



N° d'ordre :
N° de série :

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique



UNIVERSITE ECHAHID HAMMA LAKHDAR - EL OUED FACULTÉ DES
SCIENCES EXACTES Département D'Informatique

Mémoire de Fin D'étude
Présenté pour l'obtention du Diplôme de

MASTER ACADEMIQUE

Domaine : **Mathématique et Informatique**
Filière : **Informatique**
Spécialité : **Systèmes Distribués et Intelligence Artificielle**

Présenté par :

- **Ferik Brahim**
- **Guerfi Afif**

Thème

**Technique d'encryption efficace dédiée
pour une donnée volumineuse**

Soutenue le : 20/09/2020 devant le jury:

Dr	Hamoud Meriem	MCA	Président
Dr	Retima Farida	MAA	Rapporteur
Dr.	Laouid Abdelkader	MAA	Encadreur

Année Universitaire: 2019/2020



Dédicace

À l'exemple du dévouement et de la sincérité ... mon père bien-aimé.

À qui tu as offert mon bonheur et mon réconfort sur son bonheur ... ma mère généreuse.

À ceux qui représente le sérieux et la discipline ...mon second père Yahia

À ceux qui m'aident et me soutiennent chaque jour ... ma chère épouse.

À celui qui m'a donné espoir et patience ... mon amour Sidra et mon coeur Abdullah.

À celui qui représente le courage ... mon beau-frère Seghir

À tous ceux qui me souhaitent bonne chance

Je vous dédie ce travail

❖ **FERIK BRAHIM**



Dédicace

*Au symbole de l'amour et de la dévotion à **ma mère** bien-aimée*

*À celui qui m'a donné la bienveillance, le sacrifice et l'altruisme, **mon cher père***

*À l'exemple du don, de la fierté et du sacrifice, **mes frères et sœur***

*Je mentionne en particulier le poussin de la famille **Iyad***

À tous ceux qui m'aiment sincèrement et cordialement

Je vous dédie ce travail

❖ **GUERFI AFIF**

Remerciement

*Tout d'abord nous remercions notre Dieu qui nous a donné
la force et la volonté pour élaborer ce travail.*

Nous adressons nos vifs remerciements à notre encadreur

Dr. « LAOUID Abdelkader »

*Qui était chargé de fournir des conseils et des orientations
pour chaque petite et chaque grande dans ce mémoire*

*Nous adressons nos sincères remerciements à tous les enseignants du
département d'informatique*

Université Echahid Hamma Lakhdar

El-Oued.

*Aussi à nos collègues de la promotion 2019-2020. On
remercie également tous ceux qui ont participé de près ou
de loin à élaborer ce travail.*

Résumé

L'échange et le partage de données est essentiel sur l'Internet, donc le processus de sécurisation est également très nécessaire. La sécurité vidéo numérique, qui est considérée comme des importantes de données volumineuses, revêt une importance croissante avec l'utilisation généralisée des communications numériques.

De plus, la plupart des technologies des chiffrements ne sont pas très efficaces pour chiffrer la vidéo numérique. Nous visons à proposer une nouvelle méthode de chiffrement vidéo, basée sur l'hybridation entre deux algorithmes de cryptage symétrique et asymétrique, où le premier algorithme est XOR pour chiffrer la vidéo et le second algorithme est Diffie-Hellman pour échanger la clé partagée Il sert à remplir la POOL pour générer les clés de chiffremnt.

Nous avons appliqué notre approche et obtenu de très bons résultats expérimentaux, elle est considérée comme plus sécurisée.

Mots clés : chiffrement vidéo, XOR,POOL,Diffie-Hellman, image, pixel.

ملخص

يعد تبادل البيانات و مشاركتها عبر الانترنت أمر أساسي فعلية تأمينها ضرورة ملحة. أمان الفيديو الرقمي الذي يعتبر من البيانات الكبيرة التي لها أهمية متزايدة مع الانتشار الواسع في استخدام الاتصالات الرقمية.

كما أن معظم تقنيات التشفير الموجودة ليست ذات كفاءة كبيرة في تشفير الفيديو الرقمي، في مذكرتنا هذه إقترحنا طريقة جديدة في تشفير الفيديو، إستنادا إلى التهجين بين خوارزميتي تشفير متماثلة وغير متماثلة، حيث الخوارزمية الأولى هي XOR لتشفير الفيديو والخوارزمية الثانية هي Diffie-Hellman لتبادل المفتاح المشترك الذي بدوره يستخدم في ملئ الـ POOL لتوليد مفاتيح التشفير.

بالإضافة إلى ذلك، قمنا بتطبيق هاته الطريقة وتحصلنا على نتائج تجريبية جد حسنة التي تعتبر أكثر أمانا.

الكلمات المفتاحية: تشفير الفيديو، XOR، POOL، Diffie-Hellman، صورة، بيكسل.

Abstract

The exchange and sharing of data is essential on the Internet, so the process of securing is also very necessary. Digital video security, which is regarded as important to voluminous data, is gaining increasing importance with the widespread use of digital communications.

Additionally, most encryption technologies are not very efficient at encrypting digital video. On this thesis, we proposed a new method for encrypting video using security techniques will be studied, based on hybridization between two encryption algorithms symmetric and asymmetric, where the first algorithm is XOR for encryption the video and the second algorithm is Diffie-Hellman for exchange the shared key which is used for filling the POOL for generating encryption keys.

In addition, we implemented our approach and obtained very good experimental results which are considered our approach is more secure.

Key words : video encryption, Diffie-Hellman, POOL, XOR, Frame, pixel.

TABLE DES MATIÈRES

Table des matières	i
Table des figures	iv
Liste des tableaux	vi
Introduction générale	1
1 Généralité sur la cryptographie	3
1.1 Introduction	4
1.2 Vue historique	4
1.3 Notions de cryptologie	5
1.4 Terminologie	6
1.5 Attaques sur un chiffrement	6
1.6 Les buts de la cryptographie	7
1.7 L'algorithme cryptographique	7
1.7.1 Principes de Kerckhoffs en cryptographie	8
1.7.2 Principe de Shannon	8
1.7.3 La publication des algorithmes	8
1.7.4 Description formelle d'un algorithme cryptographique	9
1.8 Les types de la cryptographie	10
1.8.1 La cryptographie classique	10
1.8.2 La cryptographie modern	11
1.9 La cryptographie visuelle	15

1.10	Stéganographie	16
1.10.1	La stéganographie linguistique	16
1.10.2	La stéganographie technique	16
1.11	Conclusion	17
2	État de l'art	18
2.1	Introduction	19
2.2	Vidéo numérique	19
2.3	Conteneurs vidéo	19
2.4	Les espaces de couleurs	20
2.4.1	Image noire et blanc	20
2.4.2	Image niveau de gris	20
2.4.3	Image couleur (RGB)	21
2.5	Compression vidéo numérique	22
2.5.1	Compression sans perte	22
2.5.2	Compression avec perte	22
2.6	Classification des algorithmes de cryptage de vidéo numérique	22
2.6.1	Cryptage total	22
2.6.2	Cryptage basé sur la permutation	23
2.6.3	Chiffrement sélectif	23
2.6.4	Cryptage perceptif	23
2.7	Les travaux connexes	23
2.7.1	Approche basée sur l'hybridation	24
2.7.2	Approche Chaotique	25
2.7.3	Approche de cryptage du boîtier	26
2.7.4	Approche Moderne	27
2.7.5	Approche basée sur le RSA et PN (Pseudo Noise)	28
2.8	Conclusion	29
3	L'approche proposée	30
3.1	Introduction	31
3.2	Schéma globale	31
3.3	La création de Pool :	32
3.3.1	Processus de partage de la clé :	32

3.3.2	Remplissage de Pool	34
3.3.3	Générer les clés de chiffrement et déchiffrement	35
3.4	Les processus de chiffrement et de déchiffrement	36
3.4.1	Processus de chiffrement	36
3.4.2	Processus de déchiffrement	37
3.4.3	Description de chiffrement et déchiffrement d'image	37
3.5	Conclusion	39
4	Implémentation et Analyses	40
4.1	Introduction	41
4.2	Environnement de développement	41
4.2.1	Environnement logiciel	41
4.2.2	Environnement matériel	42
4.3	Critères d'évaluation	42
4.3.1	Espace de clés	42
4.3.2	L'histogramme	42
4.3.3	Mesure de distorsion (PSNR)	47
4.3.4	Mesure de similarité structurelle (SSIM)	47
4.3.5	L'entropie	48
4.3.6	La corrélation entre les pixels adjacents	49
4.4	Discussion	52
4.5	Conclusion	53
	Conclusion générale	54
	Bibliographie	55

TABLE DES FIGURES

1.1	Schéma global d'un algorithme cryptographique.	10
1.2	Schéma illustre l'encryption symétrique.	11
1.3	Schéma illustre l'encryption asymétrique.	12
1.4	Schéma illustre le chiffrement de flux.	13
1.5	Schéma illustre le chiffrement de bloc.	14
1.6	Schéma illustre l'encryption hybride.	15
1.7	Technique de stéganographie.	16
2.1	Représentation numérique d'une image noir et blanc.	20
2.2	Représentation numérique d'une image niveau de gris.	20
2.3	Représentation numérique d'une image en couleur.	21
2.4	Schéma illustre le processus de chiffrement d'un vidéo utilisé AES et NTRU. . .	24
2.5	Schéma illustre le processus de chiffrement une image utilisé XOR.	25
2.6	Schéma illustre le processus d'encryption et de compression vidéo.	26
2.7	Schéma illustre le processus de cryptage de la vidéo utilisé Iyer et all.	27
2.8	Schéma illustre le processus de décryptage de la vidéo.	28
2.9	Schéma illustre le cryptage et décryptage de vidéo utilisé RSA et PN.	28
3.1	Schéma global de notre proposition.	31
3.2	Le processus de Deffie Hellman	33
3.3	Schéma illustre le remplissage de Pool.	35
3.4	Schéma illustre comment extraire la clé.	36
3.5	Processus de chiffrement une video.	37
3.6	Processus de déchiffrement une video.	37

3.7	Extraire la clé de chiffrement et déchiffrement du Pool.	38
4.1	Une image clair de début de la vidéo.	43
4.2	Une image clair de centre de la vidéo.	43
4.3	Une image clair de finale de la vidéo.	43
4.4	Histogramme d'une image clair de début de la vidéo.	44
4.5	Histogramme d'une image clair de centre de la vidéo.	44
4.6	Histogramme d'une image clair de finale de la vidéo.	44
4.7	Une image chiffrée de début de la vidéo.	45
4.8	Une image chiffrée de centre de la vidéo.	45
4.9	Une image chiffrée de finale de la vidéo.	45
4.10	Histogramme d'une image chiffrée de début de la vidéo.	46
4.11	Histogramme d'une image chiffrée de centre de la vidéo.	46
4.12	Histogramme d'une image chiffrée de finale de la vidéo.	46
4.13	Les distributions des pixels adjacents sur les images originales.	51
4.14	Les distributions des pixels adjacents sur les images cryptées.	52

LISTE DES TABLEAUX

2.1	Représentation numérique d'un pixel.	21
2.2	La forme binaire d'un pixel.	21
3.1	Chiffrement d'un pixel.	38
3.2	Déchiffrement d'un pixel.	39
4.1	Les valeurs de l'entropie des images claires et leurs chiffrées.	49
4.2	Les valeurs de La corrélation des images claires et leurs chiffrées.	50
4.3	Les analyses des vidéos avant et après le chiffrement.	52

INTRODUCTION GÉNÉRALE

Actuellement, avec le progrès rapide de la technologie des réseaux notamment l'internet, les données volumineuses tel que les vidéos numériques ont un énorme type d'information impliquées dans les communications modernes.

Les vidéos numériques sont largement utilisées dans plusieurs domaines sensibles tels que systèmes de visioconférence, le commerce électronique et les diffusions vidéos ... etc, alors le rythme des progrès est donc très rapide dans les domaines de la communication vidéo et audio, grâce à cela est devenu la sécurité des données est une nécessité importante que ce soit pour le stockage ou pour la transmission et les technologies d'échange de données sont également évolué. La cryptographie est parmi les méthodes les plus efficaces pour établir la confidentialité et l'intégrité de ce type d'information.

Il est devenu clair que nous ne pouvons pas utiliser les méthodes de chiffrement standard comme RSA, DES, AES, pour le chiffrement de vidéo numériques, ces types nécessitent beaucoup de temps pour effectuer des processus compliqués ou, ils sont considérés ne fortement sécurisé pour ce type de données. Ainsi les vidéos numériques sont caractérisées par la redondance élevée, la forte corrélation et la taille volumineuse. Le problème posé est comment peut-on concevoir un système de cryptage pour assurer la sécurité de ce type de données et aussi lèges d'encryptions ?

Le contenu de ce mémoire est organisé comme suit :

Dans **le premier chapitre**, nous décrivons des généralités sur la cryptographie où nous avons parlé à principaux aspects et la terminologie qui y sont liés, on abordera également les algorithmes de chiffrement symétrique, asymétrique et hybride en citant quelques algorithmes classiques et populaires, enfin nous avons montré le rôle et le statut de la cryptographie.

Dans **le deuxième chapitre**, nous parlerons des bases de la vidéo numérique et de certaines méthodes modernes de chiffrement de la vidéo numérique et expliquerons les principes de leur fonctionnement, puisque la vidéo n'est qu'un défilement d'une séquence d'images animées, nous décrirons un aperçu des images numériques, de leurs composantes et de leurs types. et enfin nous allons parler sur les derniers recherches dans le domaine de cryptage vidéo.

Dans **le troisième chapitre**, nous allons proposer une autre nouvelle approche de chiffrement des données volumineuses s'appliquant méthode hybride qui compose par une méthode symétrique (algorithme XOR pour chiffrer la vidéo) et asymétrique (algorithme Diffie-Helman pour génère clé partagé) en outre nouvelle technique le Pool qui contient ensemble de clé.

Dans **le dernier chapitre**, nous entamons maintenant sur ce dernier chapitre la partie de la réalisation et de l'analyse, nous présentons notre application réalisée et les différentes mesures d'évaluation pour montrer la sécurité et l'efficacité de l'algorithme proposé. Puis on va terminer par une conclusion générale et quelques perspectives pouvant aider dans l'amélioration du système dans le futur.

CHAPITRE 1

GÉNÉRALITÉ SUR LA CRYPTOGRAPHIE

1.1 Introduction

Le chiffrement est la principale application de la cryptographie ; ça fait des données incompréhensible afin d'assurer sa confidentialité. Le chiffrement utilise une méthode appelée l'algorithme de chiffrement et valeur secrète appelée clé ; si vous ne le faites pas connaître la clé secrète, vous ne pouvez pas déchiffrer, ni apprendre un peu des informations sur le message crypté et aucun attaquant non plus.

Ce chapitre se concentrera sur l'introduction aux généralités sur la cryptographie, après un rapide historique de la cryptographie on examinera les principaux systèmes cryptographiques modernes utilisés pour la transmission et le stockage sécurisé de données. On ne s'intéresse qu'aux systèmes cryptographiques destinés à transmettre des flux importants et variés d'informations (paiement sécurisé par internet, données bancaires, cartes de crédit, protection des conversations entre téléphones mobile, WiFi,...) entre de nombreux interlocuteurs qui nécessitent des systèmes cryptographiques structurés et rapides.

1.2 Vue historique

Les communications ont de tout temps constitué un aspect important dans l'acquisition de nouvelles connaissances et l'essor de l'humanité. Le besoin d'être en mesure d'envoyer un message de manière sécurisé est probablement aussi ancien que les communications elles-mêmes. D'un point de vue historique, c'est lors des conflits entre nations que ce besoin a été le plus vif, pour assurer la confidentialité d'une partie de leur portée.

L'histoire retient Jules César comme précurseur et utilisateur régulier de la cryptographie pour envoyer des messages confidentiels s'ont de tout temps protégé d'éventuels messagers malhonnêtes.

Et, bien avant, entre le Xe et VIIe siècle av. J.-C. Les Grecs utilisaient déjà une technique de chiffrement par transposition qui permettait de modifier la disposition des lettres dans un message. Ils se servaient d'une scytale. Le scytale était un ruban (un cylindre) sur lequel on enroulait une bandelette de papier, le texte était alors écrit en lignes droites successives, le ruban qui constitue le texte chiffré était ensuite déroulé et expédié au destinataire qui devait le ré-enrouler sur un axe de même diamètre, le texte apparait donc en clair, sans la connaissance du diamètre du ruban qui jouait le rôle de la clé, il était impossible de déchiffrer le message.

Puis, au cours du 19ième siècle, on assista au développement plus au moins ingénieux de techniques de chiffrement expérimentales. Kirchhoff posa les principes de la cryptographie

moderne, l'un des principaux, pose que la sécurité d'un système de chiffrement ne résidait que dans la clé et non dans le procédé de chiffrement. Une dernière innovation, plus récente, fût l'utilisation de la machine "Enigma" par l'armée Allemande, durant la Seconde Guerre Mondiale. Cette machine permettait de chiffrer toutes les communications radio ou télégraphiques. Le système était simple mais efficace. Chaque lettre est substituée par une autre, avec des résultats qui changent à chaque fois. La machine est alimentée par une pile électrique. Elle contient un mécanisme de rotors qui modifie le circuit électrique à chaque frappe. Lorsqu'un utilisateur appuie sur une touche du clavier, un circuit électrique se ferme et une lampe s'allume en indiquant le caractère à utiliser.

A partir de 1970 La cryptographie, à connu une réelle explosion avec le développement des systèmes informatiques. Elle a connu un plus large progrès avec l'arrivée des systèmes d'informations modernes où il y a une nécessité de protéger les données [1].

1.3 Notions de cryptologie

La cryptographie appartient à la cryptologie du grec *kryptos* « secret, caché » et *logos* « discours », qui est la science de l'écriture secrète englobant des pratiques concurrentes à savoir la cryptographie, le déchiffrement et la cryptanalyse. La première pratique qui est la cryptographie (du grec *kryptos* et *graphein*) est : « la discipline incluant les principes, les moyens et les méthodes de transformation des données, dans le but de masquer leur contenu, empêcher leur modification ou leur utilisation illégale, ainsi que les opérations inverses, pour rendre le document à nouveau intelligible ». Le déchiffrement est le processus permettant de transformer le message chiffré en message clair. Quand à la cryptanalyse, qui est un terme créé par le cryptologue américain William Friedman en 1920 (du grec *kryptos* et *analisis* « résolution, dissolution »), est l'art de décoder un message chiffré en mêlant une intéressante combinaison de raisonnement analytique, d'application d'outils mathématiques, de découverte de redondances, de patience, de détermination, et de chance [2].

Les deux disciplines de cryptographie et de cryptanalyse s'alimentent l'une l'autre. On ne peut pas évaluer la sécurité d'un mécanisme sans le soumettre à des attaques qui, à leur tour, conduisent à des critères de conception pour rendre les procédés plus sûrs. Ces derniers seront à nouveau passés au crible du cryptanalyste.

1.4 Terminologie

Il y a plusieurs de terminologie de la cryptographie, notamment [3] :

✓ **Texte en clair** : c'est une information non chiffrée.

✓ **Texte chiffré** : appelé également cryptogramme, c'est le résultat du chiffrement du texte en clair.

✓ **Chiffrement** : c'est la méthode ou l'algorithme utilisé pour transformer un texte en clair en texte chiffré.

✓ **Déchiffrement** : c'est la méthode ou l'algorithme utilisé pour transformer un texte chiffré en texte en clair.

✓ **Clé** : Il s'agit du paramètre impliqué et autorisant des opérations de chiffrement et/ou déchiffrement. Dans le cas d'un algorithme symétrique, la clef est identique lors des deux opérations. Dans le cas d'algorithmes asymétriques, elle diffère pour les deux opérations.

✓ **Cryptographie** : cette branche regroupe l'ensemble des méthodes qui permettent de chiffrer et de déchiffrer un texte en clair afin de le rendre incompréhensible pour quiconque n'est pas en possession de la clé à utiliser pour le déchiffrer.

✓ **Cryptanalyse** : c'est l'art de révéler les textes en clair qui ont fait l'objet d'un chiffrement sans connaître la clé utilisée pour chiffrer le texte en clair.

✓ **Cryptologie** : il s'agit de la science qui étudie les communications secrètes. Elle est composée de deux domaines d'étude complémentaires : la cryptographie et la cryptanalyse.

✓ **Coder, décoder** : c'est une méthode ou un algorithme permettant de modifier la mise en forme d'un message sans introduire d'élément secret.

1.5 Attaques sur un chiffrement

Il existe plusieurs types d'attaques. Par exemple dans un cryptosystème basé sur des registres à décalage et on suppose que l'attaquant connaît la forme des récurrences linéaires ainsi que la fonction de combinaison mais pas les conditions initiales des récurrences qui constituent la clef du code.

On doit distinguer entre les types d'attaques d'un adversaire et les buts des attaques d'un adversaire. On distingue habituellement quatre méthodes d'attaque [4] :

- Une **attaque sur texte chiffré seulement** consiste à retrouver la clé de déchiffrement à partir d'un ou plusieurs textes chiffrés.
- Une **attaque sur texte clair connu** consiste à retrouver la clé de déchiffrement à partir

d'un ou plusieurs textes chiffrés, connaissant le texte en clair correspondant.

- Une **attaque sur texte clair** choisi consiste à retrouver la clé de déchiffrement à partir d'un ou plusieurs textes chiffrés, l'attaquant ayant la possibilité de les générer à partir de textes en clair.

- Une **attaque sur texte chiffré** choisi consiste à retrouver la clé de déchiffrement à partir d'un ou plusieurs textes chiffrés, l'attaquant ayant la possibilité de les générer à partir de textes en clair.

Il y a aussi des attaques physiques. Le principe de ces attaques est d'essayer de reconstituer la clef secrète par exemple en écouter la transmission entre le clavier de l'ordinateur et l'unité centrale ou en mesurant la consommation électrique du microprocesseur qui effectue le décodage du message ou encore en mesurant son échauffement.

1.6 Les buts de la cryptographie

L'utilisation aujourd'hui de la cryptographie a un intérêt non seulement à préserver la confidentialité des données mais aussi à garantir leur intégrité et leur authenticité.

La cryptographie permet de résoudre quatre importants des problèmes différents [5] :

- ✓ **La confidentialité** : consiste à rendre l'information inintelligible protéger le contenu des informations sauvegardées ou transmises à d'autres personnes que les seuls acteurs de la transaction.

- ✓ **L'intégrité** : vérifier l'intégrité des données consiste à déterminer si les données n'ont pas été modifiées durant la communication (de manière accidentellement ou volontairement).

- ✓ **L'authentification** : consiste à assurer l'identité d'un utilisateur, c'est-à-dire de garantir à chacun des correspondants que son partenaire est bien celui qu'il croit être. Un contrôle d'accès peut permettre (par exemple par le moyen d'un mot de passe qui devra être crypté) l'accès à des ressources uniquement aux personnes autorisées.

- ✓ **La non-répudiation** : de l'information est la garantie qu'aucun des correspondants ne pourra nier la transaction.

1.7 L'algorithme cryptographique

Chaque algorithme de cryptographie il doit être contenir les deux principes de kerckhoffs et de shannon.

1.7.1 Principes de Kerckhoffs en cryptographie

Un système cryptographique dont la façon dont cela fonctionne est connue et librement diffusés et qui résiste aux attaques continues de tous les cryptanalystes pourra être considéré comme un système sûr [6].

Le premier à avoir formalisé ce principe est Auguste Kerckhoffs en 1883 dans l'article "La cryptographie militaire" paru dans le Journal des Sciences Militaires. Son article comporte en réalité six principes, connus depuis, sous le nom de Principes de Kerckhoffs. On en résumera ici que trois, les plus utiles aujourd'hui [6] :

1. La sécurité repose sur le secret de la clé et non sur le secret de l'algorithme.
2. Le déchiffrement sans la clé doit être impossible (en temps raisonnable).
3. Trouver la clé à partir du clair et du chiffré est impossible (en temps raisonnable).

1.7.2 Principe de Shannon

Pas moins important que le principe précédent, Shannon énonça que pour effacer les redondances dans un texte en clair, deux techniques plus importants s'imposaient Confusion et Diffusion [7].

- **Confusion**

La confusion correspond à une souhaité de rendre la relation entre la clé de chiffrement et le texte chiffré la plus complexe possible.

- **Diffusion**

La diffusion est une propriété où la redondance statistique dans un texte en clair est dissipée dans les statistiques du texte chiffré. En d'autres termes, un biais en entrée ne doit pas se retrouver en sortie et les statistiques de la sortie doivent donner le moins possible d'informations sur l'entrée.

1.7.3 La publication des algorithmes

Selon l'endroit où réside le secret, on peut parler d'algorithme secret ou d'algorithme publié (dans lequel le secret réside dans la clé), Chacun possède ses avantages et ses inconvénients.

Algorithme secret La cryptanalyse, dans ce type, doit ici en plus retrouver l'entièreté de l'algorithme (mécanisme de récupération), Souvent, ce type est utilisé par un nombre limité d'utilisateurs. De tels algorithmes sont généralement non distribué pour un usage exclusif.

Algorithme publié Puisque l'algorithme est publié, tout le monde a le droit de l'explorer. Ainsi, les failles (laissées intentionnellement ou non par les concepteurs) peuvent être plus

facilement découvertes, la sécurité est donc améliorée, comme la publication est autorisée, il n'est pas nécessaire de chercher à protéger le code contre le reverse-engineering, cette publication permet d'étendre les travaux sur l'algorithme au niveau mondial, tout le monde utilise la même version publique ce qui permet une standardisation générale [8].

1.7.4 Description formelle d'un algorithme cryptographique

D'une manière formelle, un cryptosystème est un quintuplet (P, C, K, E, D) satisfaisant les points suivants [9] :

1. P est un ensemble fini de blocs de textes clairs possibles.
2. C est un ensemble fini de blocs de textes chiffrés possibles.
3. K est un ensemble fini de clefs possibles.
4. Pour tout $k \in K$, il y a une règle de chiffrement $e_k \in E$ et une règle de déchiffrement correspondant $d_k \in D$. Chaque $e_k : P \rightarrow C$ et $d_k : C \rightarrow P$ sont des fonctions telles que $d_k(e_k(x)) = x$ pour tout texte clair $x \in P$.

La principale propriété est la quatrième. Elle précise que si un texte clair x est chiffré en utilisant e_k , et si le texte chiffré y obtenu est ensuite déchiffré en utilisant d_k , on retrouve le texte clair x original.

Alice et Bob peuvent employer le protocole suivant pour utiliser un cryptosystème spécifique. Tout d'abord, ils choisissent une clé quelconque $k \in K$. Cette opération est effectuée lorsqu'ils se rencontrent en un même endroit loin d'être observés par Oscar, ou bien à travers un canal de communication sûr. Supposant qu'ensuite, Alice souhaite communiquer un message à Bob par un canal peu sûr, ce message étant une chaîne : $x = x_1 x_2 x_3 \dots x_n$ avec : $n \in \mathbb{Z}, n \geq 1$, $x_i \in P$ et $1 \leq i \leq n$.

Chaque bloc x_i est chiffré en utilisant la règle de chiffrement e_k spécifiée par la clé k choisie. Ainsi, Alice calcule $y_i = e_k(x_i)$, $1 \leq i \leq n$, et la chaîne chiffrée obtenue sera : $y = y_1 y_2 \dots y_n$

Cette chaîne est envoyée dans le canal et une fois reçue par Bob, il la déchiffre en utilisant la fonction de déchiffrement d_k pour récupérer le texte clair original $x_1 x_2 \dots x_n$. Le procédé de communication est illustré sur la figure 1.

Il est évident que chaque fonction de chiffrement e_k doit être injective (c'est à dire ne pas chiffrer deux blocs différents en deux valeurs égales), sinon, le procédé de déchiffrement ne pourrait être fait sans ambiguïté. Plus précisément, si $y = e_k(x_1) = e_k(x_2)$

Avec : $x_1 \neq x_2$, Bob n'a aucun moyen de savoir si y doit être déchiffré en x_1 ou en x_2 .

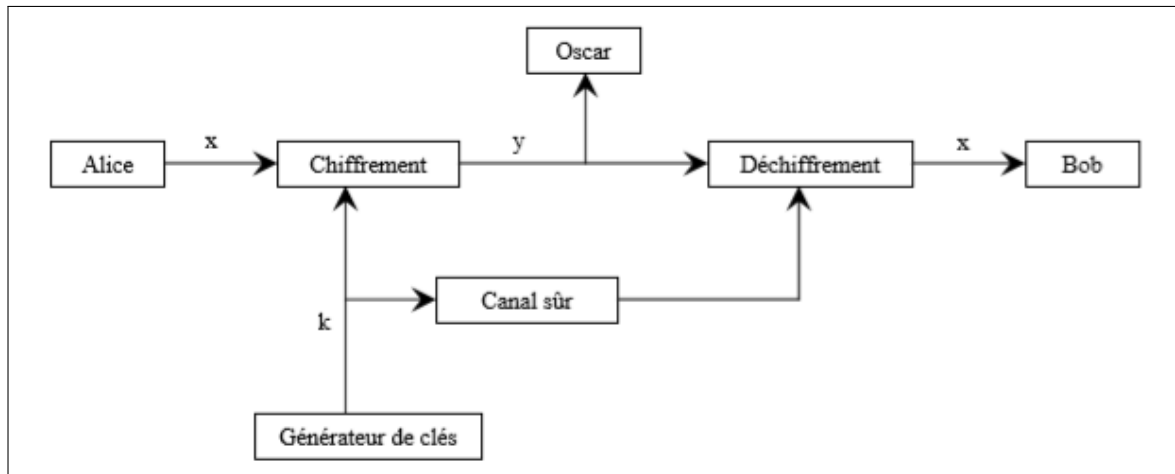


FIGURE 1.1 – Schéma global d'un algorithme cryptographique.

1.8 Les types de la cryptographie

Les types de la cryptographie est divisé en deux types, classique et moderne.

1.8.1 La cryptographie classique

La cryptographie classique décrit la période avant les ordinateurs durant laquelle, il est limité uniquement aux lettres de l'alphabetles principaux outils utilisés consistent à remplacer des caractères par d'autres ou les transposer dans des ordres différents tout en gardant secrètes les procédures de chiffrement ou de déchiffrement. Sans cela le système est complètement inefficace, puisque n'importe qui peut déchiffrer le message codé. On appelle généralement cette classe de méthodes : le chiffrement à usage restreint [10].

- **Cryptographie par substitution**

Dans ce mode de cryptage, les lettres du message en clair sont remplacées par d'autres lettres, des chiffres ou d'autres symboles. Selon la façon de substituer, on distingue la substitution mono-alphabétique, la substitution homophonique et la substitution poly-alphabétique.

- **Cryptographie par transposition**

Les méthodes de cryptographie par transposition sont celles pour lesquelles on chiffre le message en permutant l'ordre des lettres du message suivant des règles bien définies. Autrement dit, on produit un anagramme du message initial., ce qui permet de mieux cacher le message. Plusieurs types de transposition existent : Transposition simple par colonnes, transposition complexe par colonnes, transposition par carré polybique.

1.8.2 La cryptographie modern

La cryptographie modern est divisé en deux parties selon le type de clé (symétrique / asymétrique) et le type de flux de données (flot / bloc).

- **Le type de clé (symétrique/ asymétrique)**

Les algorithmes de chiffement à clé secrète (ou symétriques ou encore dits conventionnels) sont ceux pour lesquels l'émetteur et le destinataire partagent une même clé secrète, autrement dit, les clefs de chiffement et de déchiffement sont identiques. L'emploi d'un algorithme symétrique lors d'une communication nécessite donc l'échange préalable d'un secret entre les deux parties importantes à travers un canal sécurisé ou au moyen d'autres techniques cryptographiques.

Un paramètre essentiel pour la sécurité d'un système à clé secrète est la taille de l'espace des clefs. En effet, il est toujours possible de mener sur un algorithme de chiffement, une attaque dite exhaustive pour retrouver la clé. Cette attaque consiste simplement à énumérer toutes les clefs possibles du système et à essayer d'utiliser chacune d'entre elles pour décrypter un message chiffré. Si l'espace des clefs correspond à l'ensemble des mots de k bits, le nombre de tentatives d'attaque exhaustive en vue de décrypter le message chiffré est égal à 2^{k-1} . Donc, pour pénaliser une telle attaque, il faut que l'espace des clés soit suffisamment grand [11].

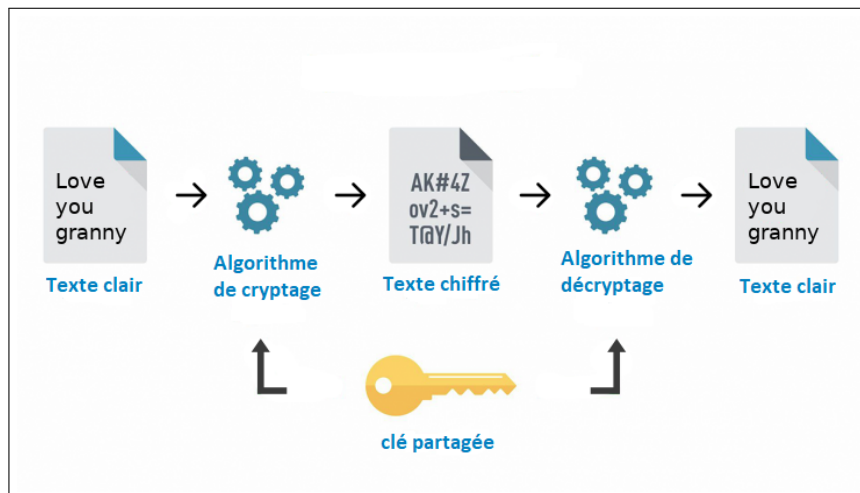


FIGURE 1.2 – Schéma illustre l'encryption symétrique.

Les exemples d'algorithmes symétriques : DES (Data Encryption Standard) : utilise une clé secrète de 56 bits, la taille des blocs est de 64 bits. 3DES (Triple DES) : utilise une clé de taille comprise entre 128 et 192 bits. La taille des blocs est de 8 octets (64 bits). AES (Advanced Encryption Standard) : il travaille avec des blocs de 128 bits et il utilise des clés de 56 bits seulement.

La cryptographie à clé publique évite le partage d'un secret entre les deux interlocuteurs puisque, chaque utilisateur dispose d'un couple de clés : une clé publique qu'il met en général à disposition de tous dans un annuaire, et une clef secrète connue de lui seul. Ces deux clés, en plus d'être distinctes, elles ne peuvent se déduire l'une de l'autre. Alors, pour envoyer un message confidentiel à Bob, Alice chiffre le message clair à l'aide de la clé publique de Bob. Ce dernier, à l'aide de la clé secrète correspondante, sera le seul en mesure de déchiffrer le message reçu.

La notion primordiale sur laquelle repose le chiffrement à clé publique est celle de fonction à sens unique avec trappe. Sachons qu'une fonction est appelée à sens unique si elle est aisément calculée, mais difficile à inverser, ou plus exactement, infaisable en un temps réalisable avec une puissance de calcul raisonnable. Et une telle fonction sera dite à trappe, si le calcul de l'inverse devient facile dès que l'on possède une information supplémentaire qui est la trappe. Donc, la construction d'un système de chiffrement à clé publique à partir d'une telle fonction, sera une chose très simple où la procédure de chiffrement consiste simplement à appliquer la fonction au message clair. L'utilité d'utilisation d'une telle fonction, difficile à inverser si on ne connaît pas la trappe, dans le domaine de la cryptographie, réside dans le fait de rendre difficile la détermination du message en clair à partir du message chiffré sans connaître la clé secrète de déchiffrement. Cependant, la définition de ces fonctions particulières n'est pas aussi facile puisqu'elle s'appuie généralement sur des problèmes mathématiques réputés difficiles tel que le problème de la factorisation de grands nombres entiers. Mais, si quelqu'un trouve un jour le moyen de simplifier la résolution d'un de ces problèmes, l'algorithme correspondant, c'est à dire construit autour de ce problème, finira par être abandonné [11].

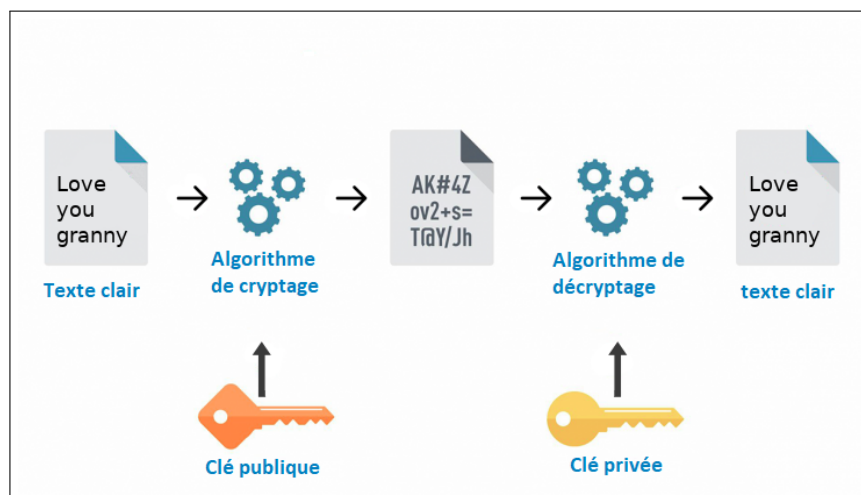


FIGURE 1.3 – Schéma illustre l'encryption asymétrique.

Les exemples d'algorithmes asymétriques : Diffie Hellman, RSA (Rivest, Shamir, Adelman), El Gamal.

À propos d'une définition de Diffie-Hellman est un algorithme mathématique qui permet à deux PC de produire un secret partagé identique sur les deux systèmes, malgré le fait que ces systèmes n'aient jamais pu communiquer entre eux. Ce secret partagé peut ensuite être utilisé pour échanger en toute sécurité une clé de chiffrement cryptographique. Cette clé crypte ensuite le trafic entre les deux systèmes [12].

- **le type de flux de données (flot / bloc)**

Cette type est concerné par de flux de données flot ou bloc [13].

- **Chiffrement par flot**

L'idée de base d'un chiffrement par flot est de diviser le texte en petits blocs, un bit ou un octet, et de coder chaque bloc en fonction de nombreux blocs précédents. Chiffrement par flot utilisent une clé de chiffrement différente, une valeur qui doit être introduite dans l'algorithme, pour chaque bit ou octet, de sorte que le même bit ou octet produit un texte chiffré différent chaque fois qu'il est chiffré.

Certains algorithmes de chiffrement par flot utilisent un générateur de flux de clés, qui produit un flux de bits aléatoire ou presque aléatoire. Le chiffrement effectue une opération booléenne, connue sous le nom de OU exclusif, entre la clé et le texte en clair pour produire un texte chiffré..

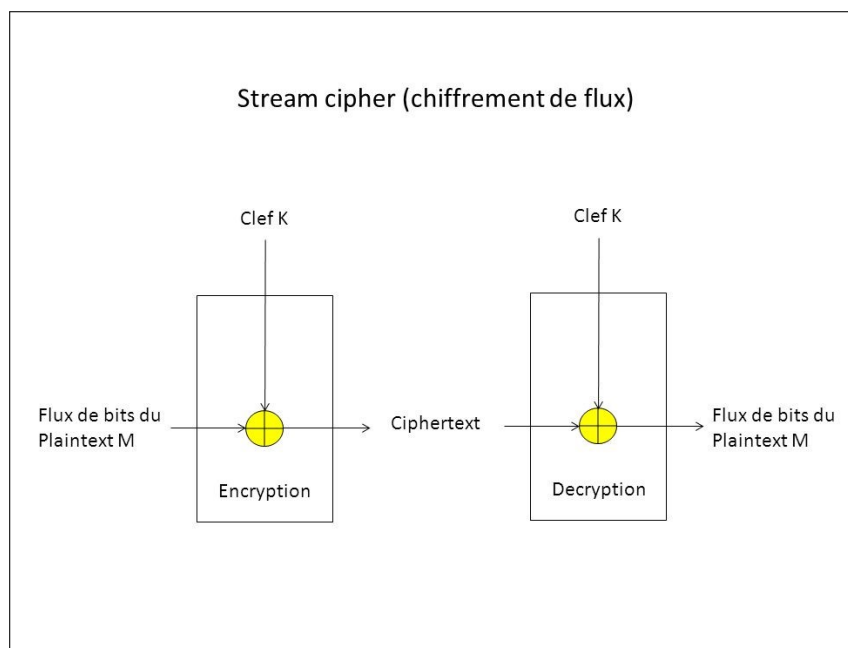


FIGURE 1.4 – Schéma illustre le chiffrement de flux.

► Chiffrement par bloc

L'idée de base d'un chiffrement par bloc est de diviser le texte en blocs relativement gros, typiquement de 64 ou 128 bits, et de coder chaque bloc séparément. La même clé de chiffrement est utilisée pour chaque bloc et c'est la clé de chiffrement qui détermine l'ordre dans lequel la substitution, le transport et d'autres fonctions mathématiques sont effectuées sur chaque bloc.

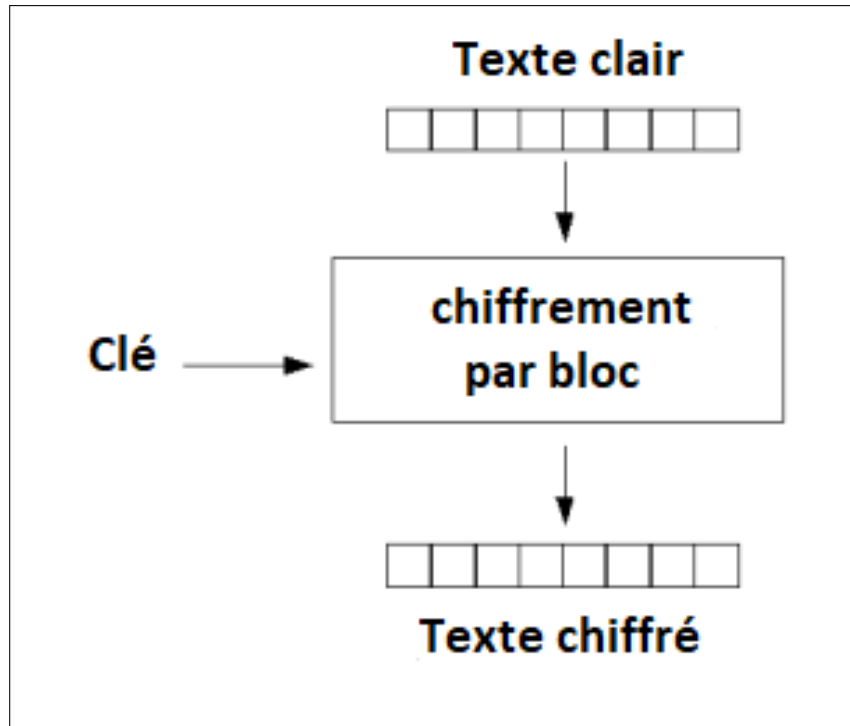


FIGURE 1.5 – Schéma illustre le chiffrement de bloc.

► Différences clés entre le Chiffrement par flot et le Chiffrement par bloc

Le chiffrement par bloc utilise à la fois la confusion et la diffusion tandis que le chiffrement par flot ne repose que sur la confusion.

La taille habituelle du bloc peut être de 64 ou 128 bits dans le chiffrement par bloc. Par contre, 1 octet (8 bits) dans le chiffement par flot.

Le chiffement par bloc utilise les modes d'algorithme ECB (Electronic Code Book) et CBC (Cipher Block Chaining). Au contraire, le chiffement par flot utilise les modes d'algorithme CFB (Cipher Feedback) et OFB (Output Feedback).

Chiffement par flot utilise la fonction XOR pour convertir le texte brut en texte chiffré, c'est la raison pour laquelle il est facile d'inverser les bits de l'XOR. Alors que le chiffement par bloc n'utilise pas l'XOR pour ce faire.

Le chiffement par bloc utilise la même clé pour chiffrer chaque bloc tandis que le chiffement par flot utilise une clé différente pour chaque octet [13].

- **Hybride**

La cryptographie hybride consiste en une combinaison des deux techniques de cryptage symétrique et asymétrique où on code tout d'abord les données avec une clé privée dite clé de session, puis cette clé est cryptée à l'aide d'une clé publique. Dans cette politique de cryptage le choix de crypter la clé d'une manière publique au lieu de crypter les messages est du au fait que, la clé est souvent de petite taille par rapport aux données à chiffrer, donc, elle consomme beaucoup moins de temps lors de son chiffrement par rapport aux données.

C'est pourquoi ces dernières sont chiffrées de manière symétrique. ensuite, il ne reste qu'à transmettre le package contenant les données cryptées avec une clé privée, cryptée de son tour avec une clé publique. Une fois le package sera reçu, le récepteur procède inversement.

Tout d'abord, il déchiffre la clé chiffrée à l'aide de sa clé privée pour obtenir la clé de session, qui sera utilisée, par la suite, pour déchiffrer les données chiffrées. Ainsi, les performances seront améliorées en associant la rapidité des systèmes de chiffrement symétriques et la bonne sécurisation des systèmes de chiffrement asymétriques [14].

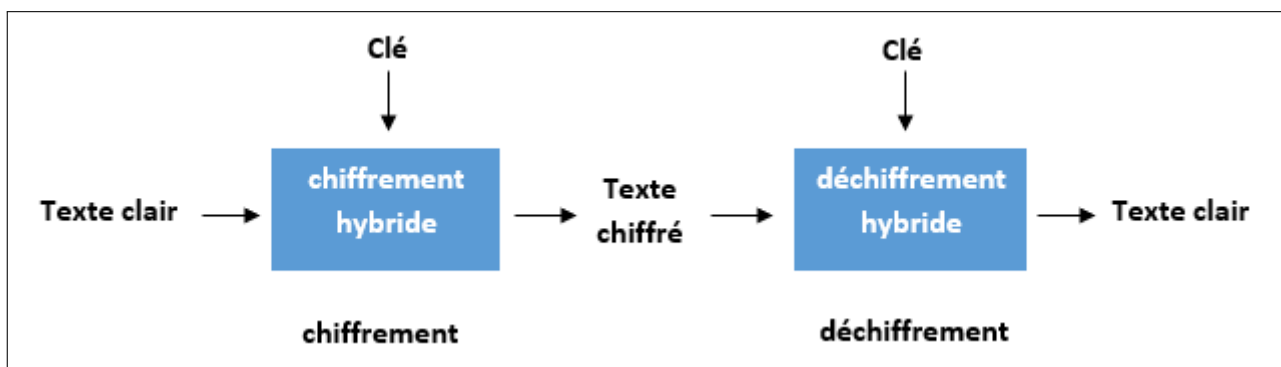


FIGURE 1.6 – Schéma illustre l'encryption hybride.

Les logiciels comme PGP et GnuPG parmi les méthodes qui fonctionnent de cette façon permettant de combiner les avantages des deux systèmes.

1.9 La cryptographie visuelle

La cryptographie visuelle est un domaine de la cryptographie dans lequel on utilise ou l'on transmet une image, dont le but est de pouvoir crypter une image en images cryptées n'ayant chacune aucune ressemblance ou corrélation avec l'originale [15].

1.10 Stéganographie

La stéganographie (du grec *steganos*, couvert et *graphein*, écriture) est l'art de cacher un message secret au sein d'un autre message porteur (texte, image, son, vidéo...) de caractère sans importance, de sorte que l'existence même du secret en soit dissimulée. Alors qu'avec la cryptographie, la sécurité repose sur le fait que le message chiffré soit incompréhensible pour les personnes non autorisées, avec la stéganographie, la sécurité repose sur le fait que la présence même d'un message secret ne sera sans doute pas soupçonnée et détectée. Il existe deux types de stéganographie : la sténographie linguistique, la stéganographie technique [16].

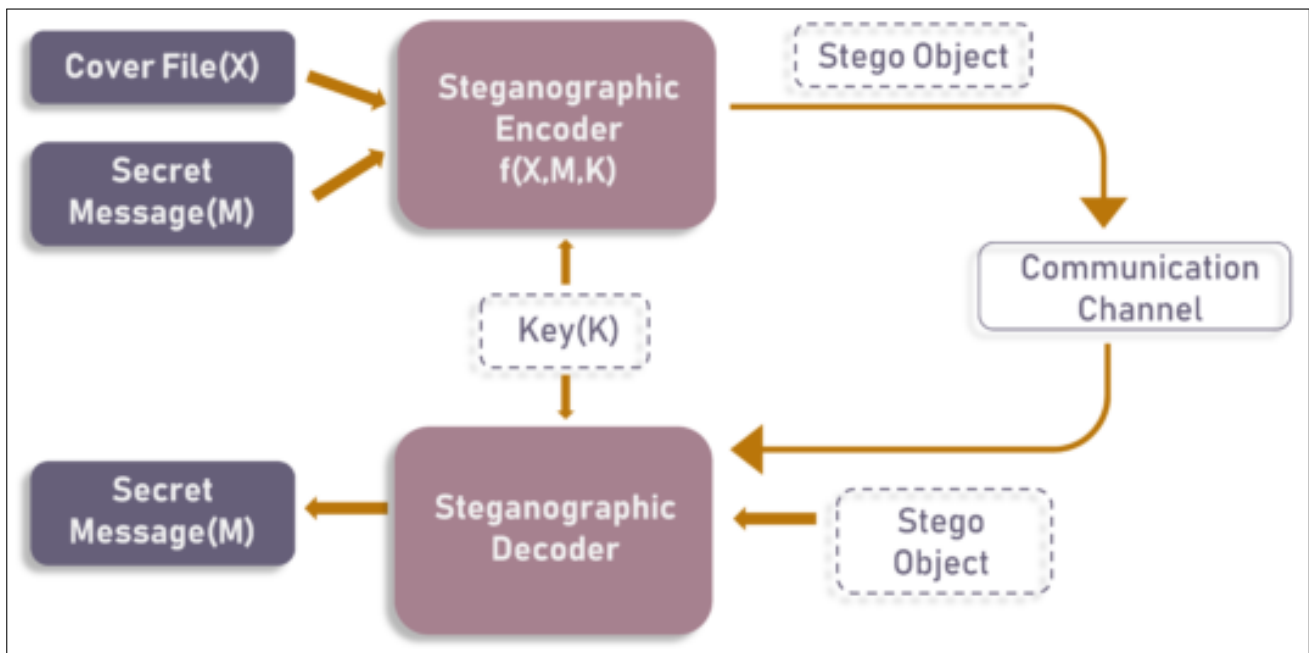


FIGURE 1.7 – Technique de stéganographie.

1.10.1 La stéganographie linguistique

La littérature sur la stéganographie linguistique, dans laquelle les propriétés linguistiques d'un texte sont modifiées pour cacher l'information, est faible par rapport à d'autres médias.

1.10.2 La stéganographie technique

La stéganographie technique regroupe toutes les techniques qui ne jouent pas sur les mots. La stéganographie technique est intéressante car elle permet de dissimuler des données dans plusieurs types de médias (Audio, Images, vidéo ...).

1.11 Conclusion

Dans ce chapitre, nous avons présenté des généralités sur la cryptographie où les types de cryptographie ont été mentionnés du traditionnel jusqu'au moderne, nous expliquons certains détails des types d'algorithmes de chiffrement modernes car nous les avons classés sous plusieurs formes via la clé de chiffrement (symétrique et asymétrique), de flux de données (flot et bloc) et hybride, aussi définir la stéganographie et leurs types les plus importants (linguistique et technique).

CHAPITRE 2

ÉTAT DE L'ART

2.1 Introduction

Maintenant, les vidéos numériques ont largement utilisé alors le rythme des progrès est donc très rapide dans les domaines de la communication vidéo et audio, grâce à cela est devenu la sécurité des données est une nécessité importante, les technologies d'échange de données ont également évolué.

Par conséquent, il nous a été imposé d'augmenter la confidentialité, la sécurité et l'intégrité des données l'utilisation des technologies vidéo numériques en particulier celles en temps réel il existe de nombreuses applications telles que WhatsApp, Skype ou système de Visio conférence Google Hangouts, Zoom, U Réunion mais le problème est la faiblesse du point de confidentialité, aussi lorsque nous attribuons le sujet au les vidéos numériques et aux méthodes d'encryptions, nous parlons ici les techniques lèges d'encryptions.

Avant de commencer à ces méthodes d'encryptions, il faut parler sur les fondamentaux de vidéo numérique.

2.2 Vidéo numérique

La vidéo numérique est une séquence animée d'images fixes qui permet de représenter la même information ou la même scène temporellement, Chaque séquence vidéo est caractérisée par des propriétés spatiales qui décrivent chaque image comme la résolution, l'espace de couleur choisi, le format de sous échantillonnage de couleur, et la résolution temporelle qui indique le nombre d'images par seconde fps (Frame per second) [17].

2.3 Conteneurs vidéo

Les vidéos numériques sont contenues dans des fichiers. Ces fichiers sont des conteneurs pour la vidéo [18].

Ils sont identifiables par un nom dont l'extension permet souvent de déterminer le type.

Les conteneurs les plus utilisés en vidéo sont :

- AVI (Audio Video Interleave).
- QuickTime (développé par Apple) avec l'extension mov.
- ASF (Advanced Streaming Format), développé par Microsoft avec l'extension asf, wmv, wma (audio).
- FLV (Flash Video), pour le lecteur Flash sur le web avec l'extension flv.

- de nombreux MPEG sous diverses formes : VOB (pour les DVD en MPEG-2), M2TS et TS (enregistrements TV), MP1, MP2(audio), M2V(vidéo), MP3(audio), MP4, etc.

La vidéo est un défilement d'une séquence d'images animées, nous allons commencer par jeter un bref coup d'œil sur les images numériques.

2.4 Les espaces de couleurs

Nous avons vu qu'une image numérique est un tableau de points représentant les couleurs (pixels). Il existe plusieurs formats de codage de l'information de couleur des pixels.

2.4.1 Image noire et blanc

Un seul bit suffit pour coder l'information, par exemple 0 pour le noir et 1 pour le blanc.

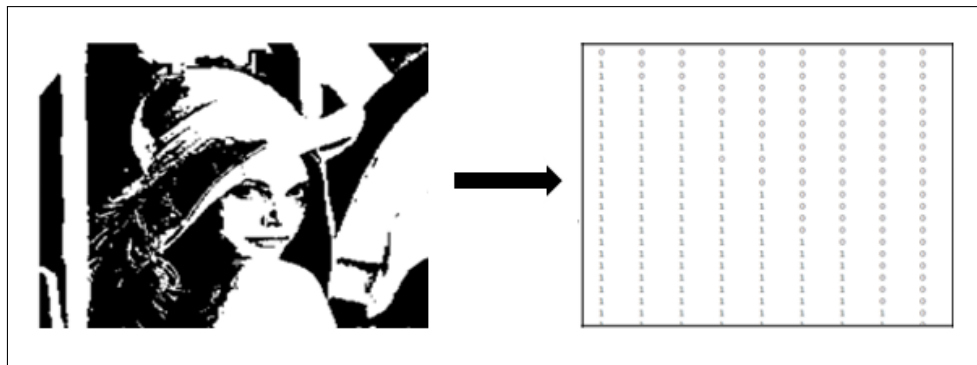


FIGURE 2.1 – Représentation numérique d'une image noir et blanc.

2.4.2 Image niveau de gris

On utilise en générale 8 bits, ce qui donne 256 nuances de gris possibles pour le pixel, de 0 (noir) à 255 (blanc).

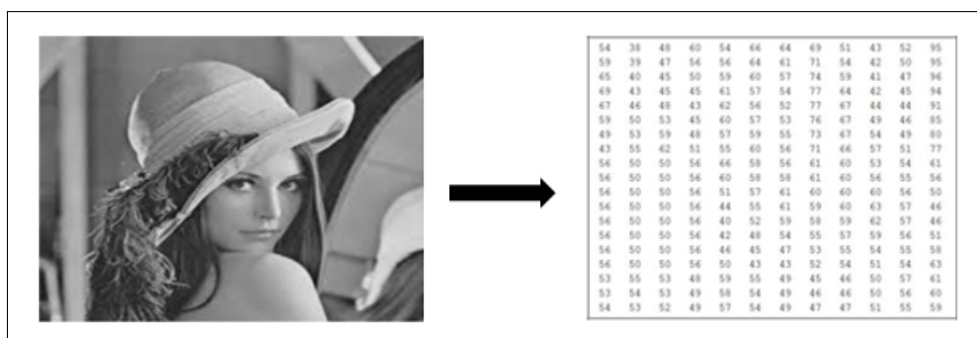


FIGURE 2.2 – Représentation numérique d'une image niveau de gris.

2.4.3 Image couleur (RGB)

La couleur de chaque pixel est définie par 3 composantes : Rouge, Vert et Bleu (système RVB où RGB en anglais).

L'intensité de chaque composante est codée sur 8 bits, donc chaque composante a une valeur comprise entre 0 (absence de couleur) et 255 (intensité maximale de la couleur).

Ainsi la couleur d'un pixel nécessite 24 bits (3 octets) pour être codée.

La couleur du pixel est obtenue par synthèse additive (RGB), en particulier :

- Si les 3 composantes sont à 0, on obtient du noir.
- Si les 3 composantes sont identiques on obtient une nuance de gris.
- Si les 3 composantes sont à 255, on obtient le blanc.

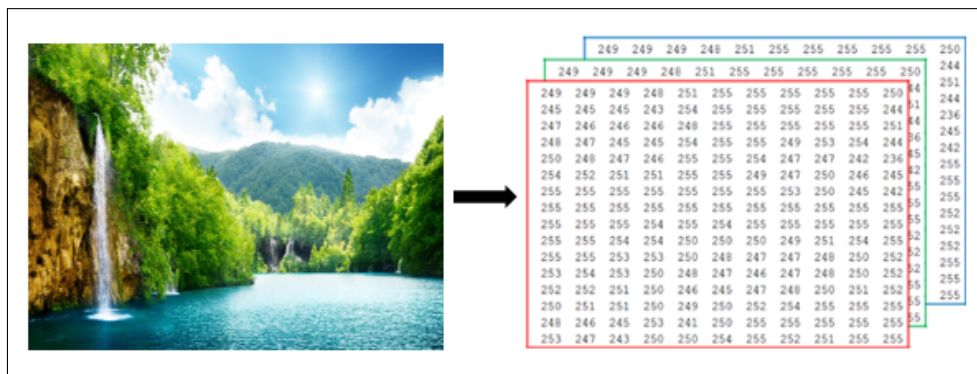


FIGURE 2.3 – Représentation numérique d'une image en couleur.

Pour illustrer, chaque pixel d'une image est représenté comme suite :

R	245
G	14
B	120

TABLE 2.1 – Représentation numérique d'un pixel.

La forme binaire de ce pixel est représentée comme suite :

R	11110101
G	00001110
B	01111000

TABLE 2.2 – La forme binaire d'un pixel.

2.5 Compression vidéo numérique

La compression vidéo est le processus de conversion d'un signal vidéo en un format qui utilise moins d'espace de stockage ou de bande passante de transmission. La compression vidéo est une technologie essentielle pour des applications telles que la télévision numérique (transmission terrestre, par câble ou par satellite), optique stockage / reproduction, télévision mobile, vidéoconférence et diffusion vidéo en continu sur Internet [19].

Il existe deux types de compression, la compression avec perte et la compression sans perte :

2.5.1 Compression sans perte

Avec la compression sans perte, les données sont compressées sans perte de données. Les techniques de compression sans perte, comme leur nom l'indique, ne comportent aucune perte d'information. Si les données ont été compressées sans perte, les données d'origine peuvent être récupérées exactement à partir là. La compression sans perte est généralement utilisée pour les applications qui ne tolèrent aucune différence entre les données originales et les données reconstruites.

2.5.2 Compression avec perte

Le type de compression le plus couramment utilisé pour la vidéo car il fournit des taux de compression beaucoup plus élevés. Il y a bien sûr un compromis : plus le taux de compression est élevé, plus la qualité de la vidéo compressée est mauvaise.

Les codecs avec perte ne conviennent pas aux données informatiques, mais sont utilisés en MPEG car ils permettent des facteurs de compression plus importants que les codecs sans perte.

2.6 Classification des algorithmes de cryptage de vidéo numérique

Nous classons les algorithmes de cryptage de vidéo numérique en quatre catégories [20].

2.6.1 Cryptage total

Dans cette classe, tout le contenu de la vidéo est d'abord compressé puis crypté à l'aide d'algorithmes traditionnels tels que DES, RSA, IDEA, AES, etc. Cette technique ne convient pas aux applications vidéo en temps réel en raison de calculs lourds et d'une vitesse lente.

2.6.2 Cryptage basé sur la permutation

Les algorithmes de cryptage de vidéo numérique de cette classe utilisent principalement différents algorithmes de permutation pour brouiller ou crypter le contenu de vidéo. Il n'est pas nécessaire de brouiller chaque octet. Certains algorithmes utilisent la liste de permutation comme clé secrète pour crypter le contenu de vidéo.

2.6.3 Chiffrement sélectif

Les algorithmes de cette classe chiffrent sélectivement les octets dans les images vidéo. Comme ces algorithmes ne chiffrent pas chaque octet de données vidéo, cela réduit la complexité du calcul.

2.6.4 Cryptage perceptif

Le cryptage perceptif nécessite que la qualité des données auditives / visuelles ne soit que partiellement dégradée par le cryptage, c'est-à-dire que les données multimédias cryptées sont encore partiellement perceptibles après le cryptage. Dans cette dégradation de la qualité auditive / visuelle peut être contrôlée en permanence par un facteur p .

Dans les techniques de protection vidéo traditionnels, appelés Cryptage total, tout le contenu est d'abord compressé, puis le flux binaire compressé est entièrement crypté à l'aide d'un chiffrement standard. Cette technique ne convient pas aux applications en temps réel en raison du retard élevé et de la complexité des calculs.

Le chiffrement sélectif traite qui ne crypte qu'un sous-ensemble des données. L'objectif du cryptage sélectif est de réduire la quantité de données à crypter tout en préservant un niveau de sécurité suffisant.

2.7 Les travaux connexes

Nous avons vu l'utilisation généralisée de la vidéo dans tous les domaines et l'importance du processus de sécurité et de cryptage de la vidéo est devenue la cible de nombreux algorithmes de chiffrement, Dans ce chapitre nous allons présenter quelques méthodes modernes pour le cryptage des vidéos numériques.

2.7.1 Approche basée sur l'hybridation

Cet approche est basée sur l'hybridation entre deux algorithmes, le premier est de type symétrique (AES), et le deuxième de type post quantum l'algorithme (NTRU), l'AES est utilisé pour chiffrer la vidéo, et l'NTRU est utilisé pour chiffrer la clé AES. Pour le scénario de transmission vidéo, l'expéditeur chiffre la vidéo et la clé et les envoie au destinataire, et pour accéder à la vidéo, le destinataire doit déchiffrer la clé AES avant de déchiffrer la vidéo [21].

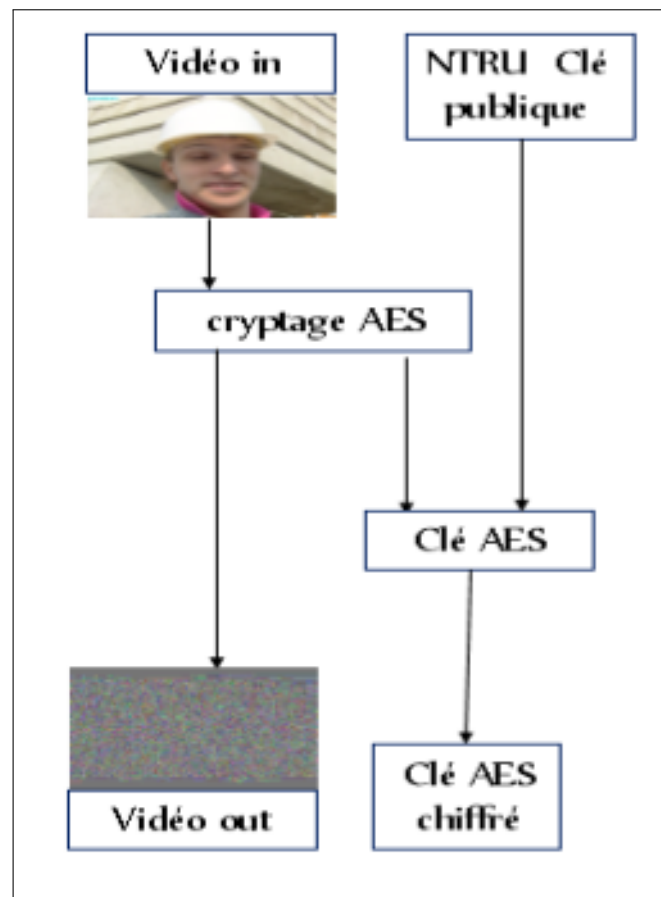


FIGURE 2.4 – Schéma illustre le processus de chiffrement d'un vidéo utilisé AES et NTRU.

Etape 1 : utiliser l'algorithme AES avec le mode CBC pour crypter la vidéo.

Etape 2 : utiliser la clé publique de crypto système NTRU pour chiffrer la clé secrète de l'AES.

Etape 3 : collecter la vidéo cryptée et la clé symétrique cryptée et les envoyer au destinataire.

Le **déchiffrement** se compose également de trois étapes, mais dans l'ordre inverse.

2.7.2 Approche Chaotique

Cette étude est utilisé deux algorithmes qui utilisent les formules mathématiques du Chaotique carte logistique et Suite de Fibonacci Modifiée pour générer un flux de clés pseudo aléatoire avec même taille d'image, puis faire l'opération XOR élément par élément entre image en clair et le flux de clés pseudo aléatoire généré, afin d'obtenir une image cryptée. Le but principal de ce chiffrement c'est masquer les bits d'image en clair avec les bits des clés pseudo aléatoire généré à travers l'opération ou-exclusif [22].

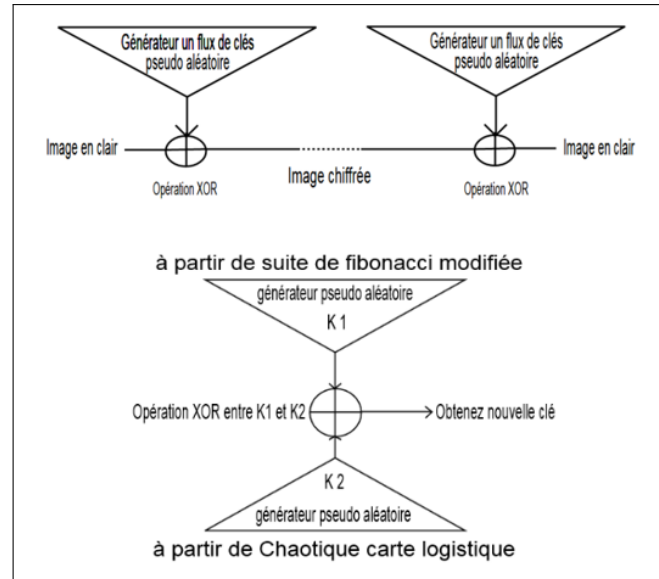


FIGURE 2.5 – Schéma illustre le processus de chiffrement une image utilisé XOR.

1) Générer un flux de clés pseudo aléatoire.

A. Définissez les valeurs initiales et le paramètre pour la Suite de Fibonacci, modifiée (L , K) et Chaotique Carte Logistique (r , X).

Générer deux flux de nombre pseudo aléatoire à travers les formules mathématiques de Suite de Fibonacci Modifiée (K_1) et de Chaotique Carte Logistique (K_2).

B. Faire la combinaison OU-exclusif (ou XOR) bit par bit entre ces flux générés. Pour obtenir un flux de clés pseudo aléatoire (Clé).

$$\text{Clé} = K_1 \otimes K_1$$

2) Convertir l'image en clair à un flux de données sous forme des bits m .

3) Faire la combinaison OU-exclusif (ou XOR) bit par bit entre le flux de données (l'image en clair) et flux de clés pseudo aléatoire. Pour obtenir un flux de données chiffrés (image chiffrée C).

$$C_i = M_i \otimes \text{Clé}$$

2.7.3 Approche de cryptage du boîtier

Cette technique de cryptage vidéo qui compte tenu du cryptage du boîtier- I (trame vidéo), pensé à partir du calcul de la grille pour créer le contour en I codé. Dans cette stratégie, nous rassemblons l'ensemble de la vidéo puis prenons décrire une par une structure et sélectionner une image clé pour le processus de cryptage et de décryptage, de sorte que cette image clé est envoyée à canal sécurisé. Autre bord brouillé en prenant après calcul.à la suite de l'application du calcul de chiffrement, Cette technique rassemble toutes les boîtes, crée une vidéo dans une image bruit et l'envoie à partir d'un canal simple.

Soit V un groupement vidéo comprenant n boîtiers désignés par I_1, I_2 Après exécuter le calcul en deux passes en utilisant les clés K_1 et K_2 . Dans la première clé de passe, K_1 est utilisé pour coder les boîtiers impairs en exécutant Bit XOR avec la première clé 1.

Dans la seconde clé de passe, K_2 est utilisé pour brouiller les bords pairs en exécutant l'opération Bit XOR avec la seconde clé 2.

Il s'agit actuellement de la dernière image encryptée. à ce stade, crée la vidéo crypté à travers ces images encryptée et envoyez-la via le canal de base.

Cependant, l'image clé transmise via un canal sécurisé. Du côté de la réception, l'opération est inversée [23].

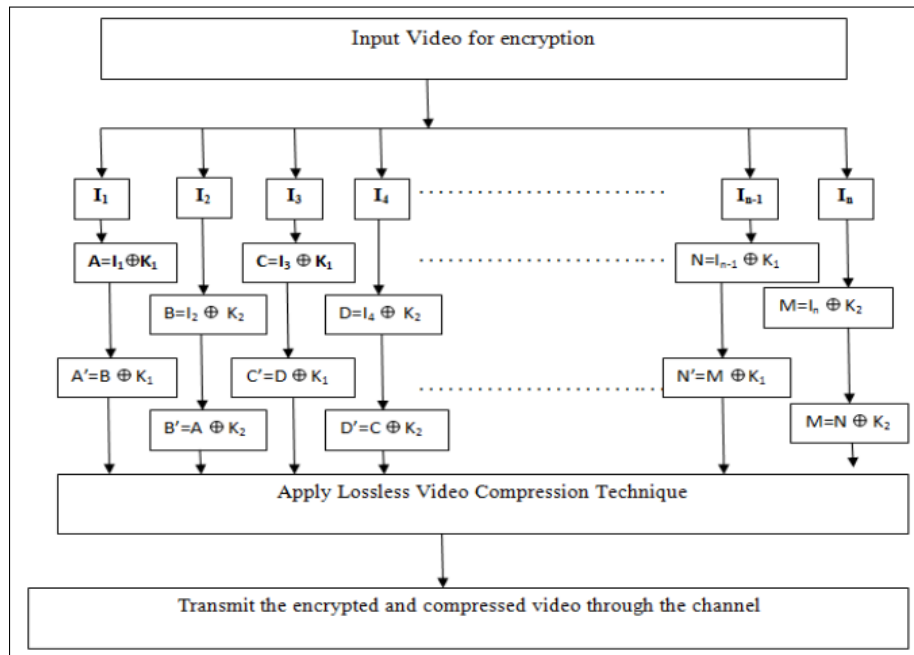


FIGURE 2.6 – Schéma illustre le processus d'encryption et de compression vidéo.

2.7.4 Approche Moderne

Cette technique de cryptage vidéo est proposée par Iyer et all. qui ont proposé une approche pour sécuriser les vidéos basées sur la cryptographie hybride. Ils utilisent des fichiers textes intermédiaires au lieu de la vidéo cadres ou images. Il convertit le flux vidéo original en un fichier texte en le réduisant en un équivalent base64. Ce fichier texte est ensuite soumis à un double cryptage, c.-à-d. ECC et AES fournissent une double sécurité [24]. Le processus de cryptage de vidéo est illustré dans la figure suivante :

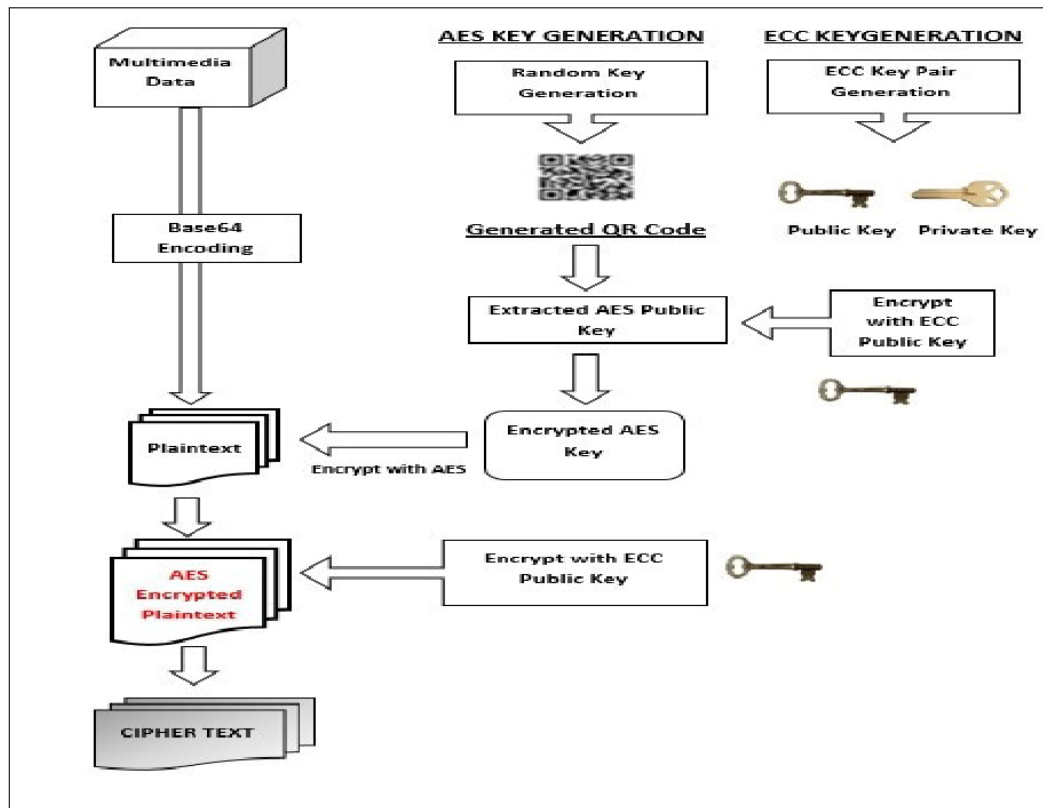


FIGURE 2.7 – Schéma illustre le processus de cryptage de la vidéo utilisé Iyer et all.

- Les étapes de processus de cryptage

- 1- Lire la vidéo d'entrée de l'utilisateur.
- 2- Convertir la vidéo en un fichier texte au format Base64.
- 3- Générer des clés publiques et privées ECC et l'AES clé symétrique.
- 4- Cryptez le fichier Base64 avec la clé publique ECC.
- 5- Chiffrer le fichier obtenu à l'étape précédente avec le Clé symétrique AES.
- 6- Créer un code QR qui contient fichier crypté final, la clé AES et la clé privée ECC.

Le principe du processus de décryptage vidéo fonctionne d'une manière dans le sens inverse et présenter dans la figure suivante :

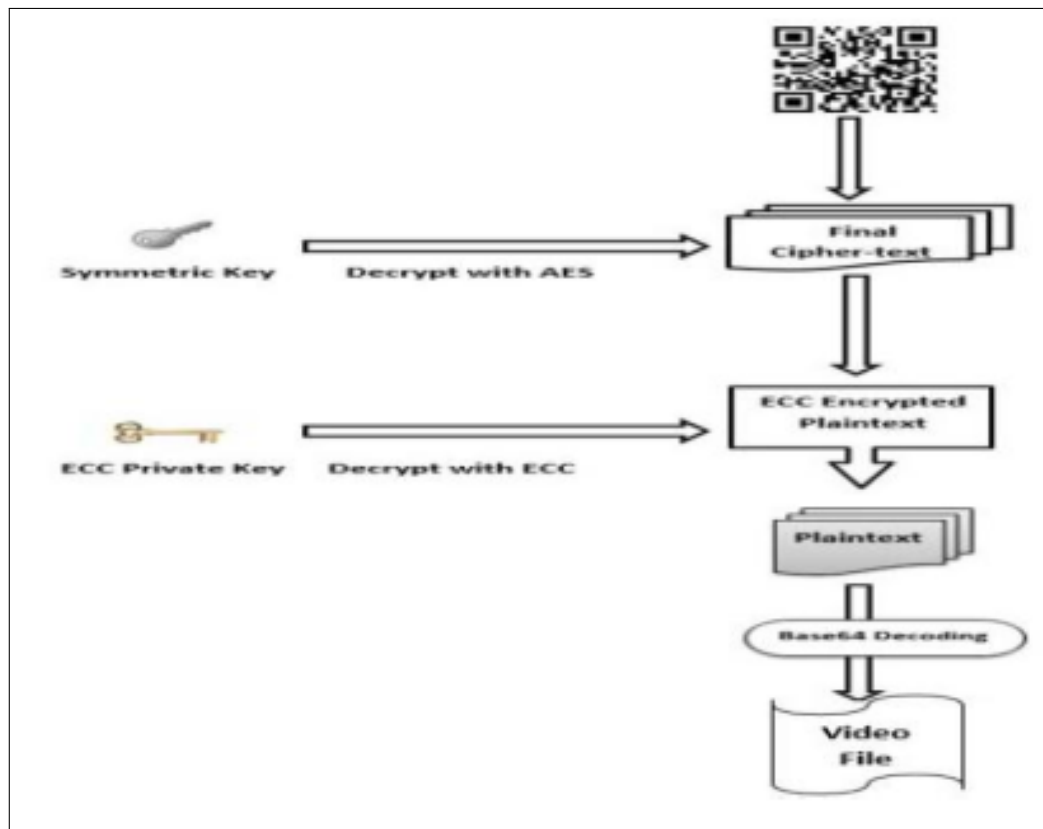


FIGURE 2.8 – Schéma illustre le processus de décryptage de la vidéo.

2.7.5 Approche basée sur le RSA et PN (Pseudo Noise)

Ce schéma propose un algorithme de cryptage vidéo. Il utilise la séquence RSA et Pseudo Noise (PN), destiné aux applications nécessitant des transferts d'informations vidéo sensibles.

Il est principalement conçu pour fonctionner avec des fichiers encodés à l'aide du codec AVI (Audio Vidéo Interleaved), la figure suivante présente le processus de cryptage et de décryptage [25].

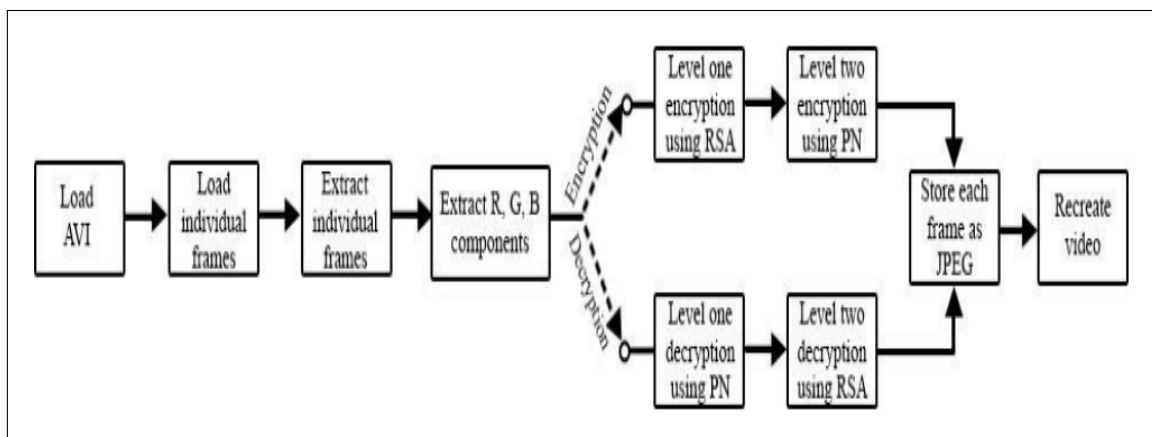


FIGURE 2.9 – Schéma illustre le cryptage et décryptage de vidéo utilisé RSA et PN.

Les étapes de processus de chiffrement

- 1- Charger Le fichier AVI.
- 2- Extraire Les images du fichier chargé une par une.
- 3- Après le processus d'extraction, les images sont chargées.
- 4- Séparez les composants RGB de chaque image.
- 5- Les trames RGB sont cryptées individuellement à l'aide de l'algorithme RSA.
- 6- Crypter les trames RGB le 2ème fois par « pseudo random sequence noise ».
- 7- Les composants RVB cryptés sont ensuite combinés en un fichier JPG.
- 8- Les étapes 3 à 6 sont répétées pour toutes les images extraites.

9- Le résultat est obtenu après utilisation de toutes les images stockées pour créer un fichier vidéo avec chaque image cryptée stockée en tant qu'image individuelle de la vidéo.

2.8 Conclusion

Dans ce chapitre nous avons parlé sur les notions de bases de vidéo numérique et quelques méthodes modernes pour le cryptage des vidéos numériques et expliquez les principes de leur fonctionnement.

Dans le prochain chapitre, nous allons parler de notre proposition, nous l'avons fait surtout pour améliorer deux points importants ils sont : la rapidité de chiffrement et les techniques légères d'encryptions qui ne nécessite pas d'équipement puissant.

CHAPITRE 3

L'APPROCHE PROPOSÉE

3.1 Introduction

En ce chapitre concerne la proposition en générale, en utilise l'encryption hybride qui consiste en une combinaison de techniques de cryptages asymétrique et symétrique, la méthode asymétrique pour générer la clé partagée et applique la méthode symétrique pour augmenter la rapidité de chiffrement et déchiffrement.

Dans notre proposition nous consacrons sur les données volumineuses comme les vidéos numériques tel que se servir du mode flux dans l'échange de données.

On propose une nouvelle technique Pool pour garantir l'extrême de la rapidité.

3.2 Schéma globale

Dans cette partie, nous présenterons la procédure globale de notre proposition, tel que la figure suivante illustrée toutes les étapes de notre proposition, à partir le partage des clés publiques jusqu'à le déchiffrement de vidéo.

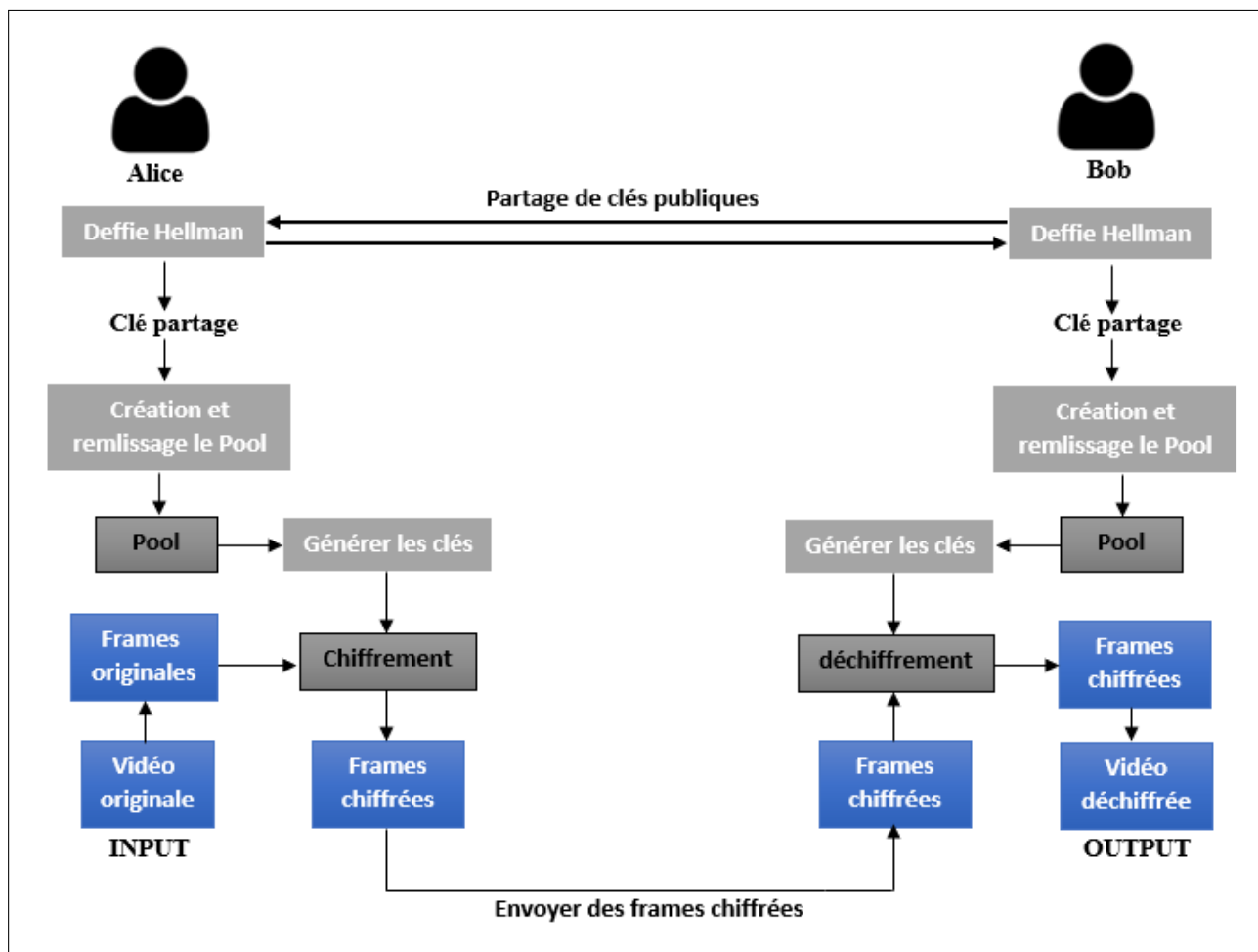


FIGURE 3.1 – Schéma global de notre proposition.

Nous allons diviser notre travail en deux parties principales, la première partie est de créer le Pool qui génère les clés en manière pseudo-aléatoire et la seconde partie est chiffrement et déchiffrement de vidéo.

3.3 La création de Pool :

Définition

Le Pool c'est allocation temporaire de mémoire ou un fichier de taille prédéfini remplir par suite des bits pseudo-aléatoire selon un algorithme, nous l'avons déjà défini pour l'objectif d'utiliser différent de clés de chiffrement et déchiffrement.

Avant la création du Pool, nous appliquerons le processus de partage de la clé, afin d'assurer le même Pool est crée sur des deux côtés.

3.3.1 Processus de partage de la clé :

La clé partagée est une partie essentielle dans notre méthode de cryptage, ils existent plusieurs techniques concernées par exemple *RSA*, *Diffie Hellman* et *El Gamal*.

Dans notre proposition, nous utiliserons la méthode Diffie Hellman pour obtenir la clé partagée. A cause de, il est basé sur des opérations mathématiques plus simples où le temps d'exécution est plus rapide aussi la réalisation au côté programmation est facile.

✓ Le processus du partage

La mise en œuvre la plus simple et originale du protocole utilise le groupe multiplicatif d'entiers modulo (p), où (p) est premier, et (g) est un modulo racine primitif (p).

1. Alice et Bob s'accordent sur un nombre premier (p) et une base (g).
2. Alice choisit un entier secret (a), puis envoie Bob.

$$A = g^a \mod p$$

3. Bob choisit un entier secret (b), puis envoie Alice.

$$B = g^b \mod p$$

4. Alice calcule.

$$K1 = B^a \mod p$$

5. Bob calcule.

$$K2 = A^b \mod p$$

6. Alice et Bob partagent maintenant un secret, c'est-à-dire que Bob et Alice peuvent utiliser ce numéro comme clé.

La figure suivante qui illustre cet processus.

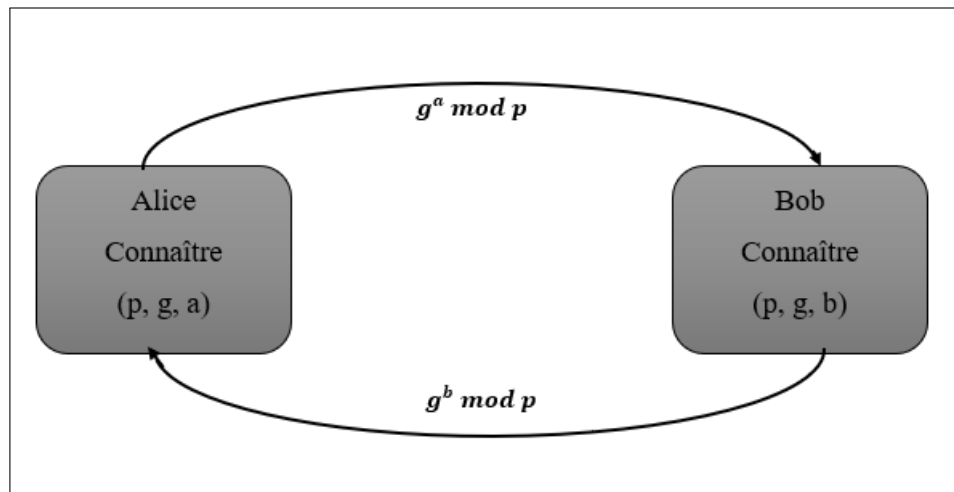


FIGURE 3.2 – Le processus de Diffie Hellman

✓ Correction de Diffie-Hellman et sa preuve

1. Alice a calculé

$$\begin{aligned}
 A &= g^a \mod p \\
 K1 &= B^a \mod p \\
 &= (g^b)^a \mod p \\
 &= (g^a)^b \mod p \\
 &= A^b \mod p
 \end{aligned}$$

2. Bob a calculé

$$\begin{aligned}
 A &= g^b \mod p \\
 K1 &= A^b \mod p \\
 &= (g^a)^b \mod p \\
 &= (g^b)^a \mod p \\
 &= B^a \mod p
 \end{aligned}$$

Par conséquent

$$K1 = K2$$

3.3.2 Remplissage de Pool

Nous proposons un algorithme de remplir le Pool qui utilise la clé partagée pour garantir la confidentialité à deux cotés (Alice et Bob).

Algorithm 1 Remplissage de Pool

```

1: Initialise:
    i = compteur.
    N = clé partagée.
    S = taille de Pool.
2:  $i \leftarrow 1$ 
3: while isEmptyPool do
4:   if length(S)  $\geq$  S then
5:     isEmptyPool = False
6:   else
7:     current =  $N^i \pmod S$ 
8:     Pool.append(current)
9:   end if
10:   $i++$ 
11: end while
  
```

Description de l'algorithme

- ✓ Commence par une boucle de vérifier si le Pool est plein ou non.
- ✓ Si le Pool est plein, quitte la boucle et le processus se termine.
- ✓ Sinon le Pool contient un espace vide, alors génère la valeur *current* par cette équation $N^i \pmod S$, on convertit en binaire et l'ajoute dans le Pool par la commande append (ajoute un élément à la fin de la liste). Le Pool est le même aux deux extrémités, pour que nous ayons la même liste de clés. Chaque fois que nous envoyons une autre clé partagée qui génère par l'algorithme de Diffie-Hellman, remplir à nouveau Pool.

La figure suivante illustre à un certain moment tel que la valeur de *current* est égale 140, on convertit en binaire et l'ajoute dans le Pool.

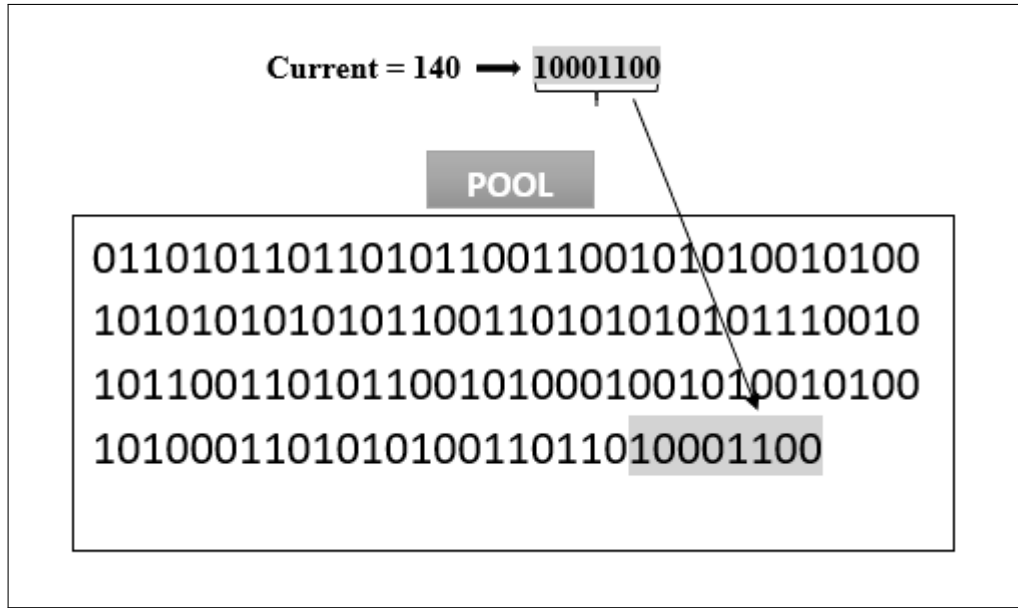


FIGURE 3.3 – Schéma illustre le remplissage de Pool.

3.3.3 Générer les clés de chiffrement et déchiffrement

Générer suite de bits à travers le Pool, après l'opération de remplissage de Pool et prêt pour générer les clés, tel que la taille de clé est un ensemble de bits prédéfini.

La méthode utilisée est présentée dans l'équation $N \times i \pmod{S}$ pour chaque i l'équation génère un pseudo-aléatoire qui utilise, pour déterminer le début de suite de bits à utiliser sur le chiffrement et déchiffrement les bits suivants de la vidéo.

Algorithm 2 Génère position de clé

1: **Initialise:**

i = compteur.

N = clé partagée.

S = taille de Pool.

K_i = la Position de début des bits de la clé numéro i .

2: $i \leftarrow 1$

3: **while** VideoNotFinished **do**

4: **if** EndOfVideo **then**

5: VideoNotFinished = False

6: **else**

7: $K_i = N \times i \pmod{S}$

8: **end if**

9: $i++$

10: **end while**

La figure suivante illustre comment extraire la clé de chiffrement et déchiffrement par le K_i .

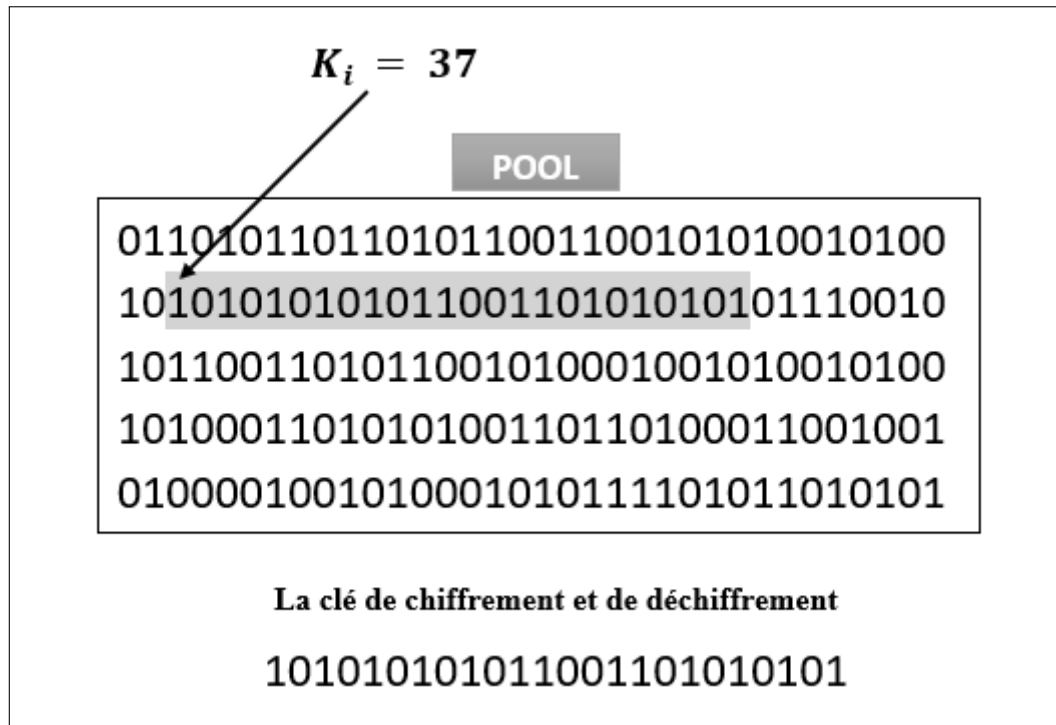


FIGURE 3.4 – Schéma illustre comment extraire la clé.

3.4 Les processus de chiffrement et de déchiffrement

Dans cette partie nous expliquons les étapes en détail les processus de chiffrement et de déchiffrement.

3.4.1 Processus de chiffrement

Nous appliquons les étapes suivantes pour chiffrer la vidéo et envoyer les suites des frames chiffrées à l'autre côté.

- Lire une vidéo.
- Extraire des images(frames) de vidéo.
- Chiffrer chaque image individuelle.
- Envoyer les images chiffrées à l'autre côté.

Où le nombre des frames dépend par du type de résolution vidéo, tel que le nombre des frames par seconde est une unité de mesure correspondant au nombre d'images affichées en une seconde par un dispositif.

La figure suivante qui illustre le processus de chiffrement et les étapes à appliquer.

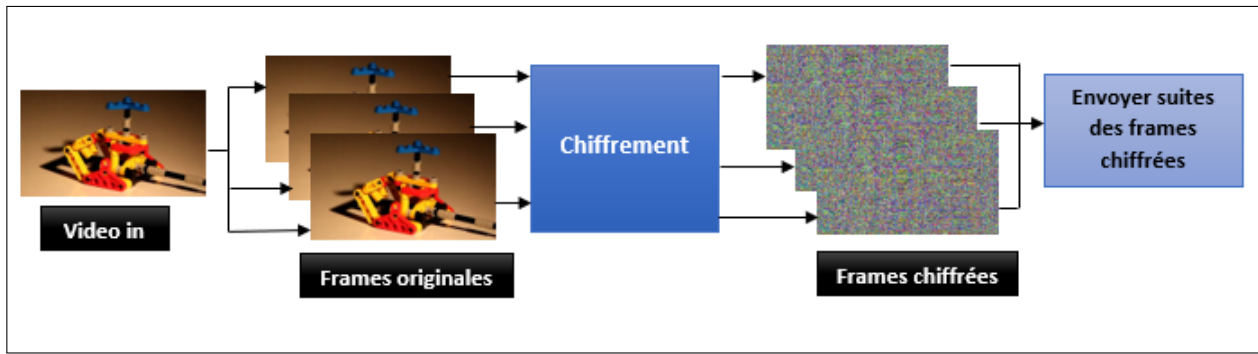


FIGURE 3.5 – Processus de chiffrement une video.

3.4.2 Processus de déchiffrement

Le déchiffrement se compose également les mêmes étapes, ceci afin d'obtenir des images claires et, enfin, de les assembler sous forme de vidéo clair.

- Réceper les images chiffrées de l'autre côté.
- Déchiffrée chaque image chiffrée individuelle.
- Assemblez toutes les images déchiffrées sous forme vidéo.
- Afficher la vidéo déchiffrer.

La figure suivante qui illustre le processus de déchiffrement et les étapes à appliquer.

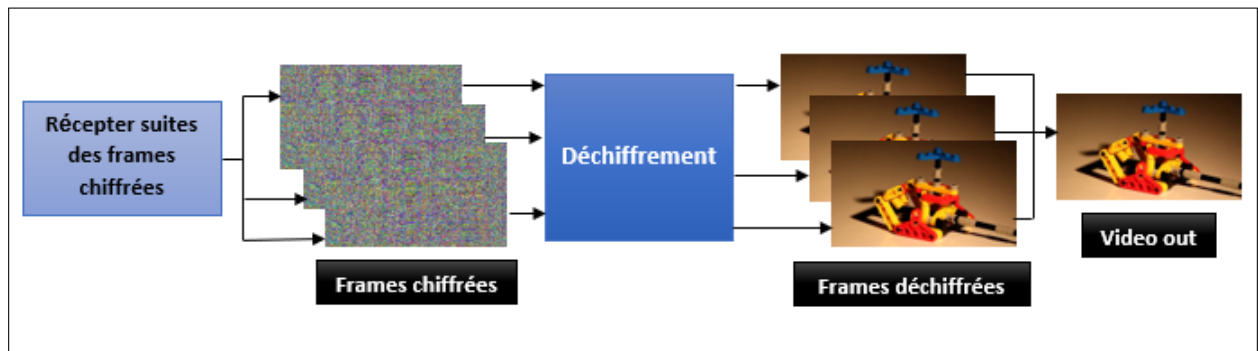


FIGURE 3.6 – Processus de déchiffrement une video.

3.4.3 Description de chiffrement et déchiffrement d'image

Le chiffrement et déchiffrement d'un seul pixel de frame passe sur les étapes suivantes :

- ✓ Après le remplissage de Pool, Nous appliquons l'algorithme de générer de clé pour extraire la clé de chiffrement tel que égale 24 bits.
- ✓ Nous appliquons l'opérateur XOR à premiers 8 bits de la clé et à la valeur de couleur rouge du pixel. Applique le même opérateur sur la seconde 8 bits de la clé et la valeur de

couleur vert du pixel. Enfin, nous répétons le même opérateur sur le dernier 8 bits de la clé avec la valeur de couleur bleu du pixel.

- ✓ Nous appliquons ces étapes à tous les pixels pour obtenir une image cryptée.
- ✓ Pour le déchiffrement d'image, nous appliquons les mêmes étapes à l'image chiffrée pour obtenir l'image claire.

1. La position début de la clé dans cet exemple est égale 9, tel que c'est pendant une étape i depuis une application de l'algorithme de génération de clés, la taille de la clé est 24 *bits*.

La figure suivante illustre cet processus, elle explique après avoir obtenu la clé, comment obtenir suites de bits que nous appliquerons dans l'opérateur XOR.

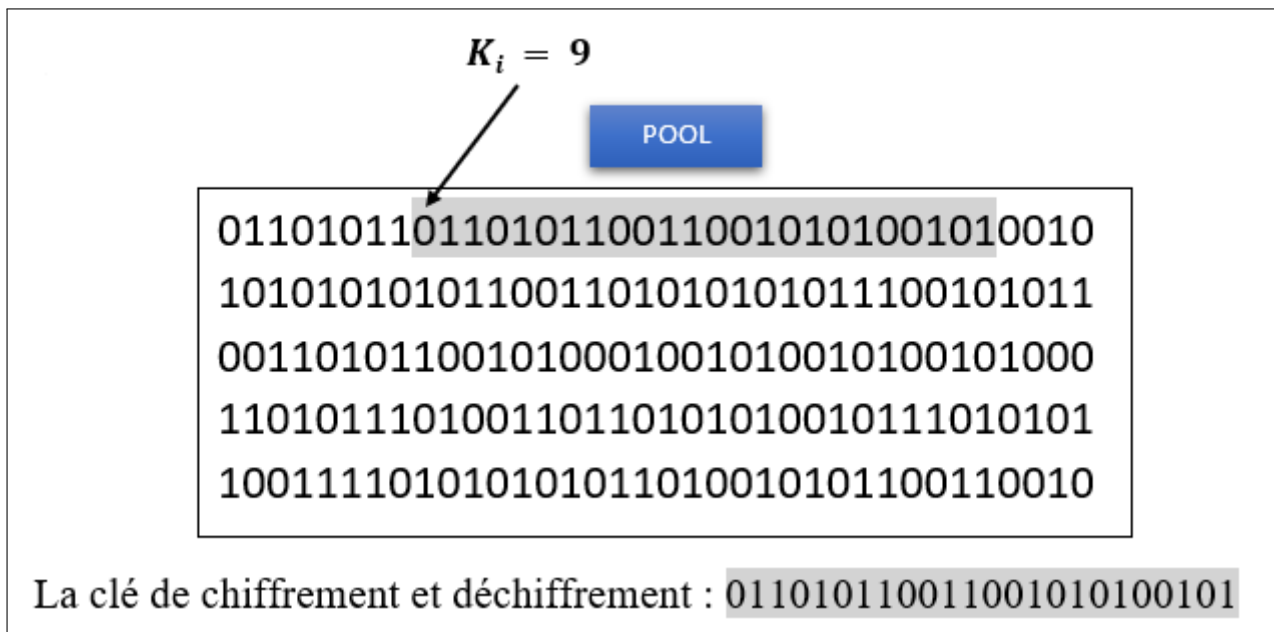


FIGURE 3.7 – Extraire la clé de chiffrement et déchiffrement du Pool.

2. Nous chiffons un pixel avec une clé spécifique, la table suivante illustre ce processus tel que après avoir extrait une clé de Pool, l'illustration est que chaque valeur de couleur RGB que nous appliquons l'opérateur XOR pour chiffrer un pixel.

	$Pixel_i$	$Pixel_i$ chiffré
R	01011101 $\oplus$ 01101011	00110110
G	10010101 $\oplus$ 00110010	10100111
B	10110011 $\oplus$ 10100101	00010110

TABLE 3.1 – Chiffrement d'un pixel.

3. Nous déchiffrons un pixel avec la même clé, la table suivante illustre cet processus.

	$Pixel_i$ chiffré	$Pixel_i$ déchiffré
R	00110110 $\oplus$ 01101011	01011101
G	10100111 $\oplus$ 00110010	10010101
B	00010110 $\oplus$ 10100101	10110011

TABLE 3.2 – Déchiffrement d'un pixel.

3.5 Conclusion

Dans ce chapitre, nous avons présenté notre proposition pour objectif chiffrement et déchiffrement des données volumineuse.

Nous avons utilisé méthode hybride qui compose par une méthode symétrique (algorithme XOR pour chiffrer la vidéo) et asymétrique (algorithme Diffie Helman pour génère clé partagé) en outre nouvelle technique le Pool qui contient ensemble de clé.

CHAPITRE 4

IMPLÉMENTATION ET ANALYSES

4.1 Introduction

Nous avons présenté dans le chapitre précédent notre proposition de chiffrement et déchiffrement de la vidéo numérique, nous entamons maintenant sur ce dernier chapitre la partie de la réalisation et de l'analyse, nous présentons notre application réalisée et les différentes mesures d'évaluation pour montrer la sécurité et l'efficacité de l'algorithme proposé.

4.2 Environnement de développement

Dans cette partie nous allons citer l'environnement logiciel (Software) et matériel(Hardware) utilisés.

4.2.1 Environnement logiciel

Nous avons utilisé les deux environnements Python et MATLAB pour l'implémentation de notre approche.

- **Python**

Python est un langage de programmation interprété, multi-paradigme et multiplateformes. Il favorise la programmation impérative structurée, fonctionnelle et orientée objet. Il est doté d'un typage dynamique fort, d'une gestion automatique de la mémoire par ramasse-miettes et d'un système de gestion d'exceptions [26].

Nous avons utilisé Python pour notre projet de traitement vidéo et d'image parce qu'il est riche en bibliothèques prêtes à l'emploi qui facilitent le processus de programmation pour que nous soyons plus concentrés sur l'idée du projet. Voici quelques-unes des bibliothèques de traitement d'image utiles et disponibles gratuitement en Python nous l'avons utilisé dans notre projet : Numpy, SciPy, Matplotlib, PIL/Pillow, OpenCV et Scipy.

- **MATLAB**

« Matrix Laboratory » est un langage de programmation de quatrième génération émulé par un environnement de développement du même nom ; il est utilisé à des fins de calcul numérique. Développé par la société The MathWorks, Matlab permet de manipuler des matrices, d'afficher des courbes et des données et de mettre en œuvre des algorithmes [27].

Nous avons utilisé le Matlab pour effectuer des opérations des analyses et de comparaison telles que entropie et la corrélation ... etc en ce qui concerne le chiffrement / déchiffrement de la vidéo.

4.2.2 Environnement matériel

L'application a été développée sur un PC(Laptop HP modèle : Latitude E5450) ayant les caractéristiques suivantes :

- **Processeur** : Intel® Core(TM) i3-5300U CPU @ 2.30GHz 2.29 GHz.
- **Mémoire installée (RAM)** : 8,00 Go.
- **Disque Dure** : HDD 500 Go.
- **Carte graphique** : Intel ® HD graphics5500.
- **Système d'exploitation** : Windows 10 Professionnel 64 bit.

Nous avons utilisé 85 frames de vidéo test.mp4 avec une résolution de 560×320 de format mp4 avec fréquence d'images est 15 fps, 237 ko taille et 5s longueur.

4.3 Critères d'évaluation

Un bon système de cryptage devrait résister à toutes sortes d'attaques connues, Donc il y a des simulations numériques qui ont été effectuées en utilisant différentes mesures d'évaluation pour montrer la sécurité et l'efficacité de l'algorithme proposé. Nous allons présenter les plus important comme : l'espace de clés, l'histogramme, l'entropie, la corrélation entre les pixels adjacents.

4.3.1 Espace de clés

Un bon algorithme de chiffrement doit être sensible aux clés de chiffrement et l'espace clé doit être suffisamment grand pour rendre les attaques de force brute impossibles. La clé utilisée dans notre schéma est des nombres pseudo aléatoires qui été généré par les paramètres (N, S), la taille de la clé secrète est dépende par la taille du pool et la clé partagée , alors chaque élément dans les nombres pseudo aléatoires est codé sur 24 bits. Donc l'espace de clés c'est $2^{24 \times N \times S}$.

Par exemple pour un Pool de taille 1 ko et N : clé partagée est 19, l'espace de clés c'est $2^{24 \times 8192 \times 19}$.

4.3.2 L'histogramme

L'histogramme d'une image fait référence à un graphique du pixel valeurs d'intensité.

L'histogramme est un graphique montrant le nombre de pixels dans une image à différentes valeurs d'intensité trouvé dans l'image [28].

Les figures suivantes illustre des images prises à différents moments de la vidéo (début, centre, finale).



FIGURE 4.1 – Une image clair de début de la vidéo.



FIGURE 4.2 – Une image clair de centre de la vidéo.



FIGURE 4.3 – Une image clair de finale de la vidéo.

Les tracés des histogrammes des images claires prises de la vidéo sont montrés dans les figures ci-dessous.

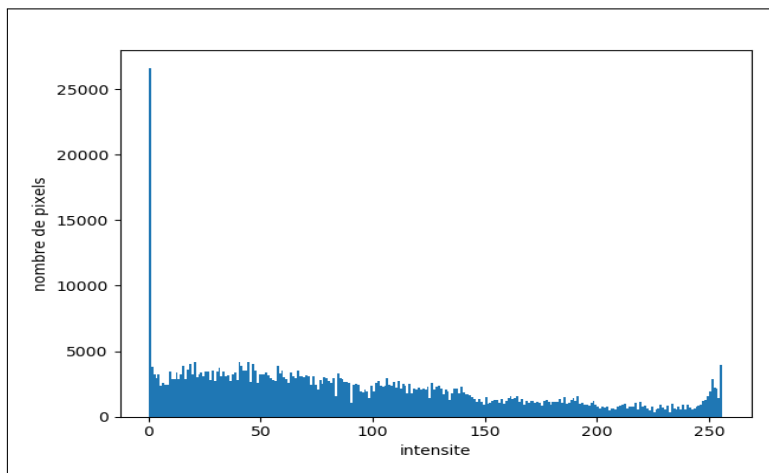


FIGURE 4.4 – Histogramme d’une image clair de début de la vidéo.

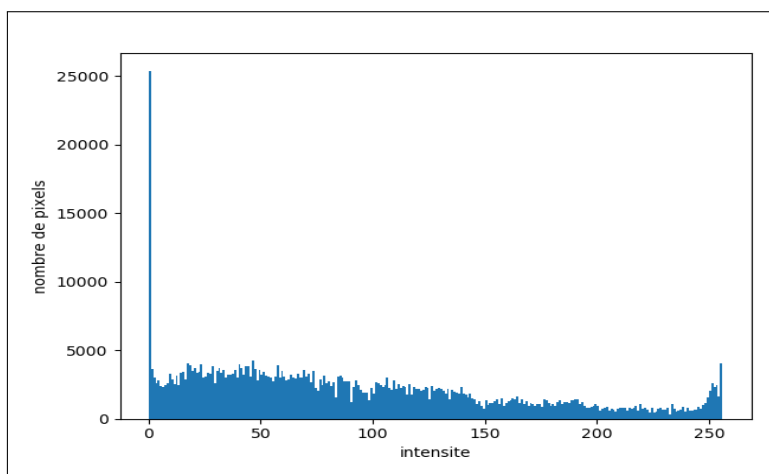


FIGURE 4.5 – Histogramme d’une image clair de centre de la vidéo.

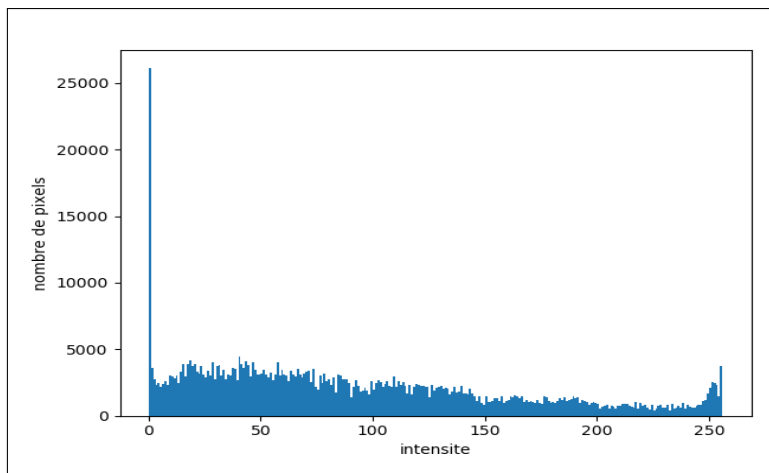


FIGURE 4.6 – Histogramme d’une image clair de finale de la vidéo.

Les figures suivantes illustre le cryptage des images précédentes prises à différents moments de la vidéo.



FIGURE 4.7 – Une image chiffrée de début de la vidéo.

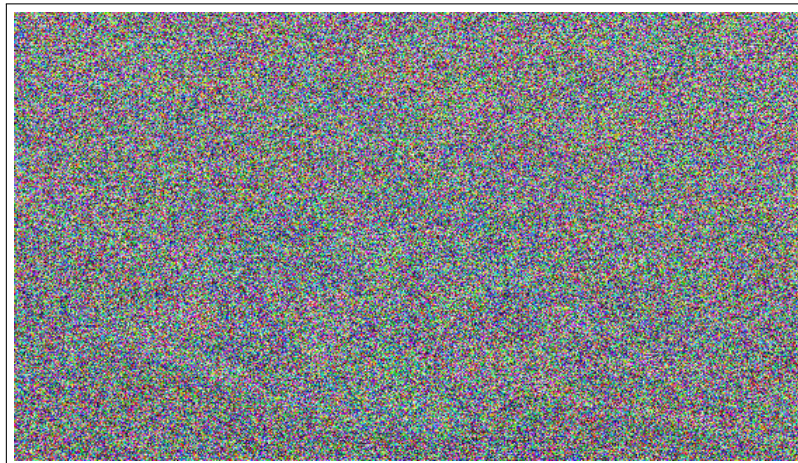


FIGURE 4.8 – Une image chiffrée de centre de la vidéo.

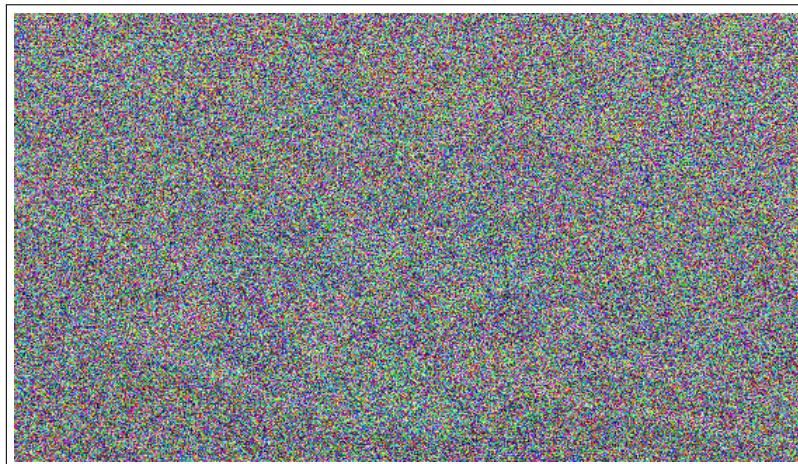


FIGURE 4.9 – Une image chiffrée de finale de la vidéo.

Les tracés des histogrammes des images chiffrées prises de la vidéo sont montrés dans les figures ci-dessous.

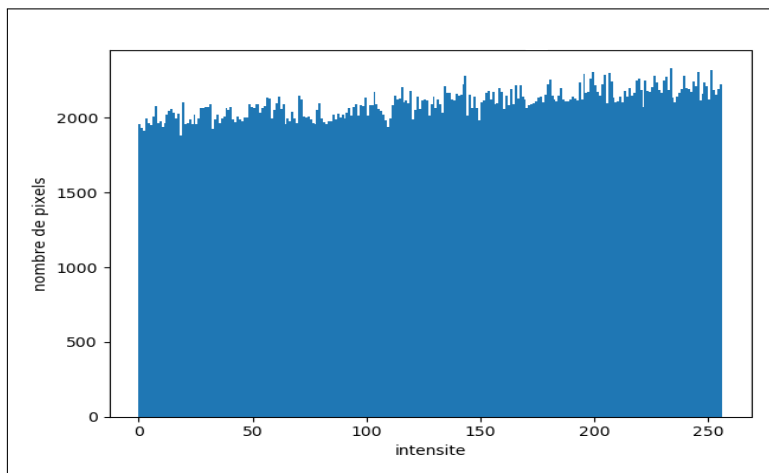


FIGURE 4.10 – Histogramme d'une image chiffrée de début de la vidéo.

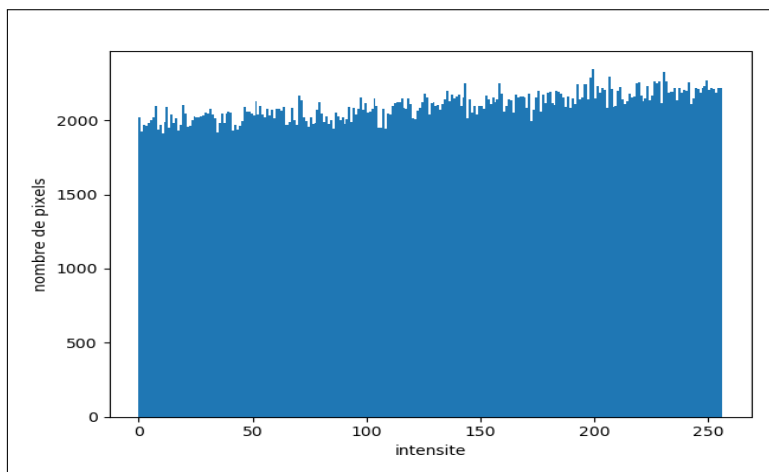


FIGURE 4.11 – Histogramme d'une image chiffrée de centre de la vidéo.

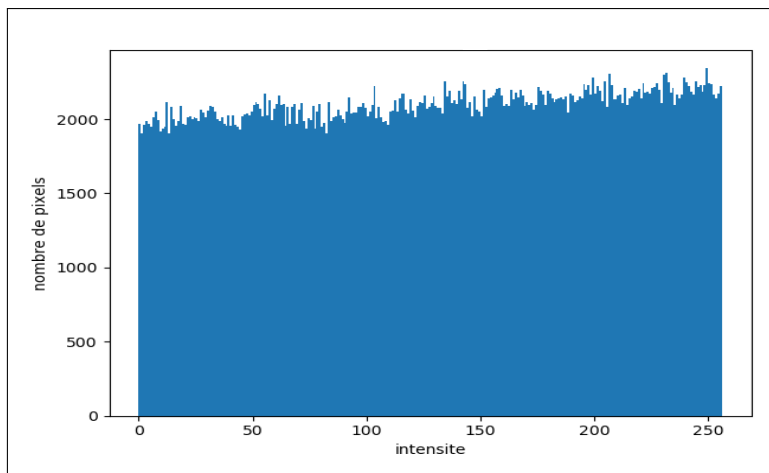


FIGURE 4.12 – Histogramme d'une image chiffrée de finale de la vidéo.

Le résultat montre que les histogrammes des images chiffrées sont uniformes après le cryptage. Par conséquent l'attaquant ne peut pas extraire des informations à partir de l'histogramme de l'image cryptée.

Aussi n'y a pas de relation ou de similitude entre les histogrammes des images chiffrées et les histogrammes des images claires, ni en couleur ni en intensité. Donc, nous ne pouvons pas prédire l'image réelle ou leur histogramme à partir de l'histogramme d'image chiffrée.

4.3.3 Mesure de distorsion (PSNR)

PSNR (Peak Signal to Noise Ratio), c'est une mesure de distorsion utilisée en image numérique, est calculé à l'aide de l'erreur quadratique moyenne (EQM) qui devrait idéalement être aussi faible que possible pour un déchiffrement sans perte [29]. Le PSNR est calculé comme suit :

$$PSNR_{dB} = 10 \log_{10} \left(\frac{L_2}{EQM} \right)$$

Où **L** : valeur maximale possible du pixel de l'image.

EQM : est l'erreur quadratique moyenne, elle est définie pour 2 images et de taille m*n par la formule suivante :

$$EQM = \frac{1}{m \times n} \sum_i^N \sum_j^M |I_{ij} - \hat{I}_{ij}|$$

Pour notre approche la valeur PSNR d'une image claire Frame40 prise à un certain moment de la vidéo et Frame40encrypted la même image est cryptée est **7.17574601379**.

Dans la même formulation, nous avons calculé la valeur PSNR d'une image claire Frame40 et Frame40 decrypted la même image après le décryptage, le résultat est les deux images sont identiques, et donc l'erreur quadratique moyenne est égale à **zéro**.

4.3.4 Mesure de similarité structurelle (SSIM)

Le SSIM (Structural Similarity Index) est un modèle basé sur la perception qui considère la dégradation de l'image comme un changement perçu dans les informations structurelles, tout en

intégrant également des phénomènes perceptifs importants, y compris des termes de masquage de luminance et de masquage de contraste et contours. Les informations structurales sont l'idée que les pixels ont de fortes interdépendances, en particulier lorsqu'ils sont spatialement proches. Ces dépendances contiennent des informations importantes sur la structure des objets dans la scène visuelle [30].

La formule de calcul de la méthode et le suivant :

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)((\mu_x^2 + \mu_y^2 + c_2))}$$

Pour notre approche la valeur de SSIM d'une image claire Frame69 prise à un certain moment de la vidéo et Frame69encrypted la même image est cryptée est **0.004784575859**.

Cela signifie que les 3 paramètres : Luminance, Contraste et Contours Ils sont très différents. Dans la même formulation, nous avons calculé la valeur SSIM d'une image claire Frame69 et Frame69decrypted la même image après le décryptage est égale à **1.0**, le résultat est les deux images sont identiques à travers luminance, contraste et contours.

4.3.5 L'entropie

L'entropie est définit l'information moyenne par symbole ou entropie par :

$$S = - \sum_{i=1}^n p_i \log(p_i)$$

Tel que la quantité d'information est attribuée à un symbole i émis par une source stationnaire avec une probabilité p_i .

La valeur de l'entropie doit être très proche de 8, principe d'utilisation pour la valeur de l'entropie comme suit si l'entropie est inférieure à 8 dans ce cas, on peut dire que il existe des degrés de prévisibilité, donc on ne peut pas assurer la sécurité maximale contre l'analyse de type statistique [31].

Le tableau ci-dessous répertorie les valeurs de l'entropie des images claires et leurs chiffrées en utilisant notre proposition.

La taille des images prises à partir de la vidéo est 560×230 , et leurs type est PNG.

Nom de l'image	Entropie de l'image claire	Entropie de l'image chiffrée
Frame1	7.698688465412056	7.99862940108750
Frame2	7.6988207753411935	7.99869100772219
Frame3	7.698672022982991	7.99865186688798
Frame4	7.700683085588485	7.99862079988941
Frame5	7.699918992561466	7.99864471917972
Frame6	7.700433735562282	7.99862360778767
Frame7	7.699809882811268	7.99865424058576
Frame8	7.698566035797114	7.99865981040799
Frame9	7.699229107222317	7.99860806401198
Frame10	7.697798122512212	7.99861865085485
Frame11	7.697943726591249	7.99862655317568
Frame12	7.697372271804132	7.99858494169455
Frame13	7.697049297819112	7.99860585072708
Frame14	7.6975507102074925	7.99862163012429
Frame15	7.6990024242469985	7.99862141652109
Moyenne de l'entropie	7.69876924376402	7.99863083737718

TABLE 4.1 – Les valeurs de l'entropie des images claires et leurs chiffrées.

Le résultat montre qu'après simuler de 15 images, la valeur moyenne de l'entropie des images chiffrées est 7.99863, c'est-à-dire il est plus proche à la valeur 8. Cela montre qu'il est difficile, ou dire que c'est impossible d'avoir la prévisibilité.

4.3.6 La corrélation entre les pixels adjacents

En plus de l'analyse, nous avons également analysé les corrélations des pixels adjacents dans le cadre des images originales et cryptées.

Le calcul du coefficient de corrélation entre les pixels adjacents nous donne une idée sur la

capacité de notre algorithme de cryptage à résister aux attaques. Le coefficient de corrélation est défini par la formule suivante [32] :

$$r(x, y) = \frac{1}{n-1} \sum_{i=1}^n \left(\frac{x_i - \bar{x}}{s_x} \right) \left(\frac{y_i - \bar{y}}{s_y} \right) = \frac{cov(x, y)}{s_x s_y}$$

telque

$$cov(x, y) = \frac{1}{n-1} \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})$$

$$s_x^2 = cov(x, x) \quad ; \quad s_y^2 = cov(y, y)$$

Le principe d'utilisation pour la valeur de corrélation proche à 1, cela signifie que l'image claire et d'image chiffrée sont très dépendantes. Et aussi si la valeur de corrélation proche à 0 $\pm$, cela signifie que l'image chiffrée et l'image claire ne sont pas corrélés.

Le tableau ci-dessous répertorie les corrélations des images claires et leurs chiffrées en utilisant le l'algorithme proposé.

Ainsi, plus faible est la valeur de corrélation, la qualité de cryptage est meilleure.

Nom de l'image	correlation images déchiffrées	correlation images chiffrées
Frame1	1	-0.079214235970988
Frame2	1	0.079109651704113
Frame3	1	0.079143448318949
Frame4	1	0.078544728230497
Frame5	1	-0.078354279254356
Frame6	1	-0.077155804025838
Frame7	1	0.077699796846926
Frame8	1	0.078074733284153
Frame9	1	-0.077334178194987
Frame10	1	0.077596357480703
La moyenne	1	0.078222721331151

TABLE 4.2 – Les valeurs de La corrélation des images claires et leurs chiffrées.

Le résultat analysé les corrélations des pixels adjacents dans le cadre des images originales et cryptées montre qu'après simuler de 10 images, la valeur moyenne des corrélations d'une image chiffrée est **0.078222721331151**, c'est-à-dire il est plus proche à la valeur 0. Cela montre que les pixels adjacents après le cryptage n'ont pas de corrélation.

Les figures suivantes montrent les distributions de deux pixels adjacents horizontaux, verticaux et diagonaux pour ce qui est des images originales et des images chiffrées.

Il ressort de ces figures que dans le cas des images originales, les pixels adjacents ont des corrélations fortes et s'alignent sur la première bissectrice. Par contre, dans le cas des images chiffrées, les pixels adjacents sont disséminés presque de manière aléatoire. D'une manière générale, l'observation des pixels fortement disséminés renvoi à un algorithme robuste à toute attaque statistique.

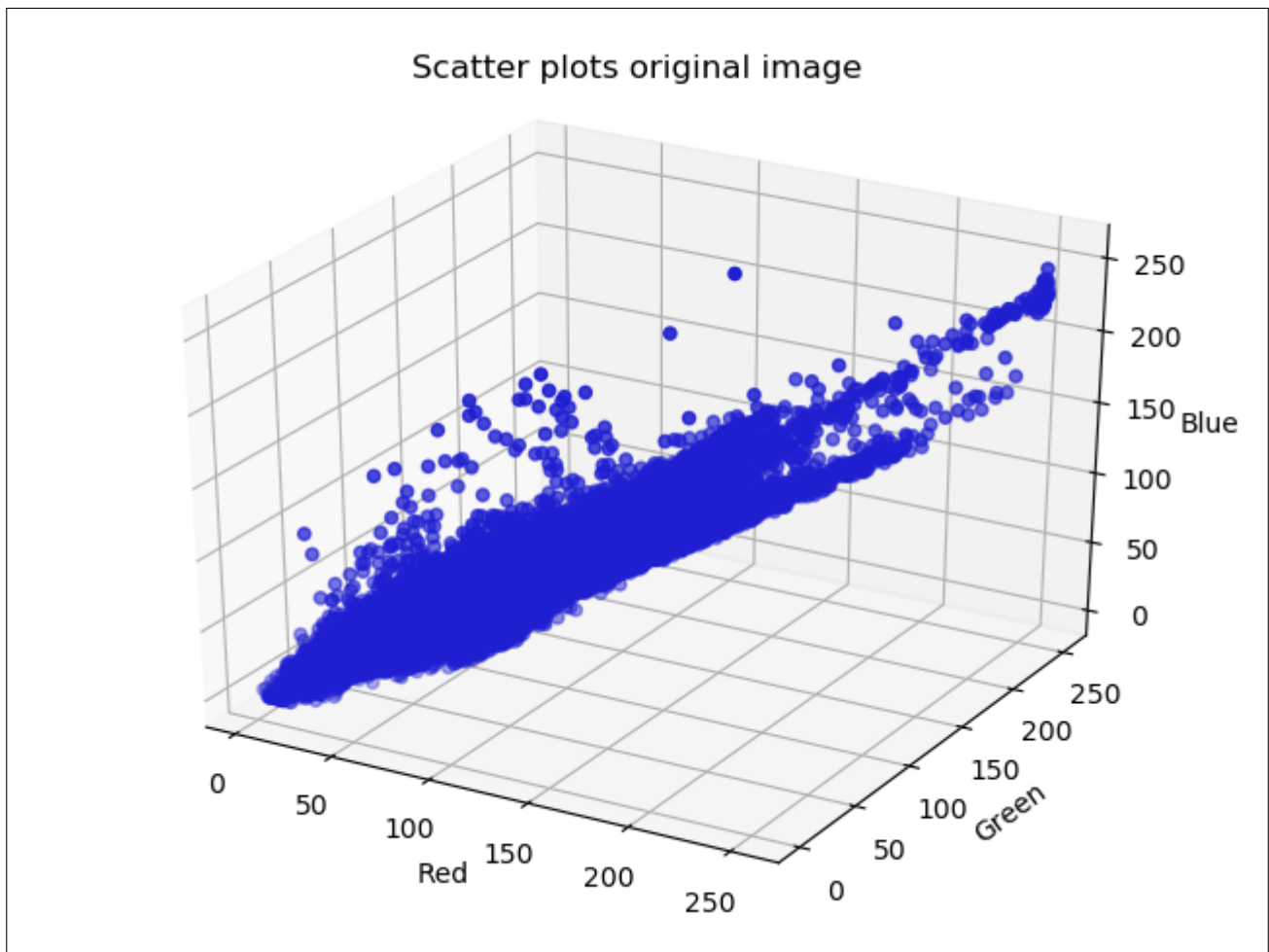


FIGURE 4.13 – Les distributions des pixels adjacents sur les images originales.

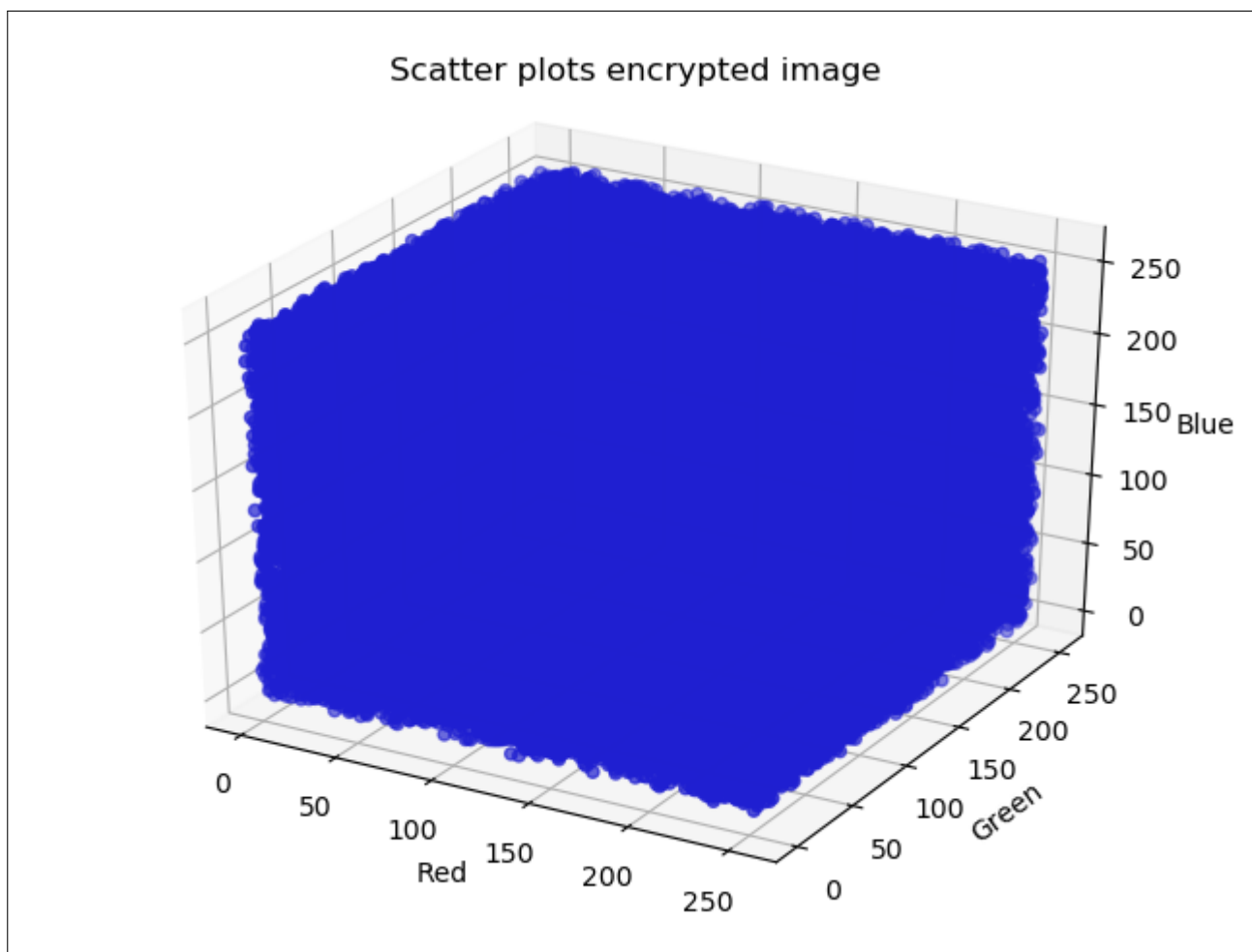


FIGURE 4.14 – Les distributions des pixels adjacents sur les images cryptées.

4.4 Discussion

A partir de la Table, nous observons qu'il y a quelques changements dans les différentes propriétés du fichier vidéo, le grand changement concerne la taille de la vidéo, l'originale est 238 ko par contre la taille de fichier vidéo après le chiffrement est 3.82Mb et leur déchiffrement est 566ko.

	Vidéo original	Vidéo chiffrée	Vidéo déchiffrée
Taille	238 <i>kb</i>	3.82 <i>mb</i>	566 <i>kb</i>
Nombre de frames	85	85	85
Longueur	5	5	5
Taux de frame(fps)	15	15	15

TABLE 4.3 – Les analyses des vidéos avant et après le chiffrement.

Nous n'avons pas remarqué de changement dans la durée de la vidéo par ce que est directement au taux des frames. Plus taux des frames est élevé, plus la longueur est courte, mais dans l'exemple, nous avons une valeur constante de taux.

Explication d'augmenter la taille de la vidéo chiffrée est très logique parce que la taille de chaque frame après le chiffrement est augmenté.

Outre les modifications apportées à la vidéo après le processus de cryptage et de décryptage. Les graphiques montrent que la précision n'a pas été affectée par le retour au facteur SSIM. La même chose est vraie pour l'index PSNR.

4.5 Conclusion

Nous avons présenté dans ce chapitre notre application réalisée, et les différents outils nécessaires utilisés dans son développement. Nous avons aussi terminé par un ensemble d'analyses et tests pour les résultats expérimentaux qui montrent la résistance de notre algorithme de chiffrement.

Tel que par l'analyse de notre méthode, possède un grand espace de clés et une sécurité de haute niveau. Ainsi l'analyse prouve la sécurité et l'efficacité.

CONCLUSION GÉNÉRALE

Le grand progrès et l'utilisation vaste des vidéos numériques dans la technologie des réseaux, en particulier l'internet, pour cela la confidentialité de vidéo numérique est devenue indispensable.

Mais le problème se pose ici est comment peut-on concevoir un système de chiffrement efficace pour assurer la sécurité de ce type de données et de plus, la taille des vidéos numériques sont souvent de grande taille, comment assurer également la rapidité du système ?

La méthode que nous présentons dans ce mémoire est une nouvelle approche de chiffrement des données volumineuses en particulier pour les vidéos numériques est basée sur l'hybridation consiste en une méthode symétrique (algorithme XOR pour chiffrer la vidéo) et asymétrique (algorithme Diffie-Helman pour génère clé partagé) est responsable de la création du pool via une équation spéciale où le Pool est contient ensemble de clés.

Pour analyser la méthode, nous avons effectué un ensemble de tests l'espace de clés, l'histogramme, l'entropie, la corrélation entre les pixels adjacents et mesure de distorsion ... etc qui montrent la bonne robustesse de notre proposition où les résultats expérimentaux montrent cela clairement.

Comme perspective à ce travail, nous allons améliorer notre approche est intéressé par le facteur de la rapidité de chiffrement tout en conservant d'un niveau des performances. Les contributions futures, Nous souhaitons continuer en utilisant de nouvelles idées telles que l'introduction du calcul parallèle dans notre méthode de chiffrement et proposer d'un algorithme de sélection de pixels spécifique pour les chiffrer.

BIBLIOGRAPHIE

- [1] B. Schneier, *Cryptographie appliquée : Protocoles, algorithmes et codes sources en C*, Jan. 1 1990.
- [2] B. Collard, *La cryptographie dans l'antiquité gréco-romaine*, 2004, vol. 7.
- [3] B. Schneier, *Cryptographie appliquée*, 2017.
- [4] D. Barsky *et al.*, “Cryptographie paris 13,” 2010.
- [5] K. Brauer, “Authentication and security aspects in an international multi-user network,” 2011.
- [6] T. BEKKOUCHE, “Développement et implémentation des techniques de cryptage des données basées sur les transformées discrètes,” Ph.D. dissertation, 2018.
- [7] V. Patidar, N. Pareek, and K. Sud, “A new substitution–diffusion based image cipher using chaotic standard and logistic maps,” *Communications in Nonlinear Science and Numerical Simulation*, vol. 14, no. 7, pp. 3056–3075, 2009.
- [8] R. Dumont, “Cryptographie et sécurité informatique,” *Eyrolles*, vol. 2010, 2009.
- [9] D. Stinson, S. Vaudenay, G. Avoine, and P. Junod, *Cryptographie-Théorie et pratique/2ème édition*. vuibert, 2003, no. BOOK.
- [10] S. Vaudenay, *A classical introduction to cryptography : Applications for communications security*. Springer Science & Business Media, 2006.
- [11] A. Canteaut and F. Lévy-dit Véhel, “La cryptographie moderne,” *Revue Armement*, 2001.
- [12] S. Kallam, “Diffie-hellman : Key exchange and public key cryptosystems,” *Master degree of Science, Math and Computer Science, Department of India State University, USA*, pp. 5–6, 2015.

- [13] (2018, jul. 28) Différence entre le chiffrement par bloc et le chiffrement par flot. [Online]. Available : <https://waytolearnx.com/2018/07/difference-entre-le-chiffrement-par-bloc-et-le-chiffrement-par-flot.html>
- [14] I. SOUICI, “Cryptographie nouvel algorithme de chiffrement evolutionnaire basé occurrences (aceo),” Ph.D. dissertation, 2008.
- [15] A. Ounzar and I. Nini, “Cryptage complet/partiel d’une image/vidéo,” Master’s thesis, 2015.
- [16] D. Battikh, “Sécurité de l’information par stéganographie basée sur les séquences chaotiques,” Ph.D. dissertation, Rennes, INSA, 2015.
- [17] M. OUAMRI *et al.*, “Sécurité et compression de l’information multimédia,” Ph.D. dissertation, 2016.
- [18] A. T. Ho and S. Li, *Handbook of digital forensics of multimedia data and devices*. John Wiley & Sons, 2015.
- [19] F. De Rango, *Digital Video*. BoD—Books on Demand, 2010.
- [20] J. Shah, D. Saxena *et al.*, “Video encryption : A survey,” *arXiv preprint arXiv :1104.0800*, 2011.
- [21] B. CHENENE, “Chiffrement des vidéos numériques,” Master’s thesis, UNIVERSITE MOHAMED BOUDIAF-M’SILA FACULTE DES MATHEMATIQUES ET DE L ... , 2019.
- [22] A. Aimeur, “Conception et implémentation d’un système hybride pour la sécurité de données : application aux images numériques,” Master’s thesis, UNIVERSITE MOHAMED BOUDIAF-M’SILA, 2017.
- [23] M. Paliwal and S. Hussain, “Selective video encryption using bit xor technique,” *International Journal for Innovative Research in Science & Technology*, vol. 2, no. 1, pp. 177–183, 2015.
- [24] S. C. Iyer, R. Sedamkar, and S. Gupta, “A novel idea of video encryption using hybrid cryptographic techniques,” in *2016 International Conference on Inventive Computation Technologies (ICICT)*, vol. 3. IEEE, 2016, pp. 1–5.
- [25] A. Chadha, S. Mallik, A. Chadha, R. Johar, and M. M. Roja, “Dual-layer video encryption using rsa algorithm,” *arXiv preprint arXiv :1509.04387*, 2015.
- [26] M. Lutz, *Learning python : Powerful object-oriented programming*. " O’Reilly Media, Inc.", 2013.
- [27] L. Olivier, *Initiation à MATLAB*, 2009.

- [28] T. Acharya and A. K. Ray, *Image processing : principles and applications*. John Wiley & Sons, 2005.
- [29] U. Sara, M. Akter, and M. S. Uddin, “Image quality assessment through fsim, ssim, mse and psnr—a comparative study,” *Journal of Computer and Communications*, vol. 7, no. 3, pp. 8–18, 2019.
- [30] D. Ślęzak, *Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing : 10th International Conference, RSFDGrC 2005, Regina, Canada, August 31-September 2, 2005, Proceedings*. Springer Science & Business Media, 2005, vol. 3641.
- [31] J. Adamek, *Foundations of coding : Theory and applications of error-correcting codes with an introduction to cryptography and information theory*. John Wiley & Sons, 2011.
- [32] P. Y. Chen, M. Smithson, and P. M. Popovich, *Correlation : Parametric and nonparametric measures*. Sage, 2002, no. 139.