inverse : Générer les deux nombres premiers (p et q), puis les multiplier pour générer le nombre n.
Le réel problème du RSA (et des autres systèmes à clé publique) n'est pas la sécurité, mais la lenteur. Tous les algorithmes à clé publique sont 100 à 1000 fois plus lents que les algorithmes à clé secrète, quelle que soit leur implémentation (logicielle ou matérielle) !
En dépit de sa lenteur pour encoder de grands volumes de données, le RSA est le système de cryptographie asymétrique le plus utilisé actuellement dans les cartes bleues, les logiciels, les cartes bancaires, les messageries, ...

2.13.1.1 Exemple :

Choisir p = 7, q = 11 premiers entre eux Pour calculer n, il faut multiplier p et q, ce qui nous donne n = 77.
Pour calculer $\varphi n, il faut faire (7-1)*(11-1) ce qui donne 60$.

Maintenant, il faut choisir l'exposant e, qui doit être premier avec φn, . Prenons donc par : e = 13 (PGCD (13, 60) = 1 donc premiers entre eux). Voici donc la clé publique e = 13 et n = 77.
Pour calculer d, il suffit de faire 13-1 mode 60, ce qui donne 37.

Pour vérifier que d est correct, il suffit de calculer e * d mode φn = 1, .
Dans cet exemple Cela revient à calculer 13 * 37 mod 60 = 1 Alice et Bob s'entendent publiquement sur un groupe G cyclique d'ordre n et sur un générateur g de ce groupe.

Alice tire secrètement un kA entier au hasard avec (1 kA n)et calcule dans G. Elle envoie alors à Bob. On remarque qu'un espion qui écoute la communication entre Alice et Bob ne peut retrouver que s'il sait résoudre le problème du logarithme discret et calculer. Bob, de son coté, tire secrètement un entier au hasard avec et calcule qu'il envoie à Alice. Alice reçoit donc et calcule et Bob qui a reçu calcule. Ainsi, Alice et Bob partagent un secret en commun sans qu'aucun des deux aient besoin de savoir le secret de l'autre. Cette valeur commune partagée va permettre de procéder à des chiffrements symétriques.

2.14 Introduction

Nous avons présenté dans ce chapitre, les résultats obtenus de l'implémentation des algorithmes cryptographie (classique, symétrique, asymétrique.). En termes de temps de chiffrement et de déchiffrement, ainsi que de l'espace consommé par chaque chiffrement. Selon les valeurs résultantes de chaque codage, nous extrayons le plus approprié ou idéal

2.15 L'environnement de développement

FIGURE 2.12: python

2.16 Qu'est-ce le langage Python en informatique ?

créé par le programmeur Guido van Rossum en 1991.
Python est un langage de programmation puissant et facile à apprendre. Il possède des structures de données de haut niveau efficaces et une approche simple mais efficace de la programmation orientée objet. La syntaxe élégante et le typage dynamique de Python, ainsi que sa nature interprétée, en font un langage idéal pour les scripts et le développement rapide d'applications dans de nombreux domaines sur la plupart des plates-formes. [33]

Python est le langage de programmation open source le plus employé par les informaticiens. Ce langage s'est propulsé en tête de la gestion d'infrastructure, d'analyse de données ou dans le domaine du développement de logiciels.

2.16.1 A quoi sert le langage Python ?

Les principales utilisations de Python par les développeurs sont :
la programmation d'applications
la création de services web

2.16.2 Qu'est-ce que Tkinter ?

Tkinter est une bibliothèque Python utilisée pour créer et développer des applications basées sur une interface graphique. Il est entièrement open source et fonctionne sous Windows, Mac, Linux et Ubuntu. Dans le système d'exploitation Windows, nous pouvons installer la bibliothèque Tkinter à l'aide de la commande pip install tkinter. Il installera tous les autres modules fournis avec la bibliothèque Tkinter. Tkinter peut également être installé sur le notebook Jupyter, en utilisant la commande pip install tkinter. Nous pouvons exécuter toutes les commandes standard de Tkinter dans le bloc-notes Jupyter.
Une fois que nous avons installé Tkinter dans Jupyter notebook, nous pouvons vérifier l'installation en tapant la commande suivante

from tkinter import *
PyCryptodome :
n'est pas un wrapper pour une bibliothèque C distincte comme Open SSL. Dans la mesure du possible, les algorithmes sont implémentés en Python pur. Seuls les éléments extrêmement critiques pour les performances (par exemple, les chiffrements par blocs) sont implémentés en tant qu'extensions C.[34]
7.Python-RSA :
est une implémentation purement Python RSA. Il prend en charge le chiffrement et le déchiffrement, la signature et la vérification des signatures et la génération de clés selon PKCS1 version 1.5. Il peut être utilisé comme une bibliothèque Python ainsi que sur la ligne de commande. Le code a été principalement écrit par Sybren A. Stüvel. [37]
8.Base64 :
est un algorithme de codage qui vous permet de transformer n'importe quel caractère en un alphabet composé de lettres latines, de chiffres, de plus et de barres obliques. Grâce à cela, vous pouvez convertir des caractères chinois, des emoji et même des images en une chaîne « lisible », qui peut être enregistrée ou transférée n'importe où.. [35]

9. timeit python :
Python fournit de nombreuses façons de mesurer le temps d'exécution d'un morceau de code Python. Une façon consiste à utiliser le module de temps python intégré et à gagner du temps avant et après l'exécution du programme ?
Le module timeit exécute le code environ 1 million de fois (valeur par d[héfaut) et prend en compte le temps minimum nécessaire pour exécuter ce morceau de code.[36]

2.17 jupyter notebook

FIGURE 2.13: jupyter

Jupyter Notebook est une application Web open source qui vous permet de créer et de partager des documents contenant du code en direct, des équations, des visualisations et du texte narratif. Les utilisations incluent :

le nettoyage et la transformation des données, la simulation numérique, la modélisation statistique, la visualisation des données, l'apprentissage automatique et bien plus encore.
Aussi, est un outil de travail devenu très populaire dans les milieux académique et scientifique pour pouvoir facilement travailler sur du code encore au stade de prototype. Il existe plusieurs systèmes de notebooks, Jupyter est le plus répandu d'entre eux dans le monde de la data science (et au-delà probablement). Mais saviez-vous que la première version de cet outil a été lancée en 2015 ? Cela veut dire qu'en 5 ans, cet outil s'est imposé comme un incontournable dans la boîte à outils du data scientist !

2.17.1 Comment fonctionne un notebook Jupyter ?

En détaillant le fonctionnement global d'un notebook, on va déjà voir apparaître plusieurs raisons qui ont rendu cet outil si populaire.

2.17.1.1 La structure de Jupyter se compose de 3 éléments principaux :

le notebook à proprement parler, qui est un fichier avec une extension .ipynb. Ce fichier contient votre code ainsi que l'output généré par ce code lors de son exécution une application web, qui est chargée d'afficher les notebooks et permet à l'utilisateur d'interagir avec un kernel, chargé d'exécuter le code contenu dans le notebook. Il peut s'agir d'un kernel Python....ect .[32] en fonction du langage que vous utilisez dans votre notebook

:

2.18 Architecteur de application

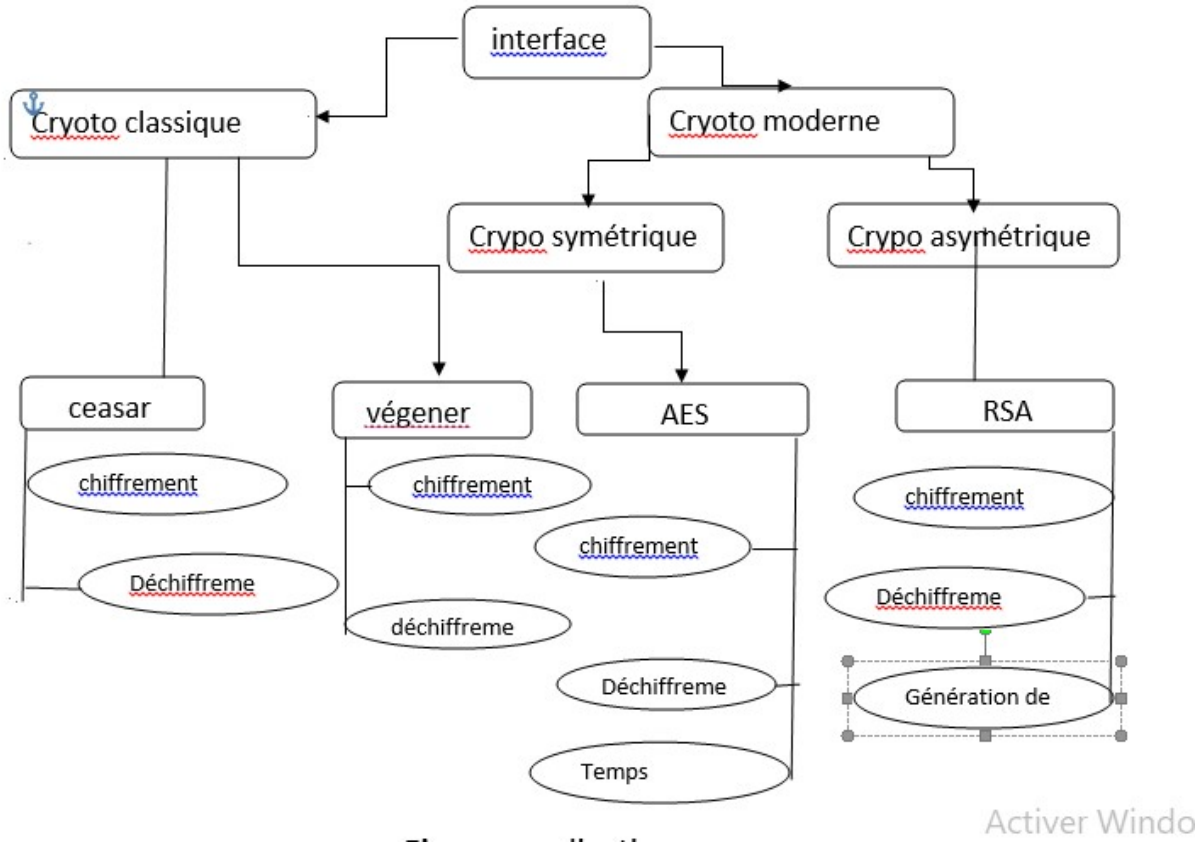


FIGURE 2.14: architecteur générale de application

2.19 Les algorithmes on pratique :

Le logiciel ce présumée sous forme de plusieurs algorithme de cryptographie on python. ils maintenant on pratique les déférant algorithme études précédemment

2.19.1 Les déffirants algorithme utilisée

Interface graphique principale d'application

h

FIGURE 2.15: sesar chiffrement

2.20 Chiffrement de sesar

interface de chiffrement sesar contient de champs texte le premier permet rentré le texte « message=master 2 sic » et affiché sous forme inverse et deux buttons « clear »et « encrypt »et « decrypt

Pour mettre en place d'algorithme la formule d'écrit dans le chapiter 1 chiffrement sesar a été utilisé .toutefois quelque traitement préalable on été effectué une foix cliqué le buttons « encrpyt » Encoder « plaintext et choisie key ou l'inverse.
Ou le texte chiffré est affiché dans label « ciphertext ».une foix cliqué le buttons « clear » Vous effacez les données

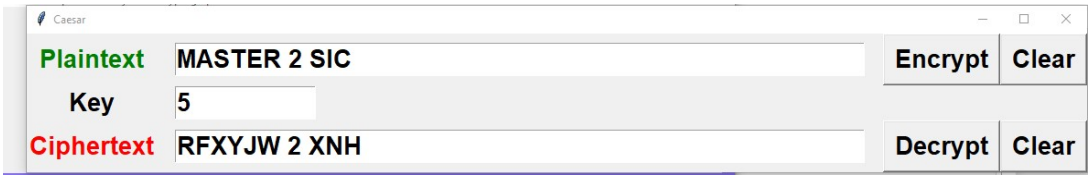


FIGURE 2.16: Déchiffrement seasar

2.21 Chiffrement de vegner

L'interface affiche quatre chiffres d'écriture, chacun avec son propre nom. « message » et « key » et mode « encryptions=e » et « décryptions=d » et résultat de chiffrement et déchiffrement le message .
Et trois boutons séparément, lorsque vous appuyez « show message » sur le résultat du cryptage apparaît
bouton pour « reset » effacer les données, et bouton pour « éxit » Pour quitter l'application

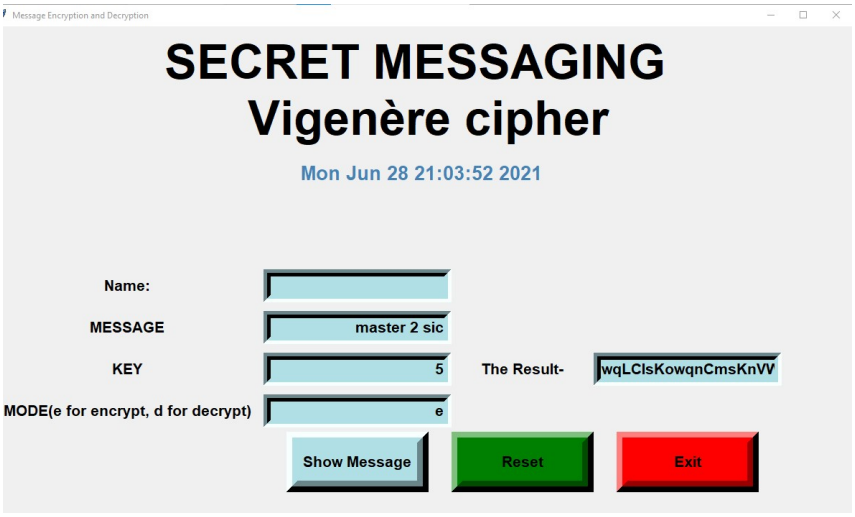


FIGURE 2.17: chiffrement végnere

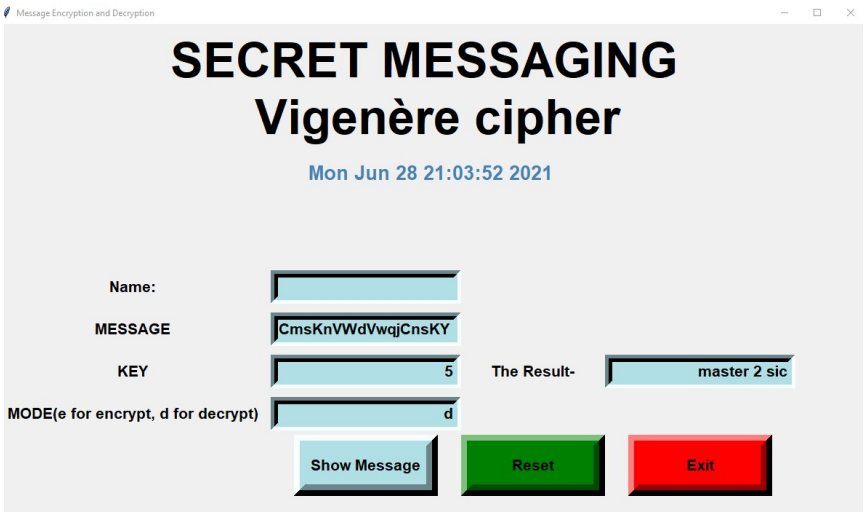


FIGURE 2.18: déchiffrement végnere

2.22 L'algorithme AES

AES a une taille de bloc de 128 bits et cette implémentation d'AES prend en charge 3 tailles de clés, 16, 24 ou 32 octets de long pour AES-128, AES-192 ou AES-256 respectivement.
1. Ouvrez l'application et cliquez sur le bouton SELECT FILE et sélectionnez votre fichier, par exemple "abc.jpg".
2. Entrez votre clé secrète (cela peut être n'importe quelle lettre alphanumérique). Souvenez-vous en pour pouvoir décrypter le fichier plus tard.
3. Cliquez sur le bouton ENCRYPTER pour crypter. Un nouveau fichier crypté avec l'extension ".kryp", par ex. "abc.jpg.kryp" sera créé dans le même répertoire que "abc.jpg".

FIGURE 2.19: Figure chiffrement TEXT.txt AES

4. Lorsque vous souhaitez décrypter un fichier, vous sélectionnez le fichier avec l'extension ".kryp" et entrez votre clé secrète que vous avez choisie au moment du cryptage. Cliquez sur le bouton DECRYPT pour déchiffrer. Le fichier déchiffré
portera le même nom qu'avant avec le suffixe "", par ex. "abcdekrypted.jpg
5. Cliquez sur le bouton RESET pour réinitialiser les champs de saisie et la barre d'état.
6. Vous pouvez également cliquer sur le bouton CANCEL pendant le cryptage/décryptage pour arrêter le processus.

2.23 L'algorithme RSA

Lapplication RSA décompose deux partie pour crée le message . Afin d'écrire le texte à crypter, deux paires de boutons les séparent, la face inférieure affiche le résultat de chaque cas.

FIGURE 2.20: déchiffrement RSA

: Dans le cas de chiffrement ,l'utilisateur doit crée le message dans label et choisie le nimber de clé « 512 ou 1024 ou 2048 »

2.24 Comparaison et performance des tests des algorithmes

le texte chéffrie c'est"master 2 sic"

2.24.1 Le temps d'exécution

Asymétrique et symétrique :
AES vs RSA

taille de clé	**128**	**192**	**256**	**512**	**1024**	**2048**
fonction	**temps d'exécution AES**			**temps d'exécution RSA**		
chiffrement	**0.00026S**	**0.0741S**	**0.0445S**	**0.0011S**	**0.0017S**	**0.0039**

TABLE 2.12: temps d'éxicution de chiffrement AES vs RSA

:

taille de clé	**128**	**192**	**256**	**512**	**1024**	**2048**
fonction	**temps d'exécution AES**			**temps d'exécution RSA**		
déchiffrement	0.0028S	0.0425S	0.0464S	0.0066S	0.0366S	0.2925S

TABLE 2.13: temps d'éxicution de déchiffrement AES vs RSA

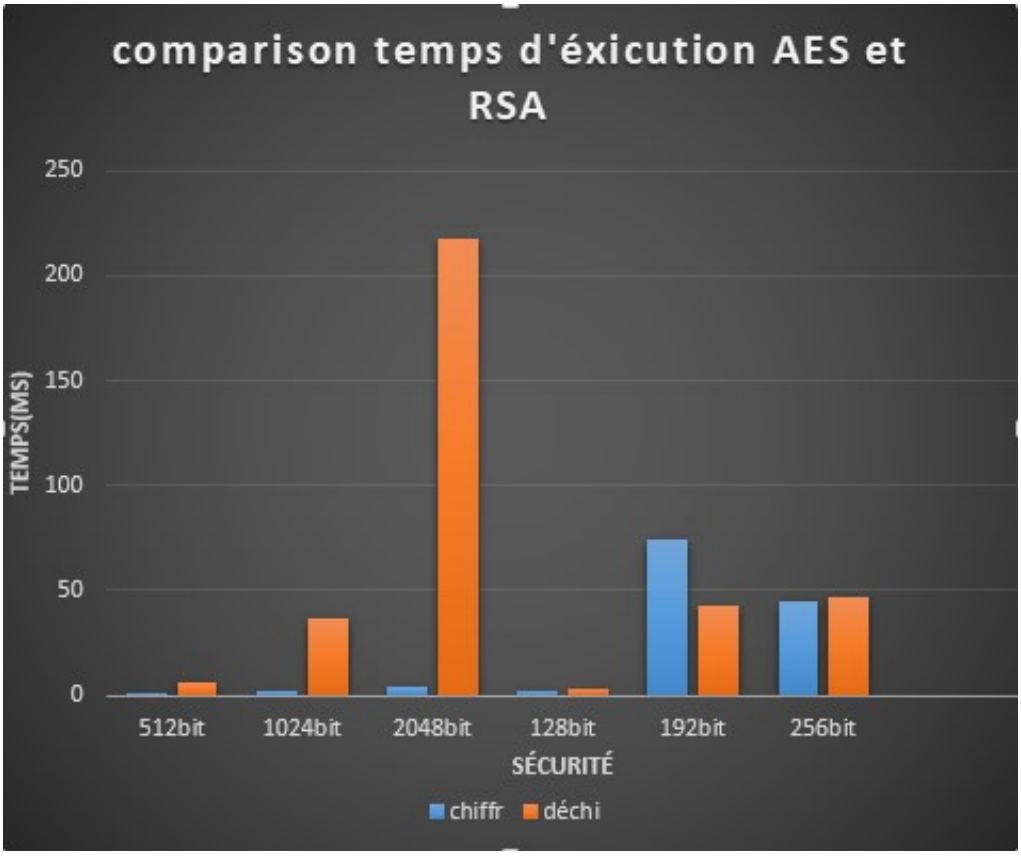


FIGURE 2.21: Histogramme1 : Comparaison 1 AES et RSA

Expérimenter avec un texte légèrement plus grand, par exemple. « En général, la cryptographie est une technique d'écriture»

taille de clé	**128**	**192**	**256**	**512**	**1024**	**2048**
fonction	**temps d'exécution AES**			**temps d'exécution RSA**		
chiffrement	0.0370 S	0.3852 S	**0.03507 S**	**0.0010 S**	**0.0016 S**	**0.0031 S**

TABLE 2.14: temps 2 d'éxicution de chiffrement AES vs RSA

:

taille de clé	**128**	**192**	**256**	**512**	**1024**	**2048**
fonction	**temps d'exécution AES**			**temps d'exécution RSA**		
déchiffrement	**0.0031 S**	**0.002453 S**	**0.024534 S**	**0.0065 S**	**0.0355 S**	**0.2158 S**

TABLE 2.15: temps 2 d'éxicution de déchiffrement AES vs RSA

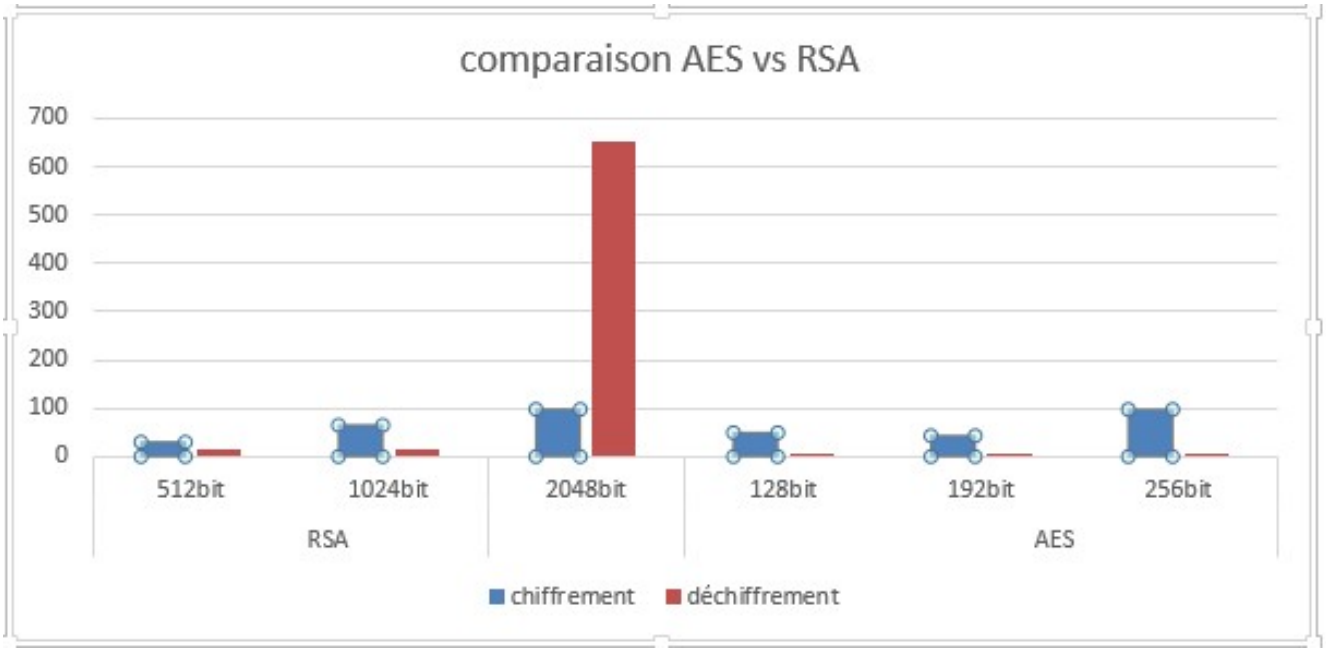


FIGURE 2.22: Histogramme2 : Comparaison 2 de déchiffrement AES et RSA

Maintenant, nous augmentons la taille du fichier « Corruption Résumé p.txt »

taille de clé	**128**	**192**	**256**	**512**	**1024**	**2048**
fonction	**temps d'exécution AES**			**temps d'exécution RSA**		
chiffrement	0.04940 S	0.4316 S	**0.09762 S**	**0.030 S**	**0.062 S**	**0.092 S**

TABLE 2.16: temps 3 d'éxécution de chiffrement AES vs RSA

taille de clé	**128**	**192**	**256**	**512**	**1024**	**2048**
fonction	**temps d'exécution AES**			**temps d'exécution RSA**		
déchiffrement	**0.0079 S**	**0.0068 S**	**0.0066 S**	**0.0147 S**	**0.0153S**	**0.6518 S**

TABLE 2.17: temps 3 d'éxécution de déchiffrementAES vs RSA

:

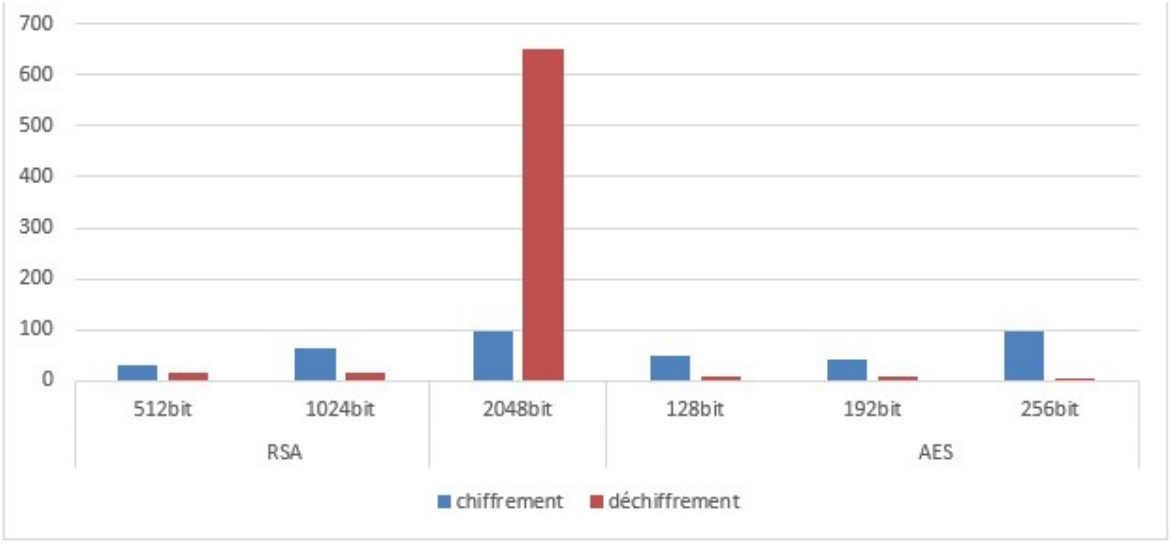


FIGURE 2.23: Histogramme3 : Comparaison 3 AES et RSA

Les résultats obtenus à partir (tableau 1 comparison AES et RSA) montrent clairement que On note une convergence dans les résultats du chiffrement et du déchiffrement du message Pour AES à la fois pour 256 bits et 128 bits, il y a une légère augmentation du temps de décodage pour 192 bits par rapport au temps requis pour le cryptage. Quant RSA le temps de décodage est supérieur à l'encodage du message. Le chiffrement prend très peu de temps, quel que soit le type de clé soit 512 bits, 1024 bits ou 2048 bits..mais on constate que plus la longueur de la clé est longue, plus le temps de déchiffrement est long classique

fonction	**temps d'xicution sésar**	**temps d'éxicution vegner**
chiffrement	0.0000565s	0.0004
déchiffrement	0.00001930s	0.0006

‘

TABLE 2.18: temps d'éxicution SESAR vs VEGNER

:

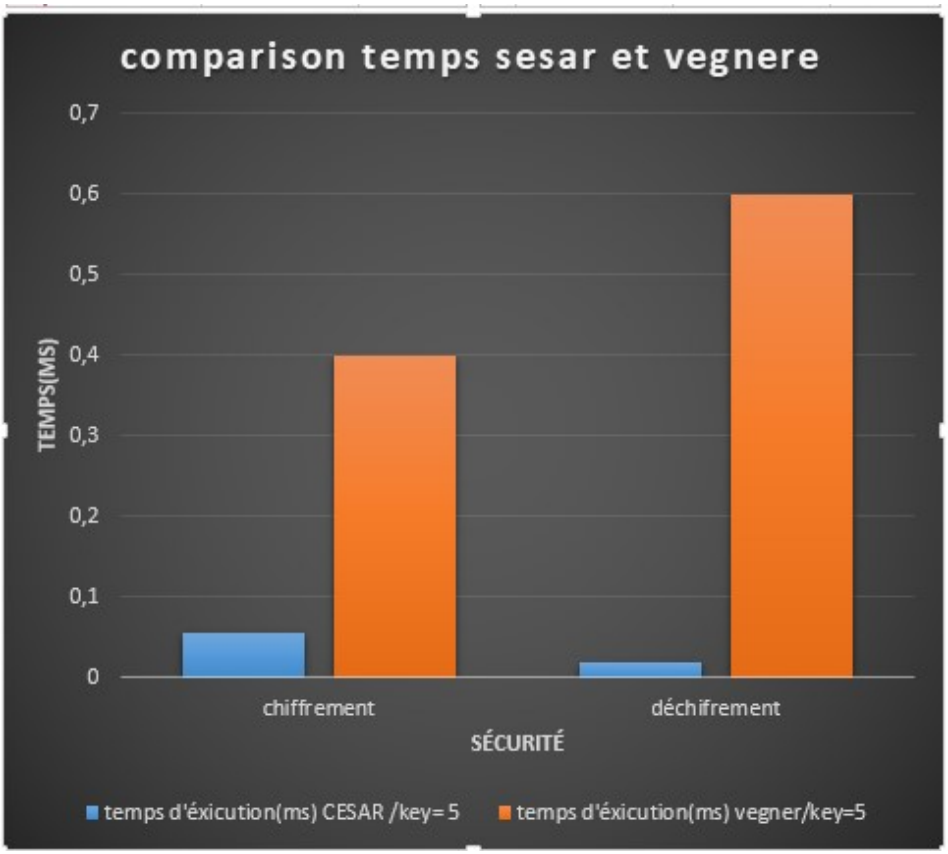


FIGURE 2.24: Histogramme4 : Comparaison temps chiffrement et déchiffrement de César Vs Vigener

Nous allons Expérimenter avec un texte légèrement plus grand, par exemple. « En général, la cryptographie est une technique d'écriture»

	temps d'exécution seasar(s)	**temps d'exécution végnere(s)**
chiffrement	**0.0001861 S**	**0.0068 S**
déchiffrement	**0.0005020 S**	**0.0040 S**

TABLE 2.19: temps 2 d'éxicution SESAR vs VEGNER vs RSA

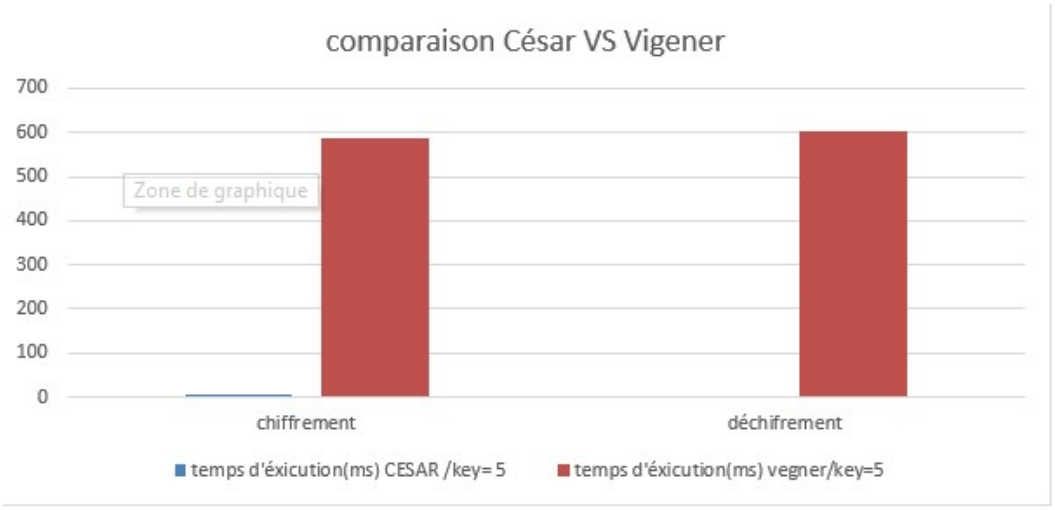


FIGURE 2.25: Histogramme5 : Comparaison temps chiffrement et déchiffrement de César Vs Vigener

Maintenant, nous augmentons la taille du fichier

	temps d'exécution seasar(s)	**temps d'exécution végnere(s)**
chiffrement	**0.004766 S**	**0.5868 S**
déchiffrement	**0.003011 S**	**0.6036 S**

TABLE 2.20: temps 3 d'éxicution SESAR vs VEGNER vs RSA

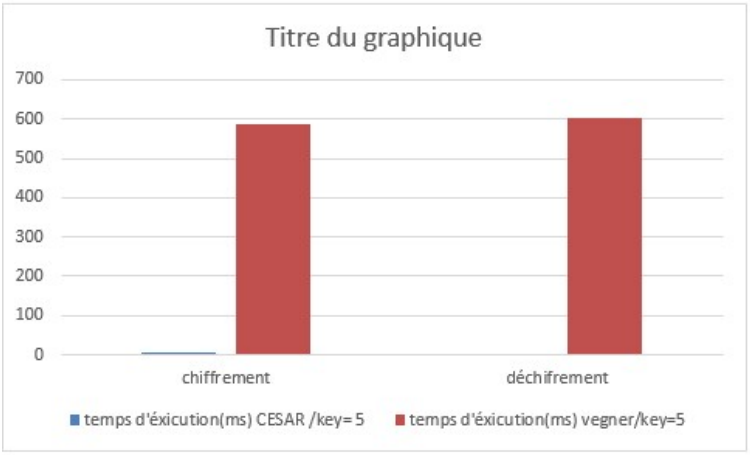


FIGURE 2.26: Histogramme6 : Comparaison temps chiffrement et déchiffrement de César Vs Vigener

Les résultats obtenus à partir (tableau 1 comparison SEASR et vgnére) montrent clairement que On note une convergence dans les résultats du chiffrement et du déchiffrement du message Pour le chiffrement sesar le temps nécessaire au déchiffrement est inférieur au temps de chiffrement du message. Quant à vegnér le temps de cryptage du message est inférieur à celui du décryptage.

2.24.1.1 conclusion

Nous concluons des résultats du tableau que RSA plus la taille de la clé est grande, plus le temps de déchiffrement et de chiffrement est long Contrairement à AES plus la taille de la clé est grande, plus le temps de chiffrement et de déchiffrement est long est inférieur à RSA. A travers le tableau, il devient clair pour nous, Cesar et Wegner, que le temps pour le cryptage et le décryptage est très court Voir Remplacer une lettre par une lettre dans le processus de cryptage et de décryptage.

2.24.2 l'espace mémoire

AES vs RSA

Résultat pour les algorithmes symétrique et asymétrique : le message chiffré (master 2 sic)

taille de clé	**128**	**192**	**256**	**512**	**1024**	**2048**
fonction	l'espace mémoire(bit) AES			l'espace mémoire(bit) RSA		
chiffrement	**48**	**62**	**65**	**113**	**1061**	**2001**
déchiffrement	37	37	37	37	37	37

TABLE 2.21: L'espace mémoir AES vs RSA

Espace mémoire pour le message« En général, la cryptographie est une technique d'écriture»

taille de clé	**128**	**192**	**256**	**512**	**1024**	**2048**
fonction	**l'espace mémoire(bit) AES**			**L'espace mémoire (bit) RSA**		
chiffrement	**128**	**158**	**158**	**261**	**1161**	**2333**
déchiffrement	**94**	**94**	**94**	**94**	**94**	**94**

TABLE 2.22: L'espace 2 mémoir AES vs RSA

Les résultats présentés dans le tableau montrent que plus la taille de la clé est grande, plus la consommation de mémoire pour RSA Quant à AES avec une clé 128 bits, il est petit et proche de l'espace consommé pour chacune des clés 129 bits et 256 bits. Mais en déchiffrement, on remarque le même espace car il revient à l'état d'origine du message

2.24.2.1 seasar vs vgner :

cryptographie classique :

	l'espace mémoire cesar(bit) (key=5)	**l"espace mémoire vgner (bit)(key=5)**
déchiffrement	**37**	**53**
chiffrement	**37**	**37**

TABLE 2.23: L'espace mémoir SEASAR vs VEGNER

2/Espace mémoire pour le message « En général, la cryptographie est une technique d'écriture»

	l'espace mémoire seasar(bit) key=5	**l'espace mémoire végnere(bit) key=5**
fonction	**96**	161
chiffrement	**96**	**96**

TABLE 2.24: L'espace 2 mémoir SEASAR vs VEGNER

Nous avons observé à travers les résultats de la tabulation ; L'espace consommé en déchiffrement et chiffrement reste constant par rapport à sesar, mais vegner on constate que ll'espace de décodage est supérieur au chiffrement du message, mais elle n'est pas grande.

2.25 conclusion

A travers les deux tableaux, il nous apparaît clairement que la surface consommée pour RSA est supérieure à AES. Et l'espace consommé par César est inférieur à celui de vegner

:

2.26 conclusion générale

Notre travail se résumé en étude et comparaison des principaux systèmes de cryptage et les technique y afférentes. A travers notre étude, nous avons vu un panel de méthodes de chiffrement de l'antiquité à nos jours,

Dans de nombreux environnements de communication modernes, y compris Internet, la majeure partie des données échangées est cryptée par l'algorithme AES rapide. Pour obtenir la clé secrète nécessaire au décryptage de ces données

Ainsi, la cryptographie est une science en perpétuelle évolution, la cryptanalyse l'aidant à trouver les failles d'un système pour toujours avancer. Cette évolution est importante car la cryptographie joue un grand rôle dans la sécurité internationale, tout étant aujourd'hui informatisé.
Dans notre projet, nous avons utilisé le python comme environnement de programmation.
En effet, malgré les grosses difficultés qu'on a rencontré, des quelles la non organisation et moins de l'expérience, le peu de temps consacré à cette étude, l'implémentation de notre projet fut l'une des majeures difficultés que nous avons rencontré, choix du langage de programmation plus efficace. Nous pouvons dire qu'on a très bien tiré profit de ce travail et qu'en toutes circonstances, on a appris beaucoup de choses : le travail en groupe, les technique et les compétences acquises dans le domaine du la cryptographie, la maîtrise et le savoir-faire dans les domaines de l'analyse et la conception, ainsi que beaucoup d'autres connaissances concernant le milieu de la sécurité.

Bibliographie

[1]. Cryptographie, libertés individuelles et codes secrets. Pierre Vigoureux. Ellipses 2016

[2]. La cryptologie, L'art des codes secrets. Philippe Guillot. EDP Sciences 2013

[3]. Initiation à la cryptographie, Cours et exercices corrigés. Gilles Dubertret. Vuibert 2012 (4eédition)

[4]. Exercices et problèmes de cryptographie , Licence 3, master, écoles d'ingénieurs. Damien Vergnaud. Dunod 2015

[5]. Cryptographie, théorie et pratique. Douglas Stinson. Vuibert 2003

[6] B. Martin, Codage, Cryptologie et application, 2001.

[7] G. Zennor, Cours de cryptographie, 2000.

[8]. Initiation à la cryptographie. Gilles Dubertret. Vuibert 2002

[9] M. Syahrizal, M. Murdani, S. D. Nasution, M. Mesran, R. Rahim, and A. P. U. Siahaan, "Modified Playfair Cipher Using Random Key Linear Congruent Method," J. Online Jar. COT POLIPD, vol. 10, no. 2, pp. 45–49, 2017.

[10]A. P. U. Siahaan, "Rail Fence Cryptography in Securing Information."

[11]. Cryptographie appliquée. Bruce Schneier. Vuibert 2001

[12]. Cryptographie en pratique. Niels Ferguson et Bruce Schneier. Vuibert 2004

[13]. Cryptographie et sécurité des systèmes et réseaux. Touradj Ebrahimi,Frank Leprévost et bertrand Warusfel. Lavoisier 2006

[14] A. Kahate, "Cryptography and network security", The Tata McGraw-Hill publishing company limited, New Delhi, (2003).La [15]

[15] Aleisa, N. (2015). A Comparison of the 3DES and AES

Encryption Standards. International Journal of Security and Its Applications, 9(7), 241-246

[16] Uskov, A., Byerly, A., & Heinemann, C. (2016). Advanced Encryption Standard Analysis with Multimedia Data on IntelÂ® AES-NI Architecture. IJCSA, 13(2), 89-105

[17] Cryptographie Nouvel Algorithme de Chiffrement Evolutionnaire basé Occurrences (ACEO).

[18] El Gamal, A public key cryptosystem and signature scheme based on discrete logarithms. IEEE Transactions on Information Theory IT-31, pp. 496-473,1976.

[19] S. Huard.Polarisation de la lumière, Editeur(s) : Elsevier-Masson.

[20] SACCO, Général L. Manuel de cryptographie, Payot (édition française), 1951.

[21] Christophe Bidan, Cryptographie et Cryptanalyse, Cours

[22] Cryptanalyses statistiques des algorithmes de chiffrement à clef secrète. (archives-ouvertes.fr)

[23] Canteaut02a.pdf (inria.fr).

[24] https://searchsecurity.techtarget.com/definition/Advanced-Encryption-
Standard

[25] Gilles Zémor, Cours de cryptographie, Paris, Cassini, 15 décembre2000, 227 .

[26] La Cryptographie militaire, L. Baudoin, 1883.

[27] Xuejia Lai. Higher Order Derivatives and Differential Cryptanalysis, pages227– 233. Springer US, Boston, MA, 1994.

[28] cours16 Auteur: Mme FADIA TALEB

[29] Daniel Genkin, Adi Shamir, and Eran Tromer. RSA Key Extraction via LowBandwidth Acoustic Cryptanalysis. In Juan A. Garay and Rosario Gennaro, editors, Advances in Cryptology - CRYPTO 2014 - 34th Annual Cryptology Conference, Santa Barbara, CA, USA, August 17-21, 2014, Proceedings, Part I, volume 8616 of Lecture Notes in Computer Science, pages 444–461. Springer,2014.

[30] A Comprehensive Evaluation of Cryptographic Algorithms: DES, 3DES, AES, RSA and Blowfish | Elsevier Enhanced Reader

[31] H. Delfs and H. Knebl, "Introduction to Cryptography: Principles and Applications "Springer- Verlag, Berlin Heidelberg New York, (2007)

[32] https://python.sdv.univ-paris-diderot.fr/
[33] https://docs.python.org/3.3/library/timeit.html
[34] http://docs.python.org/3.3/library/sys.html
[35] Mémoire de fin d'étude présenté en vue D'obtention du diplôme deMaster en Systèmes Informatiques Distribués
[36] Raymond, J.F. and Stiglic, A. (2000) Security Issues in the Diffie-Hellman Key Agreement Protocol. [Online] Available at [37]http://crypto.cs.mcgill.ca/~stiglic/publications.html [Accessed 17 March 2012].