

REPUBLIQUE DU CAMEROUN

Paix – Travail – Patrie

UNIVERSITE DE YAOUNDE I

FACULTE DES SCIENCES

DEPARTEMENT DE PHYSIQUE

CENTRE DE RECHERCHE ET DE

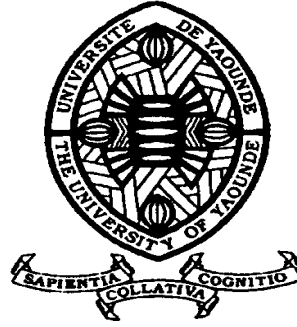
FORMATION DOCTORALE EN

SCIENCES,

TECHNOLOGIES ET GEOSCIENCES

Laboratoire d'Énergie et des Systèmes

Électriques et Électroniques



REPUBLIC OF CAMEROUN

Peace – Work – Fatherland

UNIVERSITY OF YAOUNDE I

FACULTY OF SCIENCE

DEPARTMENT OF PHYSICS

POSTGRADUATE SCHOOL FOR

SCIENCES, TECHNOLOGY AND

GEOSCIENCES

Laboratory of Energy and Electrical

and Electronics Systems

**NOUVEAUX CRYPTOSYSTÈMES
BASÉS SUR LE MIXAGE ET LA
FUSION DES CARTES DE DONNÉES**

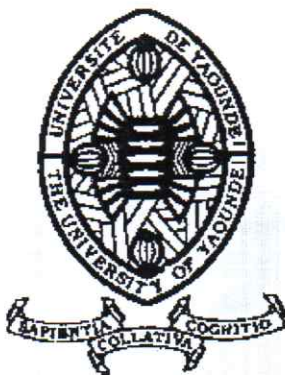
Thèse présentée et soutenue en vue de l'obtention du Diplôme de
Doctorat/Ph.D. de Physique

Par : **ABANDA Antoine Yannick**
Master of Sciences en Physique

Sous la direction de
SIEWE SIEWE Martin
Maître de Conférences UYI
ELE Pierre
Maître de Conférences UYI

Année Académique : 2018





DÉPARTEMENT DE PHYSIQUE
DEPARTMENT OF PHYSICS

ATTESTATION DE CORRECTION DE LA THÈSE DE
DOCTORAT/Ph.D

Nous, Professeur **WOAFO Paul** et Professeur **SIEWE SIEWE Martin**, respectivement Président du jury et Examineur de la Thèse de Doctorat/Ph.D de Monsieur **ABANDA Antoine Yannick**, Matricule **09W1254**, préparée sous la supervision du Professeur **TIEDEU Alain**, intitulée : « **Nouveaux cryptosystèmes basés sur le mixage et la fusion de données** », soutenue le **Judi 20 Septembre 2018**, en vue de l'obtention du grade de Docteur/Ph.D en Physique, Spécialité **Systèmes Electriques et Electroniques**, attestons que toutes les corrections demandées par le jury de soutenance ont été effectuées.

En foi de quoi, la présente attestation lui est délivrée pour servir et valoir ce que de droit.

Fait à Yaoundé le **3 0 OCT 2018**

Examineur

Prof. SIEWE SIEWE Martin



Le Chef de Département de Physique

Prof. NDJAKA Jean-Marie
Bienvenu

Le Président du jury

Prof. WOAFO Paul

Nouveaux cryptosystèmes basés sur le mixage et la
fusion de cartes de données

ABANDA Antoine Yannick

Septembre 2018

Dédicace

À mon père et ma mère.

À ma très chère Clarisse.

À mes enfants.

Aux grandes familles Mbarga et Abanda.

Remerciements

Tous mes remerciements à Dieu notre Seigneur et Père qui nous accorde ses bienfaits.

Je tiens particulièrement à remercier le Professeur Alain Tiedeu pour son encadrement, pour son soutien autant financier que moral, plus que ça il m'a encadré comme un père.

Je tiens également à remercier Madame Tiedeu pour ses encouragements et, ses petits plats que je dégustais avec un très grand appétit, surtout après des longues heures de travail.

Le professeur Kofané Timoléon crépin pour ses précieux conseils et son appui.

Je veux également remercier le Professeur Essimbi Zobo Bernard pour ses précieux conseils.

Je veux remercier le Professeur Jean-Sire Armand Eyebé Fouda pour ces précieux conseils, mais également pour son écoute lors de nos échanges sur les différents sujets qu'on a pu avoir.

Je remercie le Docteur Bertrand Bodo pour ses précieux conseils, pour sa bonne humeur et, pour son aide.

Je remercie le Docteur Kenfack Gutenbert pour son soutien, pour son appui et, pour son aide.

Je remercie le Docteur Poné et le Docteur Kom Guillaume pour leur soutien.

Je remercie le Professeur Ben-Bolie pour son aide précieuse.

Je remercie les membres du laboratoire d'électronique ainsi que ceux de l'école doctorale pour m'avoir accepté en thèse de Doctorat.

Je remercie les Enseignants du Département de Physique de la Faculté de Sciences de l'Université de Yaoundé 1 pour leur formation académique et leur encadrement.

Je remercie l'Ecole Nationale Supérieure Polytechnique pour leur soutien.

Je remercie le groupe AFSIN qui m'a permis de présenter mes travaux pour la première fois dans une conférence.

Le remercie Mr Mvogo Jean Baptiste Doctorant à l'Université de Yaoundé 1, le Docteur Alima Marie Roland Joel, Kamdeu Pascal, Yepdia Mariel, Fotso Nguemo Thierry mes collaborateurs de travail et amis.

Ma reconnaissance va également à l'ensemble des membres du jury de soutenance de cette thèse pour leurs précieux retours et conseils ainsi que leurs enrichissantes remarques. Je pense notamment aux Professeurs Ndjaka Jean-Marie Bienvenu, Siewe Siewe Martin, Moukam Kakmeni François Marie et Ele Pierre. Je remercie également le Professeur Wofo Paul, pour avoir accepté de présider ce jury.

Je remercie également ma famille pour leur soutien inconditionnel et toutes les personnes qui ont contribué à ce travail.

Je remercie la famille Lyeb pour leur accompagnement et leur affection.

Je remercie les familles Mbama, Ngadena et Zoa pour leurs précieux conseils, leur joie de vivre et leur soutien.

Je remercie particulièrement ma douce et tendre Atyame Clarisse Claire, ainsi que nos merveilleux enfants Abriel, Audrey et Tyzianna pour leur amour et leur soutien sans faille durant ces difficiles années de thèse.

Je remercie mes amis et connaissances pour leur soutien affectif et émotionnel.

Table des matières

Dédicace	i
Remerciements	ii
Liste des figures	viii
Liste des tableaux	x
Liste des abréviations	xi
Résumé	1
Abstract	2
Introduction générale	3
Chapitre 1 Notions de base sur le cryptage d'images	7
1.1 Définition et survol historique de la cryptographie	7
1.2 Concepts de la cryptographie	8
1.3 Caractéristiques des méthodes de cryptage	9
1.3.1 Algorithme à clef secrète	10
1.3.1.1 Le mode ECB	12
1.3.1.2 Le mode CBC	13
1.3.1.3 Mode CFB	16
1.3.1.4 Mode OFB	16
1.3.1.5 Mode CTR	17
1.3.2 Algorithme à clef publique	17
1.3.2.1 Générateurs de données aléatoires et pseudo-aléatoires	18
1.3.2.2 Les générateurs obtenus par procédés algorithmiques	19
1.3.2.3 Les générateurs physiques	19
1.3.3 Chaos et Cryptographie	20

1.3.4	Cryptage par cartes chaotiques	21
1.3.4.1	Masquage chaotique	22
1.3.4.2	Cryptage par modulation paramétrique	23
1.3.4.3	La commutation chaotique	24
1.3.4.4	Cryptage par injection	24
1.3.5	Diffusion et confusion	26
1.3.5.1	Domaine spatial	26
1.3.5.2	Domaine fréquentiel	28
1.3.6	Cryptanalyse	32
3.5.8.1	Attaque sur texte chiffré seul (Ciphertext only attack)	32
3.5.8.2	Attaque à texte clair connu (Known plaintext attack)	32
3.5.8.3	Attaque à texte clair choisi (Chosen plaintext attack)	32
3.5.8.4	Attaque à texte chiffré choisi (Ciphertext attack)	32
3.5.8.5	Attaque adaptative à texte chiffré choisi (Adaptative Cipher- text attack)	33
1.3.7	Autres types d'attaques	33
1.3.7.1	Attaque exhaustive ou brute	33
1.3.7.2	Attaque différentielle	33
1.3.7.3	Attaque par dictionnaire	33

Chapitre 2 Étude méthodologique du cryptage chaotique : métriques et outils statistiques d'analyse 34

2.1	Définition d'un système dynamique chaotique	34
2.1.1	Représentation mathématique d'un système dynamique	35
2.1.2	Techniques d'identification et de prédiction du chaos	36
2.1.2.1	Techniques d'identification du chaos	36
2.1.2.2	Techniques de prédiction du chaos	37
2.2	Méthodologie de cryptage d'images par chaos	37
2.2.1	Choix de la carte chaotique	38
2.2.1.1	Test binaire 0-1	39
2.2.1.2	Entropie de permutation	40
2.2.1.3	Méthode 3ST	40
2.2.2	Présentation des paramètres et génération de clef de cryptage	43
2.2.3	Fonctions de cryptage et discrétisation	45

2.2.3.1	Diffusion de l'image	45
2.2.3.2	Confusion de l'image	46
2.2.4	Fonctions de hachage	47
2.2.5	Chiffrements affines généraux	48
2.2.6	Courbes elliptiques	49
2.2.7	Transformée de Fourier et transformée de Fourier discrète	50
2.3	Outils d'analyse de sécurité	52
2.3.1	Analyse statistique	52
2.3.2	Analyse par histogramme	53
2.3.3	Analyse par Corrélation	54
2.3.4	Analyse différentielle	55
2.3.5	Analyse de la sensibilité de la clef	55
2.3.6	Analyse de l'espace des clefs	56
2.3.7	Analyse de la complexité en temps	56
2.3.8	Cryptanalyse	56
2.3.9	Méthodes numériques	57

Chapitre 3 Nouveaux cryptosystèmes basés sur le mixage et la fusion de cartes de données : Colpitts-Duffing, et Duffing-Hartley

3.1	Oscillateurs utilisés	60
3.1.1	Oscillateur de Colpitts	60
3.1.2	Oscillateur de Duffing	61
3.1.3	Oscillateur de Hartley	63
3.2	Description de la méthode de cryptage utilisant une carte chaotique	65
3.3	Analyse du cryptosystème	66
3.3.1	Analyse de la clef secrète	66
3.3.2	Analyse de la sensibilité de la clé	67
3.3.3	Analyse différentielle	68
3.3.4	Analyse de la vitesse	68
3.3.5	Histogramme des images	68
3.3.6	Analyse par corrélation	69
3.4	Cryptage par mixage des deux cartes	70
3.4.1	Technique de combinaison pour la génération de la carte chaotique	71
3.4.2	Fonction de transformation ou de brouillage	72

3.5	Analyses de la sécurité	72
3.5.1	Analyse de la clef secrète	72
3.5.2	Sensibilité de la clef	73
3.5.3	Analyse statistique	74
3.5.4	Analyse par histogramme	74
3.5.5	Analyse par corrélation	77
3.5.6	Analyse différentielle	79
3.5.7	Analyse de la vitesse	79
3.5.8	Analyse du système par Cryptanalyse	79
3.5.8.1	Attaque en texte clair choisi (CPA)	80
3.5.8.2	Attaque à texte chiffré choisi (CCA)	81
3.5.9	Etude comparative des cryptosystèmes	82
3.5.9.1	Comparaison entre les cryptosystèmes, celui utilisant une carte, et l'autre utilisant deux cartes chaotiques	82
3.5.9.2	Comparaison avec d'autres méthodes	83
3.6	Fusion de cartes	84
3.6.1	Algorithme de cryptage	84
3.6.1.1	Technique de fusion	84
3.6.1.2	Etude de la carte chaotique obtenue	84
3.6.1.3	Description de la méthode PLSE	85
3.6.1.4	Fonction de transformation	87
3.6.2	Résultats et performances du système de cryptage	87
3.6.3	Histogrammes des images	88
3.6.4	Autres analyses	88
3.6.5	Cryptanalyse du cryptosystème	90
3.6.6	Comparaison avec de nouvelles méthodes	91
	Conclusion générale et perspectives	92
	Références bibliographiques	94
	Liste des publications	104
	Conférences et ateliers	105

Liste des figures

Figure 1	Substitution et Permutation	8
Figure 2	Procédure de base pour le cryptage à clef secrète	10
Figure 3	Échantillon de chiffrement ECB	12
Figure 4	Échantillon de chiffrement CBC	13
Figure 5	Procédure de chiffrement par blocs	14
Figure 6	Décomposition d'un tour	15
Figure 7	Schéma synoptique de Feistel	15
Figure 8	Procédure de base du cryptage à clef publique	17
Figure 9	Structure de communication sécurisée par chaos	22
Figure 10	Schéma synoptique de la technique de masquage chaotique	23
Figure 11	Schéma synoptique de la technique de cryptage par modulation	23
Figure 12	Schéma synoptique de la technique de cryptage par commutation	24
Figure 13	Schéma synoptique de la technique de cryptage par injection	25
Figure 14	Permutation des pixels d'une image utilisant la technique dite du Chat d'Arnold	27
Figure 15	Schéma synoptique de la technique de cryptage par injection	28
Figure 16	Évolution de la dynamique du paramètre de détection du chaos sur la carte logistique	43
Figure 17	Principe d'une fonction de hachage	47
Figure 18	Géométrie explicite des EC	50
Figure 19	Masquage par bruit additif de l'image	54
Figure 20	Distribution des paires de pixels adjacents horizontalement dans l'image Lena	54
Figure 21	Système de Colpitts	61
Figure 22	Portait de phase de Duffing en l'absence du second membre	62
Figure 23	Oscillateur de Duffing avec second membre	63
Figure 24	Section de Poincaré de l'Oscillateur de Duffing	63

Figure 25	Système de Hartley	64
Figure 26	Dynamique des exposants de Lyapunov	65
Figure 27	Organigramme du schéma de cryptage	66
Figure 28	Cryptage de l'image Lena avec la plus petite variation paramétrique 10^{-15} pour une conversion double précision	67
Figure 29	Décryptage de l'image cryptée par deux clés relativement proches	68
Figure 30	Histogrammes	69
Figure 31	Corrélation diagonale des pixels adjacents	70
Figure 32	Organigramme du système de cryptage proposé	71
Figure 33	Images tests	72
Figure 34	Cryptage de l'image Mandrill avec la plus petite variation paramé- trique 10^{-15} pour une conversion double précision	73
Figure 35	Images tests	74
Figure 36	Histogrammes des images de Lena, image cryptée de Lena, Nebula m83, et image cryptée de Nebula m83	75
Figure 37	Images tests de Mandrill, image cryptée de Mandrill, Clown et image cryptée de Clown	75
Figure 38	Corrélation anti diagonale des pixels adjacents de Lena, image cryptée de Lena, Nebula m83, et image cryptée de Nebula m83	77
Figure 39	Corrélation anti diagonale des pixels adjacents de Mandrill, image cryptée de Mandrill, Clown et image cryptée de Clown	78
Figure 40	Images tests de Barbara et neutre	80
Figure 41	Test par CPA des images de Barbara et neutre	81
Figure 42	Images tests neutre et cryptée de Barbara	82
Figure 43	Test par CCA des images cryptée de Barbara et décryptée de l'image neutre	82
Figure 44	Organigramme du test par la PLSE de la série de données X_t	86
Figure 45	Images tests de Mandrill, Barbara et Lena	87
Figure 46	Cryptage de l'image de Barbara	88
Figure 47	Histogrammes des images originales et cryptées de Barbara, Lena et Mandrill	89
Figure 48	Analyse par corrélation des images originales et cryptées de Barbara, Lena et Mandrill	90
Figure 49	Attaque par CPA l'image neutre et celle de Barbara	90

Liste des tableaux

Tableau 1	Analogie entre systèmes cryptographiques et chaotiques	21
Tableau 2	Calcul du <i>NPCR</i> et de l' <i>UACI</i> pour chaque séquence de données issues des oscillateurs de Colpitts et de Duffing	68
Tableau 3	Variances des histogrammes de l'image originale et de l'image cryptée de Lena	69
Tableau 4	Variances des histogrammes des images cryptées pour deux clés légèrement différentes	69
Tableau 5	Coefficient de corrélation en fonction de chaque oscillateur	70
Tableau 6	Propriétés statistiques des images	74
Tableau 7	Variances des histogrammes des images en clair et des images chiffrées	76
Tableau 8	Variances des histogrammes des images cryptées pour deux clés légèrement différentes	76
Tableau 9	Coefficient de corrélation des images	78
Tableau 10	Entropie des images cryptées	78
Tableau 11	Sensibilité à l'attaque différentielle par le calcul du <i>NPCR</i> et de l' <i>UACI</i>	79
Tableau 12	Comparaison entre les cryptosystèmes	83
Tableau 13	Comparaison des méthodes de cryptage utilisant l'image de Lena comme image test	83
Tableau 14	Détection de chaos par le calcul de l'entropie sur les données utilisées par la méthode de PLSE	87
Tableau 15	Propriétés statistiques des images	87
Tableau 16	Propriétés statistiques des images	88
Tableau 17	Propriétés statistiques des images	89
Tableau 18	Comparaison avec quelques cryptosystèmes	91

Liste des abréviations

AES :	Advanced Encryption Standard
ASCII :	American Standard Code for Information Interchange
BCR :	Bit Chaotique Réarrangé
CBC :	Cipher Block Chaining
CCA :	Chosen Cipher text Attack
CFB :	Cipher FeedBack
CTR :	CounTeR
CSK :	Chaos Shift Keying
DCT :	Discrete Cosine Transform
DES :	Data Encryption Standard
EC :	Elliptic Curve
ECB :	Electronic Code Book
IBM :	International Business Machines
LFBR :	Linear FeedBack Register
MAE :	Mean Absolute Error
MD5 :	Message Digest 5
MSE :	Mean Square Error
NPCR :	Number of Pixels Change Rate
OFB :	Output FeedBack
PC :	Personal Computer
PCA :	Pixel Chaotique Aléatoire
PE :	Permutation d'Entropie
PLSE :	Permutation Largest Slope Entropy
PPCM :	Plus Petit Commun Multiple
PSNR :	Peak Signal to Noise Ratio
PWLCM :	Piece Wise Linear Chaotic Map
RC4 :	Rivest Cipher 4

RGB :	Red Green and Blue
RK4 :	Runge Kutta d'ordre 4
RSA :	Rivest Shamir Adleman
RVB :	Rouge Vert Bleu
SSIM :	Structural SIMilarity
UACI :	Unified Average Change Intensity
WIFI :	Wirless Fidelity

Résumé

Deux nouveaux algorithmes à clé privée sont proposés. Ces algorithmes traitent du cryptage des signaux 2D. Ils sont proposés pour résoudre certains problèmes liés à la vulnérabilité de certaines structures algorithmiques récentes. Ces cryptosystèmes sont bâtis autour de la fusion et du mixage des oscillateurs chaotiques de Colpitts, Duffing et Hartley. De plus, dans le processus de cryptage nous introduisons une fonction mathématique simple mais qui a une grande capacité de brouiller une image. Les algorithmes mis au point ont été testés sur des images standards de la communauté scientifique. Les résultats obtenus sont satisfaisants en comparaison à ceux de la littérature.

Mots clés : Cryptage, mixage, fusion, cryptosystème, carte chaotique

Abstract

Two new private key encryption are proposed. These algorithms are used to encrypt 2D signals. They are proposed to solve some problems related to the vulnerability of some recent algorithmic structures. These cryptosystems based on fusion and mixing of the chaotic oscillators of Colpitts, Duffing and Hartley. Furthermore, in the encryption process a simple mathematical function which has a great capacity to shuffle an image was introduced. The algorithms developed were tested on standard images from the scientific community. The results obtained are satisfactory in comparison with those presented in literature.

Key words : Encryption, mixing, fusion, cryptosystem, chaotic map

Introduction générale

Le besoin de confidentialité dans les échanges de données non classifiées a accéléré le développement de la cryptographie. Le commerce en ligne, la téléphonie mobile, le WIFI, la consultation des données bancaires sur internet, les transactions par carte à puces, les dossiers médicaux électroniques sont tous des domaines utilisateurs de cette science. Ces données ne sont pas toujours sous forme textuelle. Elles peuvent être sous forme d'images numériques, vidéo ou audio. La transmission d'images est l'une des opérations les plus menées du 21e siècle, ce qui rend leur sécurité capitale et essentielle.

Présentation et situation du thème de recherche

L'utilisation des cartes chaotiques comme générateurs de nombres aléatoires dans les cryptosystèmes s'est renforcée ces dernières années. Les systèmes cryptographiques à base de chaos analogique (Shannon, 1949; Matthews, 1989; Gonzalo et Shujun, 2006; Pecora et Carroll, 1990; Sobhy et Shehata, 2001a, 2001b; Takougang Kingni et al., 2012; Tchitnga et al., 2013), puis numérique se sont développés de manière exponentielle. De nombreux algorithmes de cryptage basé sur l'utilisation des cartes chaotiques numériques provenant des systèmes chaotiques à une dimension (1D), deux dimensions (2D), trois dimensions (3D), m dimensions ont été proposés $m > 3$ ont été proposés (Sobhy et Shehata, 2001a; Xu et al., 2016). Malheureusement, bon nombre de ces algorithmes ont été cryptanalysés ou cassés (Rhouma et Belghith, 2008; Liu et al., 2014).

Sobhy et Shehata (2001a) ont utilisés le système de Lorenz pour générer des cartes chaotiques utilisés pour le cryptage du son et de l'image, mais les mêmes auteurs ont montré que si on peut connaitre le système chaotique utilisé, il est possible de révéler toutes les séquences chaotiques à partir de la représentation retardée de leur système, et de quelques outils logiciels, et par conséquent de retrouver les conditions initiales du système utilisé.

Afin d'améliorer les algorithmes de cryptage utilisant une carte chaotique, Zhang et Jin (2008) ont proposé une technique de chiffrement d'images utilisant une carte provenant d'un système à trois dimensions dans lequel la clef secrète est composée des trois conditions

initiales du système utilisé. Cette clé est ensuite subdivisée en deux sous clés de 258 bits, l'une des sous clef est utilisée pour générer des séquences d'échantillonnage, et l'autre sous clef est utilisée pour le prétraitement de l'image dans un processus de compression en utilisant une carte logistique. L'algorithme possède une bonne probabilité de succès, une bonne complexité de calcul, une occupation de mémoire négligeable mais une faiblesse au niveau de l'espace des clefs, ce qui entraîne une vulnérabilité aux attaques. Plus tard, Gao et Chen (2008b, 2008a) ont proposé deux schémas de cryptage d'images basées sur l'hyperchaos, en utilisant le système de Lorenz amélioré (Gao et Chen, 2008b), et le système de Chen (Gao et Chen, 2008a). Les deux cryptosystèmes ont une faible sécurité en termes de sensibilité sur l'image et une vitesse d'exécution de l'algorithme de cryptage faible (Gao et Chen, 2008a) et très faible (Rhouma et Belghith, 2008 ; Arroyo et al., 2009). Guosheng et Guoqiang (2006) ont proposé un algorithme de cryptage en utilisant deux systèmes chaotiques basé sur le secret de ces deux systèmes. L'un des systèmes est utilisé pour générer une séquence chaotique, laquelle est transformée en un flux binaire par une fonction seuil. L'autre système est utilisé pour construire une matrice de permutation. Mais cet algorithme présente un désavantage certain : on ne connaît pas les systèmes chaotiques utilisés. Huang (2012) ont utilisé une fonction de Tchebychev chaotique à deux dimensions pour brouiller les pixels de l'image, mais plus tard, le cryptosystème a montré des failles de sécurité (Wang et Qiang, 2014). Liu et Liu (2014) ont trouvé quelques défauts de sécurité dans le système proposé par El-Latif et Niu (2013). Ce système de chiffrement d'image est basé sur la carte logistique hybride et une courbe elliptique cyclique. Elle est robuste mais elle présente une faible clef de cryptage.

Récemment, Song et Qia (2013) ont présenté un nouvel algorithme de cryptage basé sur un système spatio-temporel et combiné à algorithme non linéaire capable de permuter et de diffuser les pixels d'une image. Bechikh et al. (2015) ont analysé ce codage et ont conclu que la séquence de données de substitution est le même pour chaque paire d'images (image originale/image chiffrée). Wang et Qiang (2014) ont proposé un cryptage d'image basé sur les S-boxes (boîtes de substitution dynamique construites par un système chaotique), malheureusement avec un faible temps d'exécution. Ahmad et Ahmad (2014) ont révélé une faiblesse de ce système, puis l'ont cryptanalysé (Panduranga et al., 2014). Récemment, Zhang et al. (2016) ont proposé une nouvelle approche dans laquelle il considère une image de taille $M \times N$ avec 255 niveaux de gris comme une matrice 3D de taille $M \times N \times 8$ et ont conçu une nouvelle architecture de permutation au niveau du bit. Une autre technique de permutation au niveau du bit associé à la substitution au niveau du pixel utilisant la

carte chaotique du chat d'Arnold a été proposée (Fu et al., 2013), et a été rapidement cryptanalysée et améliorée (Zhang et al., 2015). Curieusement, le système amélioré a été brisé (Chen et al., 2015), les auteurs ayant trouvé des clés de permutation équivalentes dans ce système. Un algorithme de cryptage d'image basé sur le codage de l'ADN et sur une technique d'addition de deux cartes chaotiques a été pour la première fois présenté (Zhang et al., 2010). Elle présente l'avantage d'être nouvelle en possédant un espace de clés suffisamment élevée, mais cette technique a été trouvée non inversible et vulnérable à l'attaque à texte clair connu par l'auteur (Hermassi et al., 2014), pendant que la version améliorée (Liu et al., 2012) a été cryptanalysée (Belazi et al., 2014; Liu et al., 2014). Quelques versions améliorées viendront plus tard malheureusement avec un faible temps d'exécution (Zhou et al., 2014; Jain et Rajpal, 2015; Kumar et al., 2016; Chen et al., 2015; Wua et al., 2015; Enayatifar et al., 2015). Une petite revue de la littérature sur un certain nombre d'algorithmes de chiffrement basés sur le Chaos montre que les principaux problèmes rencontrés sont les suivants :

- Les failles de sécurité (Pareek et al., 2006; Patidar et al., 2010; Zhu et al., 2011; Belazi et al., 2016; El-Latif et Niu, 2013; Parvin et al., 2014; Xu et al., 2016);
- La cryptanalyse relativement facile (Li et al., 2009, 2011; Zhang et Wang, 2014; Liu et Liu, 2014; Norouzi et Mirzakuchaki, 2016);
- L'insensibilité à l'image en clair ou à la clé;
- Un petit espace de clé;
- Un comportement chaotique limité;
- La distribution de données non uniforme à la sortie des séquences chaotiques;
- Le chiffrement non inversible;
- La faible vitesse d'exécution.

Par conséquent, la conception d'un cryptosystème efficient basé sur le chaos, reste un défi pour les chercheurs. D'où la question : Est-il possible de construire un cryptosystème par mixage ou par fusion de cartes chaotiques plus robuste, simple, rapide, efficace, possédant un espace de clef suffisamment large dont les performances statistiques sont améliorées tout en résistant mieux à la cryptanalyse ?

Nous proposons dans ce travail deux algorithmes dans le but de résoudre ce problème.

- Le premier mixe les cartes de données provenant des oscillateurs de Duffing et de Colpitts, couplées à une fonction de brouillage simple et rapide pour générer l'image

cryptée. Nous démontrons au préalable la suprématie en termes de sécurité de ce dispositif par rapport à celui qui serait bâti en utilisant un seul oscillateur ;

- Fort des conclusions tirées du premier cryptosystème, nous en proposons un second où les données provenant des oscillateurs de Duffing et Hartley sont fusionnées et couplées à la même fonction de brouillage.

Dans le premier chapitre nous présentons brièvement la cryptographie, notamment ses concepts et ses caractéristiques. Un bref exposé sur les méthodes de diffusion/permutation nécessaire pour le cryptage d'une image est décrit. Nous présentons aussi quelques cartes chaotiques présentes dans la littérature et le chapitre s'achève par la définition de la cryptanalyse. Le chapitre deux se penche sur les outils utilisés pour construire les cryptosystèmes. Ainsi on y trouve un récapitulatif des méthodes permettant de qualifier un système de chaotique, ou de prédire la présence du chaos dans une séquence de données. Par la suite une méthodologie de cryptage par chaos est détaillée. Il s'agit pour nous de définir les méthodes de choix d'une carte chaotique, de présenter quelques méthodes de construction d'un algorithme de chiffrement d'image, de présenter les fonctions de cryptage possibles, et enfin de passer en revue les métriques utilisées pour évaluer la robustesse d'un cryptosystème. Le troisième chapitre a pour objet l'étude des nouveaux cryptosystèmes proposés. Il expose tout d'abord les différents éléments des dispositifs. Le premier cryptosystème est décrit, évalué et comparé à ceux de la littérature. Nous proposons ensuite un deuxième algorithme de cryptage basé sur la fusion de deux cartes chaotiques. Les fonctions de fusion et de brouillage sont présentées, ainsi que les résultats des tests et de la comparaison de ses performances avec ceux des cryptosystèmes disponibles. Nous clôturons le manuscrit par une conclusion générale ainsi que les perspectives envisagées.

Chapitre 1

Notions de base sur le cryptage d'images

Introduction

De tout temps, l'Homme a entretenu l'idée selon laquelle, dissimuler le contenu de certaines informations était absolument vital pour son évolution. Ainsi pour restreindre l'accès à certaines données à un groupe d'individus, il était nécessaire de trouver des moyens de conserver et de préserver certains paramètres secrets pour que seuls ceux appartenant à ce groupe soient capables d'interpréter ou d'exploiter ces informations. Il s'agissait de permettre à deux ou plusieurs personnes de communiquer à travers un canal peu sûr sans qu'un intrus ne puisse comprendre ce qui est échangé. La cryptographie naît de cette préoccupation.

1.1 Définition et survol historique de la cryptographie

La cryptographie est l'art de dissimuler ou de cacher un message ou un texte. On applique à un *texte en clair* une transformation qui le rend incompréhensible, c'est ce qu'on appelle le *chiffrement*. Le résultat obtenu est appelé *texte chiffré* ou *cryptogramme*. Le *déchiffrement* est l'action qui permet de reconstruire le texte en clair à partir du texte chiffré. L'art de rendre clair un texte *crypté* ou *codé* sans avoir connaissance du procédé utilisé est appelé *cryptanalyse*. L'ensemble formé de la cryptographie et de la cryptanalyse est appelé *cryptologie*. C'est cette science qui définira tous les types de communication au fil du temps.

Dans ce chapitre nous allons expliciter de manière générale les différentes composantes

de la cryptographie et ses concepts. Nous présenterons les deux grandes familles de la cryptographie, ainsi que les différents modes de cryptographie. Nous continuerons ensuite par quelques similitudes entre le chaos et la cryptographie, quelques exemples de cryptographie chaotique, un bref exposé sur quelques cartes chaotiques, une petite revue de la littérature sur le cryptage spatial et le cryptage fréquentiel d'une image, et clôturerons ce chapitre par la définition de la cryptanalyse.

1.2 Concepts de la cryptographie

A la base, la cryptographie classique repose sur deux concepts : la substitution et la transposition ou permutation. Dans la littérature on peut également retrouver d'autres appellations, S-box et P-box (figure 1). Nous reviendrons plus tard sur ces deux notions dans la suite du chapitre.

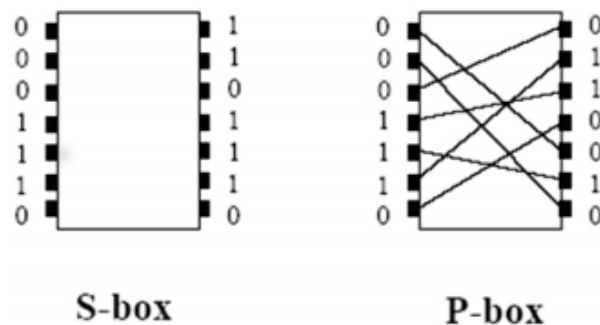


Figure 1 – Substitution et Permutation. Figure tirée de Dumont (2009)

La cryptographie moderne a développé de nombreux buts dont quelques-uns sont repris ci-dessous :

- **La Préservation de la confidentialité** : Il s'agit de rendre inintelligible à tous les intrus tant lors de la conversation qu'au cours de son transfert par un canal de communication, le contenu ;
- **L'intégrité des informations** : s'assurer que cette information n'a pas été altérée, de manière frauduleuse ou accidentelle ;
- **Le non-désaveu** : Il s'agit de garantir l'authenticité de l'échange. L'expéditeur ne peut nier le dépôt de l'information ; le destinataire ne peut nier la remise de l'information ; aucune des deux parties ne peut nier le contenu de l'information ;
- **L'identification** : qui consiste à s'assurer que le destinataire est bien légitime par le biais d'une garantie (signature, certification) ;

- **Le contrôle d'accès** : Il s'agit d'authentifier les utilisateurs de façon à limiter l'accès aux données, serveurs et ressources aux seules personnes autorisés.

Comme toute science, la cryptographie possède sa propre terminologie. Etant donnée la jeunesse de cette science, et le fait qu'un nombre plus important de publications dans ce domaine sont en langue anglaise, le problème de la terminologie française se pose, parfois par manque de traduction.

- **Le chiffrement** est le procédé grâce auquel on peut rendre la compréhension d'un document (message, texte, son, image) impossible à toute personne qui n'a pas la clé de déchiffrement ;
- **Une clef** est un paramètre utilisé en entrée d'une opération cryptographique ;
- **Déchiffrer** consiste à retrouver le texte original d'un message chiffré dont on possède la clé de déchiffrement ;
- **Décrypter** consiste à retrouver le texte original à partir d'un message chiffré sans posséder la clé de déchiffrement ;
- **Texte en clair** (Clear text ou Plain text) : c'est un ensemble de caractères ou de bits sous forme lisible par un humain ou une machine ;
- **Texte chiffré** (Cipher text) : Résultat de la manipulation de caractères ou de bits via des substitutions ou des transpositions (permutations), ou les deux ;
- **Cryptologie** c'est la partie mathématique de la cryptographie et cryptanalyse.

Traditionnellement les termes *crypter* et *décrypter* un texte ou une image sont utilisés mais certains travaux semblent vouloir les remplacer par **coder** et *décoder* dans la mesure où *décrypter* peut vouloir dire *profaner*.

1.3 Caractéristiques des méthodes de cryptage

Pour caractériser les différentes méthodes de cryptage, certains acteurs aux rôles spécifiques sont souvent cités dans les publications (Dumont, 2009 ; Dumas et al., 2006, 2007 ; Bresson, 2015) :

- Alice : l'émettrice du message ;
- Bob : le récepteur du message ;
- Eve : une espionne qui suit la conversation ;
- Martin : un espion qui suit la conversation et peut la modifier ;
- Norbert : un tiers de confiance.

Ainsi pour définir les différents protocoles d'échange d'informations entre les deux premiers protagonistes (Alice et Bob) tout en évitant les oreilles indiscretes d'Eve et les attaques malveillantes de Martin, il existe deux grandes familles d'algorithmes de cryptage à base de clef : les algorithmes à clef privée (secrète) ou algorithmes symétriques, et les algorithmes à clé publique et privée ou algorithmes asymétriques.

1.3.1 Algorithme à clef secrète

Alice et Bob possèdent une clef commune qui servira de clef de cryptage et de décryptage (figure 2).



Figure 2 – Procédure de base pour le cryptage à clef secrète. Figure tirée de Dumont (2009)

Le procédé de chiffrement est dit symétrique.

Un algorithme symétrique est composé de trois sous algorithmes :

1. Un Algorithme de génération des clés $key_{Gen}(l) = K$: À partir d'un paramètre de sécurité il produit une clé aléatoire de l bits ;
2. Un algorithme de chiffrement $E_K(m) = C$: Qui produit le message chiffré C par une clé K du message de départ m ;
3. Un algorithme de déchiffrement $D_k(C) = m$: Qui utilise la clef K pour retrouver le message m à partir de C .

Les algorithmes de cryptage symétrique sont de deux types :

- Les algorithmes de chiffrement en continu ou en flots. Les données sont traitées en flux c'est-à-dire qu'on arrive à traiter les données de longueur quelconque sans besoin de les découper. Dans la littérature on trouve les algorithmes suivants :
 - Vernam (One Time Pad)
 - Pseudo-Vernam

- RC4
- Vigenère.

Les codes de *Vernam* ou codes à *masques jetables* reposent sur la fabrication de clé au hasard aussi longue que le texte à coder et qui ne peut servir qu'une fois. La mise en œuvre de ce code est lourde et délicate. Le stockage et la transmission de cette clé pose également des problèmes de sécurité.

Le principe de ces codes est basé sur la transformation du texte en une suite de chiffres en base b . On fabrique ensuite une suite aléatoire de chiffres de même longueur et l'on ajoute les deux suites ainsi obtenues sans retenue, c'est-à-dire qu'on effectue le calcul chiffre à chiffre modulo b .

Pour la génération des clés on utilise généralement des *registres à décalage* qui sont des *suites récurrentes linéaires sur des corps finis* ou encore LFBR (linear feedback register), mais aussi les *fonctions de hachage* dont nous parlerons dans le chapitre deux.

- *Les algorithmes de chiffrement par blocs*. Ils agissent sur le texte en clair en le découpant en blocs de bits. Quelques exemples sont : DES (Data Encryption Standard), Rijndael 2000, Serpent etc...

Lors de l'implémentation pratique, l'algorithme pur est combiné à un certains nombres d'opérations simples dans le but d'améliorer la sécurité sans porter atteinte à l'efficacité de l'algorithme. Cette composition est appelée mode cryptographique. L'utilisation de ces modes repose sur différents critères :

Sur le plan sécuritaire :

- Effacement des données remarquables (titre, introduction de texte, entête) ;
- Protection contre la modification du texte chiffré ;
- Chiffrement de plusieurs messages avec la même clé.

En termes d'efficacité :

- L'utilisation du mode cryptographique ne doit pas pénaliser l'efficacité du cryptosystème ;
- Limitation de la propagation des erreurs qui apparaissent dans le texte en clair ou dans le texte chiffré.

Les algorithmes de chiffrement par blocs peuvent être subdivisés en deux principaux modes : le mode ECB (Electronic Codebook) et le mode CBC (Cipher Block Chaining).

De nouvelles recherches ont introduit de nouveaux modes qui sont plus ou moins des variantes des deux premiers modes : CFB (Cipher Feedback), OFB (Output Feedback), CTR (Counter).

1.3.1.1 Le mode ECB

Le mode ECB ou encore *carte de codage électronique* dont le principe consiste à chiffrer de manière indépendante chaque bloc du texte en clair pour ainsi obtenir des blocs de texte chiffré. Le message M est découpé en bloc m_i de taille fixe. Chaque bloc est alors crypté séparément par une fonction E_k . La figure 3 présente un petit synoptique de chiffrement. Ce mode de chiffrement possède un certain avantage car il permet de chiffrer en parallèle des blocs d'un message.

Ce mode possède également quelques inconvénients : Le chiffrement d'un bloc de texte donnera toujours le même bloc de texte chiffré, ce qui cause certains problèmes du point de vue de la cryptanalyse car les données chiffrées pourront avoir des textures similaires facilement repérables. D'autre part, la prolifération des erreurs est assez importante car la modification d'un bit du texte chiffré lors d'un échange, entraînera la modification du texte en clair.

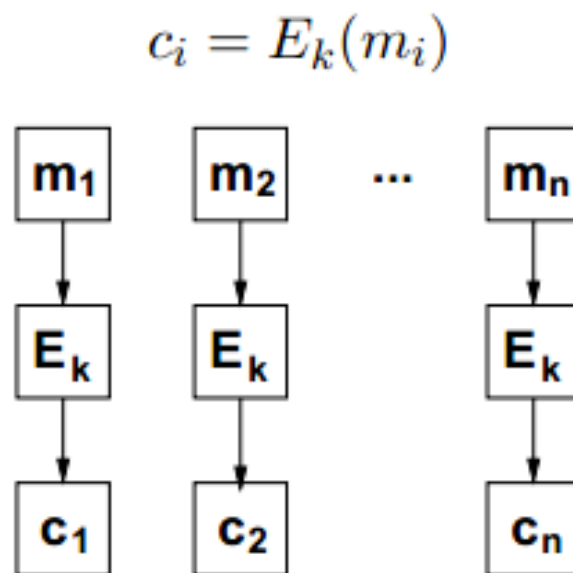


Figure 3 – Échantillon de chiffrement ECB. Figure tirée de Dumas et al. (2007)

1.3.1.2 Le mode CBC

Le message m_i est découpé en blocs C_i , la modification du bloc C_i dépend du bloc précédent C_{i-1} , ainsi les blocs sont liés entre eux ($C_i = f(C_{i-1})$). On parle alors de chaînage. Chaque bloc de texte en clair est combiné en ou exclusif avec le bloc chiffré précédent avant de subir encore un chiffrement conformément à la figure 4. Le premier bloc de texte en clair est d'abord combiné avec un bloc initial appelé vecteur d'initialisation, dans la littérature il est représenté par le symbole (IV). Ce symbole peut être un mot de passe ou un marqueur temporel (Timestamp). C'est un bloc de données aléatoires qui permet de commencer le chiffrement du premier bloc et qui fournit ainsi une forme de hasard indépendant du document à chiffrer. Ce vecteur d'initialisation peut être aussi utilisé dans des procédures d'identification et d'authentification.

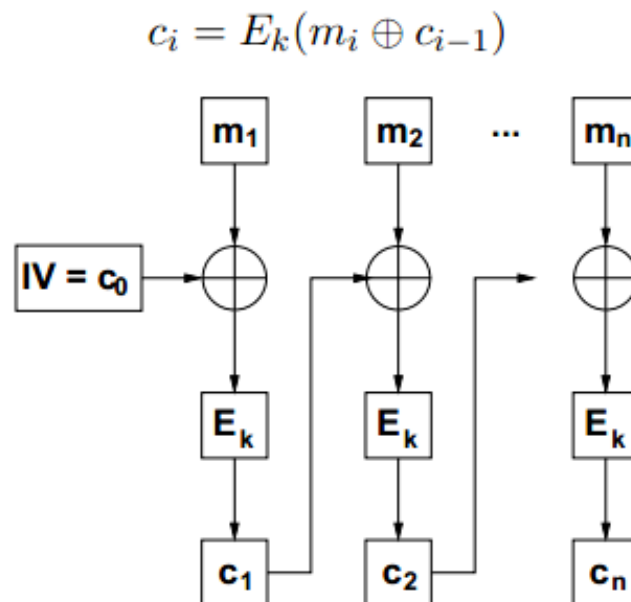


Figure 4 – Échantillon de chiffrement CBC. Figure tirée de Dumas et al. (2007)

Ce mode présente quelques avantages par rapport au mode ECB. La structure du texte en clair est masquée par le chaînage, en termes de sécurité, une nouvelle donnée est associée au mode de cryptage : le vecteur d'initialisation (IV). Il n'y a plus de risque de répétition des blocs chiffrés, les clés sont réutilisables. En ce qui concerne la propagation des erreurs, une erreur dans m_i affecte tous les C_i suivants mais ne se retrouve que dans le m_i , une erreur dans C_i affecte un bloc entier de m_i et le bit correspondant dans m_{i+1} . On parle parfois de mode CBC auto-réparateur.

Chiffrement par blocs avec itérations

Il s'agit d'algorithme de chiffrement par blocs dont la procédure comporte plusieurs rondes n_r au cours desquelles on peut utiliser des sous clés. Son schéma de principe est illustré à la figure 5. Ces sous-clés K_i dérivent toutes de la clef K . La fonction de ronde (tour) est destinée à être optimisée. On peut effectuer des substitutions et des permutations. La figure 6 présente un exemple de transition entre deux tours de cryptage.

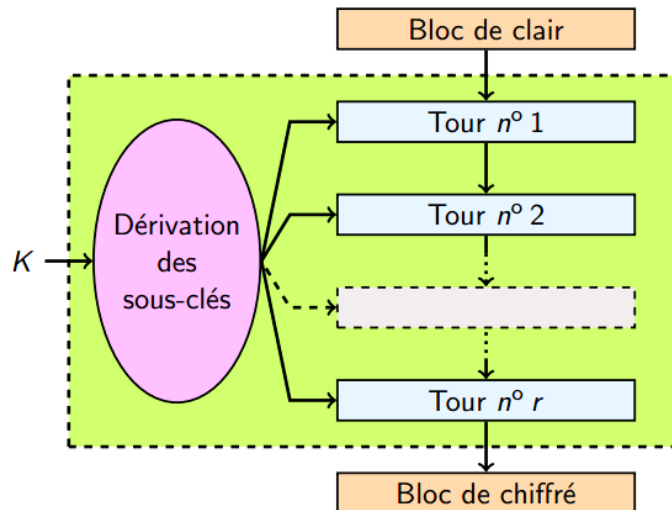


Figure 5 – Procédure de chiffrement par blocs. Figure tirée de Bresson (2015)

Comme algorithme de chiffrement, le schéma de Feistel est le candidat idéal. Dans ce type de chiffrement, un bloc de texte en clair est divisé en deux parties égales et la transformation de ronde est appliquée à un des deux sous blocs puis le résultat est ajouté à l'autre sous bloc par ou exclusif. Les deux sous blocs sont alors inversés pour le tour suivant. La figure 7 en présente le schéma synoptique.

La fonction f est appelée fonction de confusion. La complexité du schéma croit de façon exponentielle avec le nombre de tours.

De nombreux algorithmes de chiffrement utilisent des variantes du schéma de Feistel notamment, DES (Data Encryption Standard) créée par IBM en 1977, qui utilise une clé de 56 bits et des blocs de 64 bits, 16 tours avec des sous-clés de 48 bits. La répétition de cet algorithme mélange les bits du texte en clair en respectant les principes de Shannon (1949) : confusion et diffusion. Conçus et standardisés dans les années 70 et optimisés selon des besoins, grâce aux travaux de recherche on a pu mettre en place la DES triple à deux ou trois clés. IDEA (International Data Encryption Algorithm) qui utilise une clef de 128 bits à 8 rondes et RC5 (Rivest's Code 5) sont tous deux, des algorithmes qui utilisent également

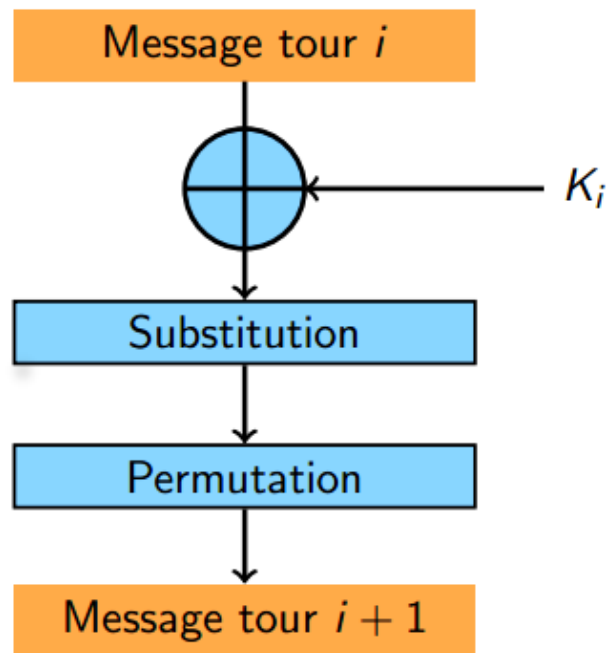


Figure 6 – Décomposition d'un tour. Figure tirée de Bresson (2015)

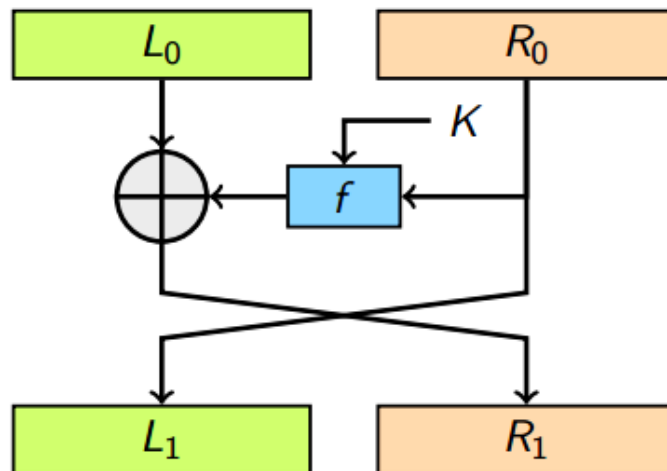


Figure 7 – Schéma synoptique de Feistel. $L_1 = R_0, R_1 = L_0 \oplus f(R_0)$ Figure tirée de Bresson (2015)

le schéma de Feistel. Les systèmes les plus récents qui utilisent le schéma de Feistel font partie d'un groupement d'algorithmes symétriques qui a vu le jour, lors d'un appel d'offre international destiné à remplacer les plus anciens réputés vulnérables. Ce groupement porte le nom AES (Advanced Encryption Standard). Nous pouvons citer : CAST256, CRYPTON, RIJNDAEL, TWOFISH etc. Ces algorithmes ont tous été développés de 1997 à 2000. Le candidat retenu était RIJNDAEL, conçu par Daemen et Rijmen en 1997, il est capable

d'utiliser une clef allant de 128 à 256 bits avec des substitutions/permutations de 10 à 14 tours. L'AES a été conçu pour résister au mieux à la cryptanalyse linéaire et différentielle, la meilleure attaque reste à l'heure actuelle la recherche exhaustive de la clef (Bresson, 2015).

1.3.1.3 Mode CFB

Le texte en clair est ajouté à la sortie du bloc chiffré. Le résultat est stocké dans un registre qui servira de feedback pour l'étape suivante. Le registre peut utiliser un certain nombre de bits en général qui se limite à 64 bits.

- Sécurité
 - Les fonctions de chiffrement et déchiffrement sont similaires,
 - Aucune répétition au niveau des blocs si le vecteur d'initialisation est différent,
 - Effacement du format standard.
- Efficacité
 - Vitesse de chiffrement identique à celui du mode ECB,
 - Pas de parallélisme.
- Propagation des erreurs
 - Lors de la perte d'un bloc C_i le synchronisme (c'est-à-dire que C_i ne dépend pas de C_{i-1}) est obtenu dès que le C_i est sorti du registre.
 - Une erreur dans m_i n'affecte pas tous les C_i suivants mais uniquement le m_i correspondant lors du déchiffrement.

1.3.1.4 Mode OFB

Le feedback est indépendant du texte, en fait toute la procédure est indépendante des blocs m_i et C_i . Il est plus utilisé dans le chiffrement de flux dans des canaux bruités.

- Sécurité
 - Vecteur d'initialisation unique et aléatoire,
 - Effacement du format standard,
 - Clef réutilisable,
 - Pas de répétition de blocs,
 - Une seule opération de ou exclusif (XOR) dans le cryptage est suffisante.
- Efficacité

- Vitesse de chiffrement identique à celui du mode ECB,
- Pas de parallélisme.
- Propagation des erreurs
 - Une erreur dans C_i affecte uniquement le bit correspondant de C_i .
 - Pas de mécanisme de récupération du synchronisme.

1.3.1.5 Mode CTR

Plus utile dans les réseaux à grande vitesse il est très rapide. On remplace le vecteur d'initialisation par un compteur qui est une fonction simple et on utilise une clef nouvelle pour chaque bloc de texte clair.

- Sécurité
 - Non utilisation du compteur.
- Efficacité
 - Parallélisme,
 - Accès aléatoire possible,
 - Simplicité du mode.
- Propagation des erreurs
 - Une erreur d'un bit dans C_i affecte uniquement le bit correspondant dans m_i .

1.3.2 Algorithme à clef publique

Les algorithmes à clef publique sont fondés sur l'existence de fonctions à sens unique, c'est-à-dire, une application simple au texte à chiffrer mais extrêmement difficile à inverser. Une autre particularité réside sur la clef. On utilise une clé pour chiffrer et une clef différente pour déchiffrer comme décrit dans la procédure de la figure 8.



Figure 8 – Procédure de base du cryptage à clef publique. Figure tirée de Dumont (2009)

- Une clef publique P_k (horizontale) connue de tous ;
- Une clef secrète S_k (verticale) connue uniquement des deux protagonistes de la conversation.

De temps à autre, ces deux clés peuvent être liées entre elles ou pas. L'inconvénient de la cryptographie à clé privée réside sur le procédé d'échange des clés entre les interlocuteurs et le nombre de clés dépend du nombre de communications établies. C'est pour pallier à ce problème que *Withfiel Diffi et Martin Hellman* ont introduit ce concept en 1976 mais de nombreux travaux allant dans le même sens ont été réalisés quelques années plus tard (Kwok et Tang, 2005). Avec un algorithme à clé publique pas besoin de plusieurs clés, une seule clé rendue publique par un acteur de l'échange peut servir pour toutes les autres communications. Ces algorithmes présentent également un inconvénient, ils sont assez lents en terme de fonctionnement vu qu'ils sont le plus souvent basés sur des structures mathématiques difficiles à résoudre. Une solution à ce problème est l'utilisation des algorithmes hybrides, qui est une association des deux algorithmes, un algorithme à clé privée dont l'échange des clés se fait par un algorithme à clé publique. Comme exemple d'algorithme à clé publique on peut citer : RSA (Rivest Shamir et Adelman), le système El Gamal, les codes basés sur les courbes elliptiques, et NTRU qui est basé sur la recherche du plus court vecteur dans un réseau adapté pour les systèmes quantiques.

1.3.2.1 Générateurs de données aléatoires et pseudo-aléatoires

Chercher à créer des mécanismes permettant de générer des séquences de nombres qui semblent avoir un comportement imprévisible et hasardeux mais néanmoins prévisible a abouti à la création des générateurs pseudo-aléatoires. Les générateurs pseudo-aléatoires sont d'une très grande importance dans la cryptographie. On les utilise dans la création des clés publiques et privées. Cette gestion de clés doit être totalement hypothétique et totalement aléatoire aux yeux des attaques extérieures afin qu'on ne puisse pas obtenir des mêmes séquences de données. Il n'existe à l'heure actuelle aucun standard concernant la génération des nombres aléatoires, ainsi de nombreux procédés de génération employés aujourd'hui sont inadaptés (Ferrenberg et al., 2001). Dans les cryptosystèmes actuels on utilise de moins en moins des générateurs aléatoires qui sont en général des boites noires et dont la connaissance du principe général de fonctionnement du générateur peut suffire pour identifier les paramètres de celui-ci à partir d'un échantillon des nombres qu'il produit. Les générateurs aléatoires produits par les ordinateurs sont périodiques et prévisibles sur la durée, comme substitut on utilise plus des générateurs pseudo-aléatoires car ils ont la

particularité d'être déterministes tout en respectant certaines propriétés statistiques et certains critères de qualité attribués aux nombres aléatoires. Pour la mise en œuvre pratique des simulations on distingue deux grandes familles de générateurs pseudo-aléatoires : les générateurs engendrés par des procédés algorithmiques et les générateurs physiques.

1.3.2.2 Les générateurs obtenus par procédés algorithmiques

Généralement chaque procédé algorithmique de génération de nombres pseudo-aléatoires est défini par un triplet (S, f, g) :

- S est un espace d'états (entiers, vecteurs, ensemble fini de données) ;
- f un endomorphisme, $f = S \rightarrow S$;
- g est une fonction de S dans $[0, 1]$.

A partir d'une valeur initiale $S_0 \in S$ appelée germe ou graine, le générateur fonctionne en fonction des itérations k , $S_k = f(s_{k-1})$, $k = 1, 2, 3, \dots M$.

Pour obtenir un bon générateur :

- les $(S_i, S_{i+1}, \dots, S_{i+M-1})$ doivent être répartis de manière uniforme dans l'intervalle $[0,1]$ ou dans une représentation multidimensionnelle d'ordre k avec $k = 1, 2, 3, \dots M$. Avec M très grand ;
- Le générateur doit posséder une période très élevée ;
- Procéder à l'analyse de la répartition des nombres produits par le générateur en particulier pour évaluer les corrélations qui peuvent exister entre les valeurs successives des différentes données.

Quelques exemples de ces générateurs sont : les générateurs linéaires congruentiels (L'Ecuyer, 2001), les générateurs multi récursifs (L'Ecuyer, 2004), les algorithmes à récurrences matricielles binaires (Matsumoto et Nishimura, 1998).

1.3.2.3 Les générateurs physiques

En observant les phénomènes physiques tels que : l'écoulement d'un fluide, le lancé de pièce double face, le tirage de boules dans une urne, les dispositifs expérimentaux de type physique, chimique ou biologique, l'observation d'un système optique, l'étude des composants électroniques etc. On peut générer de manière naturelle des données qui peuvent sembler être aléatoires en récoltant des données en temps réel issues de ces différents phénomènes par l'utilisation des capteurs et d'autres systèmes d'acquisition de données. Ils

peuvent présenter certains désavantages par rapport aux procédés algorithmiques car leur fonctionnement en temps réel est difficilement accommodable avec la notion de générer rapidement un très grand nombre de séquences pseudo-aléatoires à cause de la fiabilité de certains dispositifs expérimentaux, la présence des composants électroniques engendre la gestion du bruit dans la lecture des données, et la notion d'assonance dans la mesure où nous devons pouvoir reproduire les séquences de données utilisées. Mais pour palier à certaines de ces difficultés une autre science semble pouvoir se greffer à la cryptographie et à la génération des données il s'agit du chaos.

1.3.3 Chaos et Cryptographie

Pour une définition suffisante mais non exhaustive, on peut dire que le *Chaos* s'identifie à un type d'étude du comportement dynamique complexe pour découvrir la régularité statistique cachée dans des systèmes qui possèdent des caractéristiques particulières. Comme caractéristique on peut citer, l'extrême sensibilité à de petites variations dans leur évolution, tels que des conditions initiales ou la modification infinitésimale d'un paramètre du système à étudier. En conséquence, ces petites incertitudes dans la mesure où elles sont amplifiées au fil du temps, ce qui rend les systèmes chaotiques prévisibles en principe, mais imprévisible dans la pratique.

On retrouve ce genre de phénomène dans de nombreux systèmes physiques mais également dans beaucoup d'autre domaines comme les mathématiques, l'économie et la finance, la médecine et bien d'autres. Comme phénomènes physiques on a quelques exemples : les conditions météorologiques, la mécanique des fluides, la mécanique quantique, les systèmes électroniques, électrotechniques, optiques, optoélectroniques. En mathématique on peut citer : les équations différentielles, les équations récursives, les fonctions non linéaires etc...

Pour explorer les régimes chaotiques, on dénombre un certain nombre d'outils mathématiques comme les exposants de Lyapunov, le diagramme de bifurcation, le portrait de phase, la densité spectrale, etc...

Depuis les années 90, de nombreux chercheurs ont remarqué qu'il existe une relation intéressante entre le chaos et la cryptographie car de nombreuses propriétés des systèmes chaotiques ont leurs homologues correspondants dans les cryptosystèmes traditionnels. Le tableau 1 contient une liste partielle de ces propriétés (Gonzalo et Shujun, 2006).

Tableau 1 – Analogie entre systèmes cryptographiques et chaotiques. Tiré de Gonzalo et Shujun (2006)

Propriété chaotique	Propriété cryptographique	Description
Ergodicité	Confusion	La sortie présente une même distribution pour chaque entrée
Sensibilité aux conditions initiales / paramètre de contrôle propriété de mélange	Diffusion	Un petit écart de l'entrée peut provoquer un changement important à la sortie
État final/ État initial	Texte clair/texte crypté	Pour un ensemble à une entrée on a une sortie
Dynamiques déterministes	Données pseudo aléatoires déterministes	Un processus déterministe peut engendrer un comportement aléatoire (pseudo-aléatoire)
La complexité de la structure	Algorithmes (et attaques) complexes	Un ensemble de processus simples qui crée une très grande complexité
Nombre d'itérations presque infini	Nombre d'itérations fini	

1.3.4 Cryptage par cartes chaotiques

Le chaos déterministe peut générer des comportements dynamiques d'apparences aléatoires. Il serait donc intéressant d'utiliser ces derniers comme porteuses d'informations en télécommunications. Le diagramme principal de la communication sécurisée par le chaos est illustré sur la figure 9. Le principe est de masquer une information par des signaux chaotiques et de l'envoyer vers le récepteur dans un canal public. La clé du système de transmission peut être constituée de quelques paramètres ou de l'ensemble des paramètres des deux générateurs chaotiques à l'émission et à la réception. L'information cryptée est récupérée au niveau du récepteur.

Avec F : Opération mathématique plus ou moins complexe et F^{-1} : Opération inverse de F .

Le chiffrement d'un message par le chaos s'effectue donc en transformant l'information initiale grâce à un signal chaotique. On envoie par la suite le message noyé dans le chaos à un récepteur qui doit connaître les caractéristiques du générateur de chaos utilisé à l'émetteur. Il ne reste alors plus au destinataire qu'à effectuer la fonction inverse pour retrouver l'information. Si la génération du chaos et le cryptage du message ne présentent pas de problèmes majeurs, on va voir par la suite que du fait de la nature même du chaos, le décryptage va

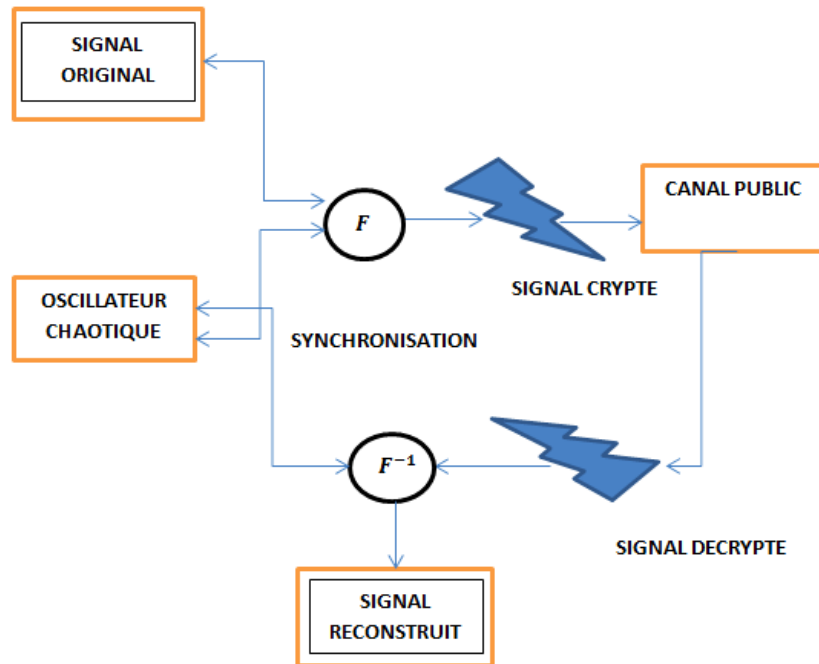


Figure 9 – Structure de communication sécurisée par chaos. Figure tirée de Pecora et Carroll (1990)

quant à lui présenter des étapes critiques notamment pour recréer la composante chaotique du message (synchronisation) et inverser la fonction. Il existe deux principales approches pour la conception de systèmes cryptographiques par chaos : analogique et digitale.

Approche analogique

Dans les techniques de communications traditionnelles utilisant le chaos basé sur l'approche analogique, l'information peut être transmise par un ou plusieurs signaux chaotiques et de plusieurs manières. On peut citer quelques-unes : le masquage chaotique (Memon, 2003 ; Effa et al., 2009), la modulation paramétrique (Oppenheim et al., 1993 ; Feki, 2003), la commutation chaotique (Dedieu et al., 1993 ; Parlitz et al., 1993), le cryptage par injection (Chen et al., 2005).

1.3.4.1 Masquage chaotique

Dans cette méthode appelée, cryptage par addition, l'émetteur est un système chaotique autonome dont le signal de sortie $y(t)$ est ajouté au signal du message $m(t)$. La somme de deux signaux est transmise au récepteur à travers le canal de transmission, qui est un canal public. Le récepteur est constitué d'un système chaotique identique à l'émetteur et un simple soustracteur. Ainsi, après la synchronisation des deux systèmes chaotiques (émetteur

et récepteur), le message est extrait à l'aide d'une opération de soustraction le principe est illustré sur la figure 10.

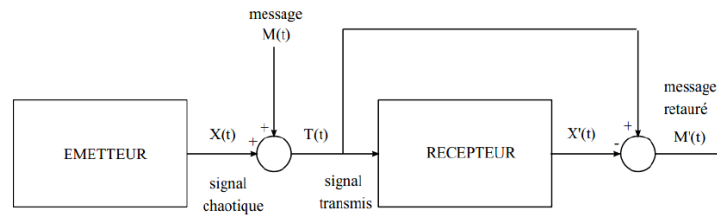


Figure 10 – Schéma synoptique de la technique de masquage chaotique. Figure tirée de Chen et al. (2005)

1.3.4.2 Cryptage par modulation paramétrique

Cette technique utilise le message contenant l'information pour moduler un paramètre de l'émetteur chaotique. Un contrôleur adaptatif est chargé de maintenir la synchronisation au niveau du récepteur, tout en suivant les changements du paramètre modulé. Le schéma correspondant est présenté à la figure. 11. Au niveau de l'émetteur, le fait de moduler un (ou plusieurs) paramètre(s) impose à la trajectoire de changer continûment d'attracteur, et de ce fait, le signal transmis est plus Complexe qu'un signal chaotique normal. Cependant, la façon d'injecter le message et donc la fonction de modulation des paramètres ne doivent pas supprimer le caractère chaotique du signal envoyé au récepteur. Il est important de souligner que cette technique exploite pleinement les qualités des systèmes chaotiques.

Elle n'a pas d'équivalent parmi les systèmes de communications classiques. Cependant, le cryptage par modulation s'est avéré sensible à certaines attaques.

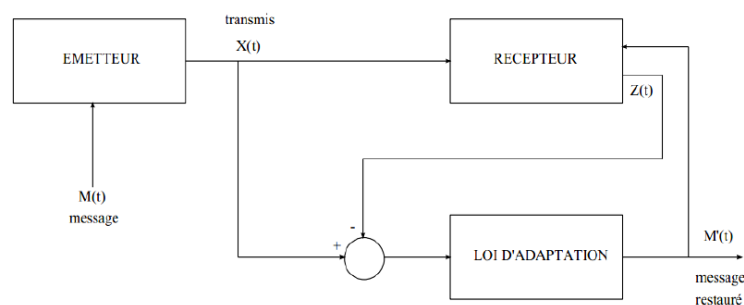


Figure 11 – Schéma synoptique de la technique de cryptage par modulation. Figure tirée de Yang (2004) ; Chen et al. (2005)

1.3.4.3 La commutation chaotique

Cette méthode (en anglais Chaos Shift Keying, CSK) est utilisée pour transmettre un message binaire (voir figure 12). L'émetteur est composé de deux systèmes chaotiques et pour chaque niveau de message $m(t)$ (0 ou 1), l'un des systèmes envoie sa sortie sur la ligne de transmission. Ainsi, le signal transmis commute entre deux attracteurs étranges. Le récepteur est constitué de deux systèmes chaotiques identiques à ceux de l'émetteur et un bloc de comparaison permet de relever la valeur du message noté $m'(t)$.

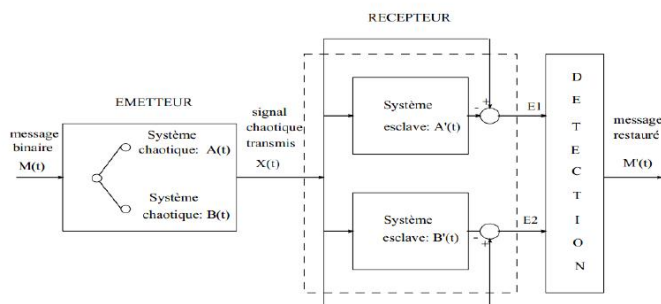


Figure 12 – Schéma synoptique de la technique de cryptage par commutation. Figure tirée de Hasler (1995) ; Dedieu et Kennedy (1995)

1.3.4.4 Cryptage par injection

Le schéma de principe de cette technique est représenté dans la figure 13. Il s'agit d'injecter le signal d'information dans la dynamique de l'émetteur chaotique. Le récepteur a pour but de synchroniser avec l'émetteur et de reconstruire le signal d'information. Le système esclave peut être conçu sous la forme d'un observateur à entrées inconnues ou un observateur à modes glissants (Chen et al., 2005). Cette technique est valable pour transmettre un message de nature binaire ou analogique, mais la puissance de ce dernier doit être suffisamment petite pour ne pas détériorer le comportement chaotique du système maître. Cette technique présente un niveau de sécurité nettement élevé par rapport aux techniques précédentes puisque le signal d'information est masqué dans la dynamique du système maître et que le signal chaotique disponible dans le canal public ne porte pas l'information d'une manière directe comme dans le cas de la technique de masquage chaotique.

Approche digitale

Les cryptosystèmes à base de chaos numérique (aussi appelés chiffrements chaotiques numériques), sont conçus pour les ordinateurs. Les cartes chaotiques sont générées de ma-

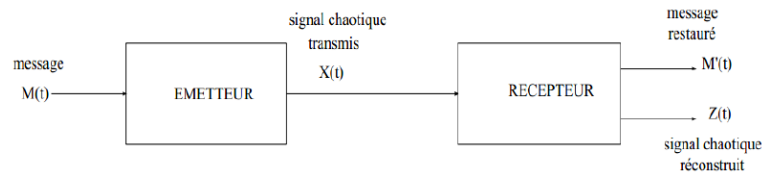


Figure 13 – Schéma synoptique de la technique de cryptage par injection. Figure tirée de Chen et al. (2005); Dedieu et Kennedy (1995)

nière discrète pour chiffrer le message en clair. On peut les classer en deux catégories : chiffrement chaotique par flot et chiffrement chaotique par bloc.

- Chiffrement par flot (Ying et al., 2004 ; Chen et al., 2005) : en anglais *stream cipher*, le message est crypté bit par bit, à l'aide d'un XOR appliqué à la sortie du générateur pseudo aléatoire chaotique. Les fonctions de cryptage sont généralement fonction du temps.
- Chiffrement par bloc (Masuda et Aihara, 2002 ; Xiao et al., 2005) : en anglais *block cipher*, c'est une fonction ou ensemble de fonctions qui construit un bloc de n bits simples d'un texte en clair en un bloc de n bits de texte chiffrée, n est appelé longueur du bloc. Il peut être considéré comme un chiffrement par substitution simple avec grande taille de caractère. La fonction de cryptage est paramétrée par une clé K de k bits, prenant des valeurs à partir d'un sous-ensemble K (l'espace de clés) afin de produire un vecteur V_k comportant k bits.

Par définition la fonction de cryptage par bloc de n bits est une application E :

$E : V_n \longrightarrow V_n / \forall K \in \mathbb{K}, E(P, \mathbb{K})$ difficilement inversible pour une application de E sur V_n , noté $E_{\mathbb{K}}(P)$. Le schéma inverse est la fonction de décryptage noté $D_{\mathbb{K}}(C)$. $C = E_{\mathbb{K}}(P)$ où C est le texte chiffré et P le texte en clair. Il est généralement admis que la clé est choisie au hasard.

Plusieurs algorithmes de chiffrement chaotique par bloc proposés dans la bibliographie utilisent des tables de substitution non linéaire ou S-Box. Elles adoptent pour la plupart d'entre elles la structure de Feistel (Kocarev, 2001 ; L'Ecuyer, 2001 ; Li et al., 2003 ; Benjeddou et al., 2008 ; Xiang, 2007).

Certains travaux estiment que le chiffrement par bloc présente certains avantages par rapport au chiffrement par flot, notamment pour la rapidité d'exécution, elle est plus appropriée pour certains types de communications (Barenghi, 2010 ; Yang, 2004) surtout si nous sommes limités en espace mémoire, mais aussi lorsque le canal utilisé pour la transmission

peut produire du bruit.

1.3.5 Diffusion et confusion

Shannon, dans ses travaux sur les fondements théoriques de la cryptographie (Shannon, 1949), a défini deux propriétés nécessaires à la conception d'un bon cryptosystème afin de se prémunir de certaines attaques statistiques. Il s'agit : de la diffusion et de la confusion. Diffusion signifie que si nous changeons un caractère du texte brut, alors plusieurs caractères du texte chiffré doivent changer, et de même, si l'on change un caractère du texte chiffré, plusieurs caractères du texte brut devraient changer. Cela signifie que la fréquence statistique des lettres dans le texte brut sont diffusées sur plusieurs caractères dans le texte chiffré, ce qui signifie qu'il faudrait plusieurs textes chiffrés afin d'effectuer une attaque statistique significative. Confusion signifie que la clé n'est pas liée au texte crypté d'une seule manière, en particulier chaque caractère du texte crypté devrait dépendre de plusieurs parties de la clé.

L'état de l'art sur les techniques de cryptage d'image numérique répartit ces deux notions en deux domaines : le domaine spatial et le domaine fréquentiel.

1.3.5.1 Domaine spatial

L'image numérique est l'image dont la surface est divisée en éléments de tailles fixes appelés cellules ou pixels, ayant chacun comme caractéristique un niveau de gris ou de couleur. Dans le domaine spatial les différentes procédures sont directement appliquées sur ces pixels. Les fonctions de traitement d'image dans le domaine spatial peuvent être exprimés sous forme de :

$$g(x, y) = T[f(x, y)] \quad (1.1)$$

$f(x, y)$ est l'image initiale à crypter, $g(x, y)$ est l'image de sortie et T représente une opération bijective sur f au voisinage de (x, y) .

Le domaine spatial est l'espace d'image normale, dans laquelle un changement de position dans l'image P projette directement un changement de position dans l'image intermédiaire S . La différence entre P et S (en pixels) correspond à une distance réelle. On peut également discuter de la fréquence avec laquelle les pixels de l'image changent. Dans la littérature on dénombre plusieurs techniques de transformations spatiales : la transformation horizontale/verticale, la transformation circulaire, la transformation en Zig Zag, la transformation par décalage de bits, la transformation hybride qui consiste en général au mixage des dernières transformations. Un cas très répandu est présenté sur la figure 14.

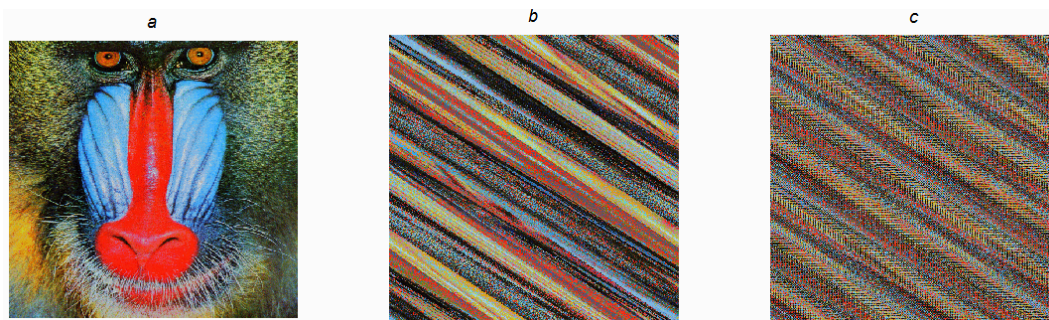


Figure 14 – Permutation des pixels d'une image utilisant la technique dite du Chat d'Arnold (en anglais *Arnold cat map*). (a) Image originale, (b) Image P après une itération, (c) Image P après 200 itérations

C'est un modèle de système dynamique qu'on applique sur une image, cette image est déformée lorsqu'on l'étire suivant un modèle mathématique modulaire. En partant d'un pixel d'une image P (de taille $n \times n$) de coordonnées (x, y) on obtient une image brouillée S dont les pixels ont pour coordonnées (x', y') par la transformation T :

$$P = (x, y) | x, y = 0, 1, 2, \dots, n - 1$$

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = T \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} x + y \\ x + 2y \end{pmatrix} \begin{pmatrix} 1 & p \\ q & pq + 1 \end{pmatrix} \text{mod}(n) \quad (1.2)$$

où p et q sont des entiers positifs.

Cette technique porte ce nom, car son auteur *Vladimir Arnold* a utilisé un chat pour illustrer ce mode de permutation. Si on répète de nombreuses fois la transformation, bien que l'on puisse exactement calculer l'image finale, il est tout à fait impossible de deviner l'image de départ, ni de savoir à l'avance comment sera la millièmes transformée, les seuls pronostics que l'on peut faire sont de l'ordre des statistiques. Tout semble indiquer que l'image est peinte au hasard.

Une autre technique qui s'impose dans les transformations des pixels d'une image produisant des textures remarquables se nomme la technique du boulanger (en anglais *baker's map*, figure 15). Cette transformation comporte trois phases :

- Transformation par une affinité rendant l'image deux fois plus longue dans le sens des abscisses et deux fois plus étroite dans le sens des ordonnées ;
- Découpage vertical en deux parties égales ;
- Superposition des deux moitiés pour obtenir l'image transformée de taille identique

à celle de l'image de départ.

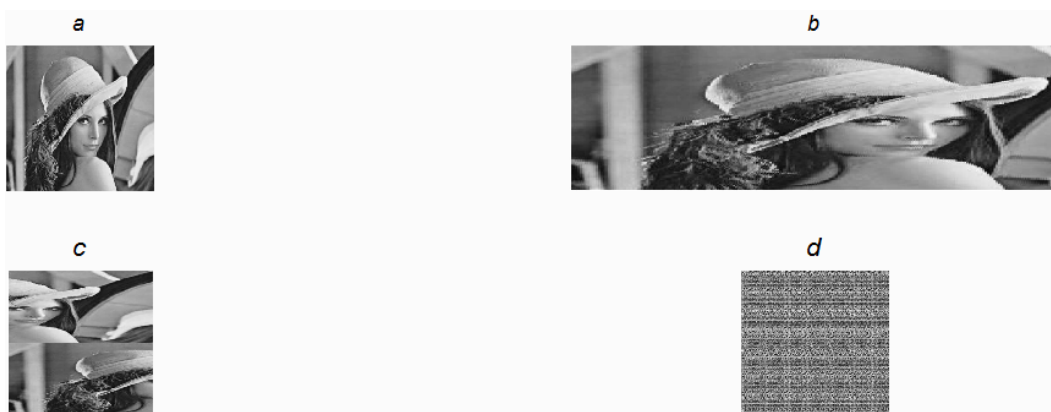


Figure 15 – Schéma synoptique de la technique de cryptage par injection. Figure tirée de Chen et al. (2005) ; Dedieu et Kennedy (1995)

L'expression mathématique de cette transformation T pour obtenir une image intermédiaire S , à partir d'une image P dont les pixels ont pour coordonnées (x, y) est représentée par l'équation (1.3) :

$$T : [0, 1]^2 \longrightarrow [0, 2] \times \left[0, \frac{1}{2}\right] \begin{cases} S(x, y) = P\left(x, \frac{y}{2}\right) & \text{si } 0 \leq x \leq \frac{1}{2} \\ S(x, y) = P\left(2x - 1, \frac{y+1}{2}\right) & \text{sinon} \end{cases} \quad (1.3)$$

La transformation utilisée qui est en fait une permutation périodique, et la période (nombre d'itérations de la transformation permettant de revenir à l'image initiale) est finie et est fonction des dimensions de l'image. Elle se calcule à partir de la décomposition en cycles de la permutation : la période est le PPCM (plus petit commun multiple) des longueurs des cycles.

1.3.5.2 Domaine fréquentiel

Le domaine fréquentiel est un espace dans lequel l'image sera considérée comme une somme de fréquences de différentes amplitudes. Toute image dans le domaine spatial peut être représentée dans un domaine de fréquence. Pour une image donnée les composantes de fréquence sont divisées en deux composantes principales :

- Les composantes hautes fréquences : elles correspondent aux bordures et aux contours d'une image ;

- Les composantes basses fréquences : elles correspondent à des régions régulières sur une image.

Dans le domaine fréquentiel, les fonctions de codage sont plus adéquates pour le cryptage, on développe également des masques de cryptage qui seront ensuite mêlés aux images à crypter par un processus de convolution (Vashisth et al., 2014). Une convolution permet de modifier le pixel courant par différentes opérations sur les valeurs des pixels du voisinage. La transformée de Fourier et la transformée de Fourier discrète sont d'ailleurs les fonctions les plus utilisées, une description de ces fonctions sera faite dans le chapitre deux.

Dans le domaine spatial, les images sont traitées directement par pixel ou par bit de pixel pour passer d'une texture à une autre, alors que dans le domaine des fréquences, on s'occupe plus de la vitesse à laquelle les valeurs de pixels changent dans le domaine spatial.

La revue de la littérature sur le cryptage d'images : notamment dans le domaine spatial, le domaine fréquentiel ainsi que les techniques hybrides est assez vaste. Un certain nombre de stratégies développées pour le chiffrement de l'image, basé sur le Chaos ont été récemment proposées, dans le but d'améliorer les multiples carences localisées dans les algorithmes plus anciens comme RSA, DES, AES. Nous nous intéresserons particulièrement aux procédures de chiffrement d'image centrées autour des cartes chaotiques, et nous présenterons aussi quelques travaux utilisant des suites à récurrence.

Deng et al. (2005), dans leur cryptage d'image ont utilisé un système neuronal chaotique et la carte d'Arnold. Dans cet article, les réseaux de neurones ont été utilisés pour la fabrication du chaos. Pour optimiser les cartes chaotiques il utilise un système retardé qui produit plusieurs exposants de Lyapunov positifs synonyme de grande complexité du système chaotique utilisé, mais l'utilisation d'une simple carte d'Arnold peut engendrer une faille de sécurité comme le révèle les auteurs de Zhang et Wang (2014) et en plus, aucune étude statistique n'a été proposée, donc le système ne fournit aucune garantie.

Xiao et Zhang (2006), utilisent deux systèmes chaotiques. L'un des systèmes chaotiques est utilisé pour produire une séquence chaotique transformé en un flux binaire par une fonction seuil. L'autre système chaotique a été utilisé pour construire une matrice de permutation. Tout d'abord, les valeurs de pixels de l'image originale ont été ajustées de façon aléatoire, et le flux binaire est utilisé comme une clé. Puis, l'image modifiée est de nouveau cryptée par la matrice de permutation.

Le chiffrement par flots a été utilisé dans divers travaux, dont certains que nous allons présenter dans ce paragraphe. Nien et al. (2009) ont utilisé une technique de chiffrement hybride pour l'image de couleur RGB, sur la base des systèmes chaotiques multiples. Ils

ont consolidé la méthode du pixel chaotique aléatoire (PCA) et le réarrangement de bit chaotique (BCR). Le PCA, qui est une méthode de chiffrement rapide qui peut faire varier les positions de chaque pixel, en utilisant des systèmes chaotiques comme *Henon*, *Lorenz*, *Chua et Rossler*. Ensuite, le BCR, qui utilise les cartes chaotiques pour effectuer la modification des valeurs des pixels. L'association du PCA et du BCR augmente l'espace des clés et éradique totalement les contours des images cryptées, brouille les caractéristiques de distribution des matrices d'une image de couleur (RGB), mais ce cryptosystème n'est pas fiable car il possède une corrélation assez faible et en plus les auteurs n'ont pas effectué d'autres tests afin de s'assurer de la qualité du cryptosystème.

Rhouma et al. (2009), ont proposé deux cartes chaotiques définies par morceaux pour construire leur cryptosystème. La première carte utilisée provient du système présenté à l'équation (1.4), on la nomme : PWLCM (*piece wise linear chaotic map*). La seconde carte utilisée est présentée par l'équation (1.5) elle est appelée carte *skew tent*. Les caractéristiques de la PWLCM sont appropriées pour la conception de systèmes de cryptage. Cette méthode transforme l'image de couleur en trois vecteurs puis cartographie l'ensemble des valeurs de chaque matrice à l'aide d'une carte chaotique appelée *Skew tent*. L'espace de phase de la carte *skew tent* est découpé en 256 sous-intervalles de largeur équivalente afin de gérer chaque pixel de l'image. Ce cryptosystème est robuste aux attaques différentielles et statistiques mais présente une faible entropie, l'entropie évalue le degré de désordre dans l'image cryptée.

$$x(n) = F[x(n-1)] = \begin{cases} x(n-1) \times \frac{1}{r} & \text{si } 0 \leq x(n-1) < r \\ [x(n-1) - r] \times \frac{1}{0,5-r} & \text{si } r \leq x(n-1) < 0,5 \\ F[1 - x(n-1)] & \text{si } 0,5 \leq x(n-1) < 1 \end{cases} \quad (1.4)$$

r est le paramètre de contrôle, il est contenu dans $[0;0,5]$ et la valeur initiale $x(0)$ est contenue dans $[0;1]$.

$$x(n) = F[x(n-1)] = \begin{cases} \frac{x(n-1)}{r} & \text{si } 0 \leq x(n-1) \leq r \\ \frac{1-x(n-1)}{1-p} & \text{si } 0 \leq x(n-1) \leq r \end{cases} \quad (1.5)$$

où $x(n) \in [0;1[$ et $r \in]0;1[$, r étant le paramètre de contrôle.

Eyebe et al. (2014) ont utilisé la carte chaotique PWLCM dans leur cryptosystème, mais ils ont également développé une nouvelle technique de diffusion et de confusion en utilisant

la résolution de l'équation diophantienne suivante :

$$au + bu = c \tag{1.6}$$

a, b et c sont des entiers naturels. La carte chaotique est utilisée pour générer les valeurs de a et b , puis à partir de la résolution de l'équation diophantienne on produit une matrice de permutation afin de brouiller les pixels de l'image à crypter. Malgré la taille réduite de l'espace des clés, le cryptosystème est robuste et assez rapide. Il résiste aux attaques statistiques et aux attaques différentielles, l'entropie de l'image cryptée est bonne, mais l'algorithme n'a pas de mode de chaînage, étant donné que le mode de cryptage utilisé est par bloc, le système est vulnérable à une attaque de sécurité de type cryptanalyse (Rhouma et Belghith, 2008 ; Li et al., 2011).

Sinha et Singh (2013) ont proposé une autre stratégie pour le chiffrement d'image des niveaux de gris, en utilisant une transformation en puzzle 3D. Dans un premier temps, l'image est transformée en vecteurs de bits, où chaque vecteur de bits est séparé en plusieurs blocs minuscules. Le puzzle 3D transforme chaque bloc de bits. Ils ont utilisé des transformées de Fourier fractionnaires pour crypter l'image. Cette méthode possède un bon MSE (mean square error), qui doit être supérieur à 0,05 pour une protection optimale des données (Sinha et Singh, 2013), par contre une seule métrique a été utilisée pour évaluer le cryptosystème, et en plus cette méthode nécessite au moins huit tours de permutations afin d'obtenir un meilleur brouillage ce qui dessert énormément la méthode proposée.

Abuturab (2012) a proposé un système de cryptage optique qui sécurise les images de couleurs en utilisant la lumière produite par un gyrateur. Pour leur stratégie, l'image de couleur est divisée en matrices R, G et B, ensuite, chaque matrice sera chiffrée indépendamment en appliquant d'abord le masque de phase aléatoire produit par la carte de premier ordre du chat d'Arnold et ensuite, celui produit par le gyrateur. Ces techniques améliorées sont utilisées en tant que touches supplémentaires de complexité dans le cryptage/décryptage d'une image, dans le but d'offrir une vigueur contre les agressions. Mais cet algorithme révèle quelques motifs de l'image originale, lors du test de sensibilité de la clef pendant le décryptage (Zhengjun et al., 2011).

Sui et Gao (2013) ont proposé un cryptage d'image de couleur, utilisant la transformée de Fourier fractionnaire itérative et une carte logistique couplée. Chaque matrice de couleur est permutée par une succession d'ensembles chaotiques qui est créé par la carte logistique à deux dimensions. L'image de permutation se dégrade en trois parties : Les deux premières parties sont encodées par la transformée de Fourier fractionnaire itérative, et la troisième partie est codée dans l'échelle de gris avec une distribution de bruit blanc stationnaire, qui

a la propriété de camouflage dans une certaine mesure. La permutation chaotique brouille l'image à la fois dans le domaine spatial et dans le domaine fréquentiel. En outre, le plan de cryptage amplifie l'espace des clés du cryptosystème. Les résultats de la simulation ont confirmé la faisabilité et l'efficacité de leur système mais avec quelques incertitudes.

1.3.6 Cryptanalyse

Un algorithme est vulnérable s'il est possible de le casser, c'est-à-dire quand il est possible de retrouver la clé en effectuant moins d'opérations qu'en utilisant la force brute. D'après les principes de Kerckhoffs (1883), la première hypothèse lors de l'analyse d'un cryptosystème réside sur la connaissance de la technique de cryptage par l'homme du milieu (celui qui veut décrypter l'information). Ce dernier doit pouvoir connaître la technique utilisée et les différents mécanismes de calcul. La seule information qui lui est caché est la clé secrète utilisée dans le cryptage. Selon ce principe on peut dénombrer 5 différentes attaques développées ces dernières années. Ces attaques sont citées ci-dessous et sont de difficultés variables :

3.5.8.1 Attaque sur texte chiffré seul (Ciphertext only attack)

Le cryptanalyste possède des exemplaires chiffrés des messages, il peut faire des hypothèses sur les messages originaux qu'il ne possède pas.

3.5.8.2 Attaque à texte clair connu (Known plaintext attack)

Le cryptanalyste possède des messages ou des parties de messages en clair ainsi que les versions chiffrées. La cryptanalyse linéaire fait partie de cette catégorie.

3.5.8.3 Attaque à texte clair choisi (Chosen plaintext attack)

Le cryptanalyste possède des messages en clair, il peut créer les versions chiffrées de ces messages avec l'algorithme que l'on peut dès lors considérer comme une boîte noire.

3.5.8.4 Attaque à texte chiffré choisi (Ciphertext attack)

Le cryptanalyste possède des messages chiffrés et demande la version en clair de certains de ces messages pour mener l'attaque.

3.5.8.5 Attaque adaptative à texte chiffré choisi (Adaptative Ciphertext attack)

C'est une attaque à texte chiffré choisie ou le choix du texte chiffré peut dépendre du texte clair reçu précédemment.

1.3.7 Autres types d'attaques

1.3.7.1 Attaque exhaustive ou brute

Le principe est ici de tester toutes les clés possibles de manière exhaustive. La limite maximale est donnée par T^N avec T représentant la taille de l'alphabet, et N la taille de la clé.

1.3.7.2 Attaque différentielle

Il s'agit de l'étude (modélisation) des transformations subies par le message durant son passage dans l'algorithme de chiffrement. Le principe est de modéliser ce qu'une modification en entrée induira sur le résultat de l'algorithme.

1.3.7.3 Attaque par dictionnaire

Lorsque la clé est un mot (un mot de passe, un symbole), on peut tenter de court-circuiter la Force Brute. Le principe est ici d'utiliser un recueil de mots possibles (le dictionnaire), et de tester tous les mots de ce dictionnaire.

Conclusion

Dans ce chapitre nous avons présenté de manière générale la cryptographie, la cryptanalyse ainsi que le cryptage par chaos. Nous avons présenté quelques caractéristiques majeures dans la construction d'un cryptosystème. Par ailleurs nous avons effectué une revue de la littérature sur les cryptosystèmes anciens et nouveaux. Dans le chapitre suivant il sera question de présenter les outils mathématiques ainsi que les mécanismes physiques utilisés pour bâtir et tester notre cryptosystème.

Chapitre 2

Étude méthodologique du cryptage chaotique : métriques et outils statistiques d'analyse

Introduction

Le but de ce chapitre est de présenter la méthodologie que nous avons utilisé dans la mise au point de notre cryptosystème. Nous commençons par rappeler les éléments fondamentaux nécessaires à la compréhension des différents outils utilisés.

2.1 Définition d'un système dynamique chaotique

Un système dynamique est une structure qui évolue au cours du temps tout en présentant quelques particularités :

- Causalité : Son avenir ne dépend que de phénomènes du passé ou du présent ;
- Déterministe : C'est-à-dire qu'à partir d'une condition initiale donnée à l'instant présent va correspondre à chaque instant ultérieur un et un seul état futur possible ;
- Contenir de la récurrence, c'est-à-dire qu'un mouvement partant d'un point repassera une infinité de fois aussi près que l'on veut du point initial. En particulier, il peut y avoir beaucoup de mouvements périodiques, qui au bout d'un certain temps reviennent exactement à leur point de départ.

Un système dynamique chaotique est un système qui en plus de ceux cités si dessus, dépend de plusieurs autres paramètres :

- La non-linéarité : un système linéaire ne pouvant pas être chaotique. En général, pour prévoir des phénomènes réels générés par ces systèmes, la démarche consiste à construire un modèle mathématique qui établit une relation entre un ensemble de causes et un ensemble d'effets. Si cette relation est une opération de proportionnalité, le phénomène est linéaire. Dans le cas d'un phénomène non linéaire, l'effet n'est pas proportionnel à la cause.
- Sensibilité aux conditions initiales : Les trajectoires issues de conditions initiales proches s'écartent exponentiellement dans le temps. En d'autres termes les phénomènes chaotiques sont très sensibles aux perturbations. L'un des premiers chercheurs à s'en être aperçu fut *Edward Lorenz* qui s'intéressait à la météorologie et par conséquent aux mouvements turbulents d'un fluide comme l'atmosphère. Lorenz venait de découvrir que dans des systèmes non linéaires, d'infimes différences dans les conditions initiales engendraient à la longue des trajectoires totalement différentes. Il a illustré ce fait par *l'effet papillon* (Lorenz, 1963).

2.1.1 Représentation mathématique d'un système dynamique

Un système dynamique est :

- Soit un système d'équations différentielles de la forme :

$$\dot{X} = F(X, t; \mu) \quad (2.1)$$

où $t \in I \subset \mathbb{R}_+$ est le temps, $X : t \in I \longrightarrow X(t) \in \mathbb{X} \subset \mathbb{R}^n$ sont les degrés de liberté du système, \mathbb{R}^n est l'espace des phases et $\mu \in \mathbb{P} \subset \mathbb{R}^p$ est un ensemble des paramètres et F une application définie telle que : $F : \mathbb{R}^n \times \mathbb{R}^p \longrightarrow \mathbb{R}^n$

- soit un système d'équation aux différences (système dynamique discret) :

$$X_{k+1} = G(X_k; \mu) \quad (2.2)$$

où $k \in \mathbb{N}$ représente le temps discrétisé, $X_k \in \mathbb{X} \subset \mathbb{R}^n$ sont les degrés de liberté du système. \mathbb{R}^n est l'espace des phases et $\mu \in \mathbb{P} \subset \mathbb{R}^p$ est un ensemble des paramètres et F une application.

Lorsque F ne dépend pas explicitement du temps : $F(X, t; \mu) = F(X; \mu)$, le système est dit autonome, il est non autonome si F dépend explicitement du temps.

Dans l'espace des phases, une solution de l'équation (2.1) est appelée une trajectoire ou une orbite.

On appelle alors flot du système dynamique la famille d'applications

$$\phi_t : X_0 \in \mathbb{X} \longrightarrow X(X_0, t), t > 0$$

Le flot d'un système dynamique est un point de vue global et géométrique sur les équations différentielles. Pour fixé, le flot est représentatif de la manière dont l'application fait évoluer un ensemble donné de points dans l'espace des phases. L'analogie avec un flot qui s'écoule est bien sûr à l'origine de la dénomination.

2.1.2 Techniques d'identification et de prédiction du chaos

Le chaos est tellement omniprésent dans la quasi-totalité des phénomènes qui gouvernent notre monde, ainsi le comportement des marchés financiers et des consommateurs, les phénomènes météorologiques, l'évolution d'un écosystème ou encore le mouvement de certains corps célestes fournissent des exemples de ces phénomènes imprévisibles qui ont un impact sur l'activité humaine. C'est ainsi que de nombreux chercheurs ont essayé de développer de nombreuses techniques de détection de chaos dans le but de les identifier et de les révéler.

2.1.2.1 Techniques d'identification du chaos

Dans la littérature on dénombre un certain nombre de techniques afin de caractériser un système chaotique :

- **Le portrait de phase** ou l'attracteur : d'un système dynamique est une représentation graphique de plusieurs trajectoires représentatives dans l'espace des phases ;
- **La section de Poincaré** : à partir d'une condition initiale, on obtient ainsi un ensemble de points formant la section ou coupe de Poincaré, c'est-à-dire une carte à deux dimensions du système dynamique étudié ;
- **Le diagramme de bifurcation** : une bifurcation peut correspondre à l'apparition ou la disparition de nouvelles singularités, un changement dans la stabilité du système ou dans la forme d'un attracteur chaotique ;
- **Les exposants de Lyapunov** : mesure le degré du chaos présent dans un système (signal, séquence de données). Dans la littérature on le note λ ;
- **La dimension de Lyapunov** : très utile pour effectuer une classification du comportement chaotique de plusieurs systèmes (Li et York, 1975 ; Parker et Chua, 1987).

2.1.2.2 Techniques de prédiction du chaos

L'exposant de Lyapunov est une méthode quantitative capable de prédire si le système étudié est réellement chaotique ou non, cette technique est moins utilisée lorsqu'on veut étudier des séquences de données dont on ignore la provenance, comme par exemple des données expérimentales (Gottwald et Melbourne, 2009). La prédictibilité d'un système reflète une grande partie de son comportement dynamique. La détection du chaos s'effectue avec une simple visualisation de la série temporelle. Un système périodique peut être prédictible sans effort après l'observation d'une période, un système linéaire est prédictible par une simple opération d'extrapolation d'un nombre fini de points et un système aléatoire a une prédictibilité aléatoire caractérisée par des moments statistiques calculables. Les systèmes chaotiques ont une prédictibilité spécifique qui dépend de l'intervalle temporel entre les points utilisés pour la prédiction et les points projetés dans le futur. De nouvelles recherches ont classé les méthodes de prédiction en deux catégories : les modèles de prédiction linéaires (Kostelich et Lathrop, 1992), et les modèles de prédiction non linéaires plus récentes et plus intéressantes (Gottwald et Melbourne, 2009 ; Bandt et Pompe, 2002 ; Eyebe et al., 2016). Ces derniers modèles sont très utiles dans le choix des cartes chaotiques utilisées pour le cryptage, nous en définirons quelques-uns dans la section suivante.

2.2 Méthodologie de cryptage d'images par chaos

Dans la littérature certaines notations mathématiques sont utilisées pour définir une image numérique dont la taille $L = H \times V$, où H représente le nombre de lignes et V le nombre de colonnes. $p(i, j)_{i=1, j=1}^{H, V}$ et $p(k)_{k=1}^L$ sont utilisées pour représenter une image sous forme matricielle (image 2D), ou une image représentée sous forme de vecteur (image 1D). Pour crypter une image numérique, on peut la décomposer mathématiquement en niveaux de gris (pixels), à l'aide de ses composantes primaires (RVB ; Kostelich et Lathrop, 1992), puisse que l'image de couleur est composée de rouge (R) de vert (V) et de bleu (B). La représentation YUV est aussi très utilisée dans le cryptage fréquentielle, principalement dans tout ce qui est compression d'image ou crypto-compression de l'image (El-Latif et Niu, 2013). Y représente la luminance de la couleur, et U et V , la chrominance de cette couleur dans le rouge et le bleu (El-Latif et Niu, 2013).

Il existe trois principales façons d'utiliser le chaos dans le cryptage d'image :

- Comme source pour générer des bits Pseudo-aléatoires avec des propriétés statistiques souhaitées pour réaliser une opération de permutation secrète (Mao et al.,

2004 ; Guan et al., 2005 ; Eyebe et al., 2014) ;

- Comme source pour générer des pixels pseudo-aléatoires avec des propriétés statistiques souhaitées pour réaliser une opération de substitution secrète (Chen et al., 2004 ; de Oliveira et Sobottka, 2008) ;
- Comme générateur à la fois en permutation et en substitution des pixels de l'image (Lian et al., 2005 ; Zhou et al., 2008 ; Pisarchik et Zanin, 2008).

Une approche générale de chiffrement à base de chaos se décompose en quatre étapes standards :

- Choisir une carte chaotique : il est nécessaire d'envisager des cartes avec de bonnes qualités et de bonnes propriétés (implémentation logicielle et matérielle facile) ;
- Présentation des paramètres, génération de la clef de cryptage ;
- Fonction de cryptage, discrétisation ;
- Analyse de sécurité et Cryptanalyse.

2.2.1 Choix de la carte chaotique

La carte chaotique choisie doit :

- Être simple à réaliser ;
- Avoir une grande dimension, tout en possédant un large ensemble de paramètres (paramètres de contrôle de chaos, paramètre de liaison par exemple) ;
- Déterministe, afin de rendre le décryptage possible.

Les oscillateurs chaotiques possèdent plusieurs bonnes propriétés (Matsumoto, 1997) :

- Réalisation simple ;
- Présence de non linéarité dans le système ce qui accroît la complexité dans le cryptosystème ;
- Présence de grand nombre de paramètres.

Après avoir choisi un système chaotique, un certain nombre de tests peuvent être effectués sur les séquences de données générées, afin de valider le caractère chaotique de la carte chaotique utilisée pour le cryptage : il s'agit du test binaire 0-1 (Gottwald et Melbourne, 2004), et les méthodes basées sur l'étude de l'entropie du système (Bandt et Pompe, 2002 ; Eyebe et al., 2016).

2.2.1.1 Test binaire 0-1

Proposée par Gottwald et Melbourne (2004), cette méthode permet de détecter le chaos dans une série temporelle. Les auteurs ont souligné que contrairement à l'exposant de Lyapunov la méthode peut être appliquée directement à une série de données et ne nécessite pas une connaissance préalable du système sous-jacent. Aussi, la méthode est indifférente à la forme du système testé et des équations différentielles à l'origine de la série, elle fonctionne aussi bien sur les systèmes dynamiques ou systèmes discrets. Ainsi, le test prend une série de données à l'entrée et délivre une sortie 0 ou 1 suivant l'existence d'une nature chaotique ou pas. Dans cette méthode on calcule les variables de translation $p(t)$ et $q(t)$ en fonction d'une série de données (échantillon du système) à observer $\phi(t)$ d'un système temporelle $x(t)$. Le paramètre est calculé par la formule suivante :

$$p(t) = \int_0^t (\phi(x(s)) \cos(\theta(s))) ds \quad (2.3)$$

$$q(t) = \int_0^t (\phi(x(s)) \sin(\theta(s))) ds \quad (2.4)$$

avec,

$$\theta(t) = cte + \int_0^t \theta(s) ds \quad (2.5)$$

Ainsi une fonction $D(t)$ du déplacement moyen carré de $b(t)$ est défini par :

$$D(t) = \lim_{t \rightarrow \infty} \frac{1}{T} \int_0^T \{ [p(t+\tau) - p(\tau)]^2 + [q(t+\tau) - q(\tau)]^2 \} d\tau \quad (2.6)$$

Et le paramètre qui permet d'établir la présence du chaos dans la série temporelle $x(t)$ est noté K , et obtenu par la formule suivante :

$$K = \lim_{t \rightarrow \infty} \frac{\log(D(t))}{\log(t)} \quad (2.7)$$

Le test est capable de faire la distinction entre la dynamique régulière et chaotique en utilisant uniquement les séries de données générées par les systèmes dynamiques. Il présente également l'avantage d'être binaire, il ne dépend pas de la nature des champs de vecteurs, ainsi que de la dimension du système, et il ne nécessite pas la reconstruction de l'espace de phase. Le test 0-1 a été appliqué avec succès à de nombreux systèmes dynamiques (Jing et al., ; Gottwald et Melbourne, 2009). Cependant, le test 0-1 par lui-même ne fait aucune différence entre les orbites quasi-périodiques et périodiques et il est également coûteux en temps de calcul.

2.2.1.2 Entropie de permutation

La permutation d'entropie (PE; Bandt et Pompe, 2002) a récemment été proposée comme une nouvelle mesure pour caractériser la complexité des séries temporelles non linéaires. Considérons une série temporelle de données $[x_t]_{t=1}^T$ et ses dérivées retardées $X_j^{m,\tau} = [x_j, x_{j+\tau}, x_{m-1}]$ avec $j = 1, 2, \dots, T - (m - 1)\tau$, où m et τ sont respectivement le nombre de variables contenu dans l'échantillon et l'écart entre deux symboles de la variable x . Pour calculer l'entropie de permutation de chacune des séries des données de taille $N = T - (m - 1)\tau$, on utilise l'entropie de *shannon* des $m!$ Différents symboles contenue dans chaque série de données. Cette entropie est obtenue par la formule suivante :

$$H_p(m) = - \sum_{j=1}^k P_j \ln P_j \quad (2.8)$$

Où $P_j = 1/m!$ correspond à la probabilité de distribution de chaque symbole d'une série de données.

L'entropie de permutation caractérise la diversité des modèles classiques. Cette méthode n'est ni sensible au bruit ni à la différence d'amplitude entre les points des séries temporelles. Elle se comporte également comme l'exposant de Lyapunov. La principale lacune dans cette méthode réside dans le fait qu'aucune information en dehors de la structure de commande n'est conservée lors de l'extraction des motifs simples de chaque série temporelle. Cette méthode a une certaine faiblesse, une incapacité à différencier les modèles distincts d'un même motif de données.

2.2.1.3 Méthode 3ST

La méthode 3ST est basée sur l'analyse de la structure des séries de données analogiques et discrètes. Cette technique prend en compte les propriétés du signal à étudier, c'est à dire si le signal est périodique ou quasi-périodique puis il détermine si la dynamique est régulière ou chaotique. Un signal périodique se caractérise par sa période, et la période elle est caractérisée par un modèle de base, de sorte qu'il peut être défini comme étant une répétition périodique du modèle de base. Un signal quasi-périodique présente un modèle de base qui tend à être répétée périodiquement après une longue durée d'observation. Cette méthode étudie la distribution des états du système dans une série de données en fonction du temps. Considérons une série temporelle de données $[x_t]_{t=1}^T$ et ses dérivées $X(k) = [x_1, x_2, \dots, x_m]$ avec $j = 1, 2, \dots, m, k \in \mathbb{N}$, où m représente la dimension et k le vecteur d'état. La série observée $x_j(k) = \phi(x(k))$. Pour définir les différents modèles, les auteurs définissent d'abord

une fonction g telle que $u_j = g(x_j)$ de telle sorte que u_j soit la série de données triées par ordre croissant :

$$x_j = x_j(1), x_j(2), \dots, x_j(L) \quad (2.9)$$

où M est la longueur de la série x_j avec :

$$\begin{aligned} x_j(2) < x_j(M) < \dots < x_j(L) < \dots < x_j(1) \\ u_j = g(x_j) &= [x_j(2), x_j(M), \dots, x_j(L), \dots, x_j(1)] \end{aligned} \quad (2.10)$$

q est une fonction telle que $v_j = q(x_j)$, de telle sorte que v_j soit la distribution des différents indices de sortie des séries u_j dans le vecteur initiale x_j :

$$v_j = q(x_j) = (2, M, L, \dots, 1) \quad (2.11)$$

Pour série observée $\phi(x)$ T -périodique comportant P périodes, la longueur M de la série des données $x_j = \phi(x)$ vaut : $M = P \times T$ et le nombre d'états distincts vaut T . Ainsi u_j et v_j sont obtenues par les équations suivantes :

$$u_j(k) = x_j(T \times i + b) \quad (2.12)$$

avec

$$\begin{cases} k = 0, 1, \dots, T \times P - 1 \\ i = k \bmod(P) \\ b = \text{floor}(k/P) \end{cases}$$

Où \bmod représente la fonction modulaire et la fonction floor représente une fonction arrondie par défaut.

$$v_j(k) = T \times i + b \quad (2.13)$$

Si la distribution des états est périodique, la distribution correspondante des indices u_j est une fonction linéaire par morceaux, où toutes ses fonctions linéaires présentent la même pente et cette pente est égale au cycle de la fonction observable.

Dans le cas des signaux quasi-périodiques, la distribution des indices $v_j(k)$ reste une fonction linéaire par morceaux. Les modèles de base ne sont pas régulièrement répétées et les pentes de $v_j(k)$ dépend de chaque état de $\phi(x)$, alors que le nombre d'états dépend du temps d'observation (la longueur de M de la série de données), d'où :

$$v_j(k, M) = T(b) \times i + b(M) \quad (2.14)$$

Cependant, même si le volume de distribution $v_j(k)$ ne correspond pas à un motif périodique de signaux, il présente une limite de telle sorte que le signal quasi périodique peut être caractérisé par l'une de ses pentes. Afin de tenir compte de la dépendance temporelle de $v_j(k, M)$ les auteurs considèrent comme motif caractéristique, la plus grande pente (GP) définie comme suit :

$$GP = \max_{1 \leq k \leq M-1} [v_j(k+1, M) - v_j(k, M)] \quad (2.15)$$

$$T = \lim_{M \rightarrow \infty} GP(M) \quad (2.16)$$

Pour effectuer la détection du chaos dans la série de données on calcul d'abord l'erreur quadratique moyenne :

$$\sigma_{GP}(M, n) = \sqrt{\frac{1}{N} \sum_{j=0}^P [GP(jM_0 + n) - \overline{GP}]^2} \quad (2.17)$$

avec,

$$\overline{GP} = \frac{1}{M} \sum_{j=0}^P GP(jM_0 + n) \quad (2.18)$$

où $jM_0 + n$ représente la longueur de la série de données, n représente la plus petite durée d'observation pour une grande pente, et $\sigma_{GP}(M, n)$ mesure la capacité du système dynamique à générer de nouveaux modèles pendant son évolution dans le temps. Après avoir calculé $\sigma_{GP}(M, n)$ on calcule les paramètres $\mu(M, n)$ et $K(n)$ par les formules suivantes :

$$\mu(M, n) = \frac{\log[1 + \sigma_{GP}(M, n)]}{\log(M)} \quad (2.19)$$

$$K(n) = \lim_{M \rightarrow \infty} \mu(M, n) \quad (2.20)$$

K représente l'évolution asymptotique de l'erreur quadratique moyenne des GP .

K doit être égale à zéro dans le cas des signaux périodiques ainsi que des signaux quasi-périodiques, mais doit être supérieure à zéro dans le cas de comportement chaotique. Ainsi, peut être utilisée comme dans le cas du test 0-1, en supposant que sa sortie soit binaire : $K = 0$ pour la dynamique régulière et $K > 0$ pour une dynamique chaotique. Il est donc possible d'observer la dynamique d'évolution de ce paramètre comme l'exposant de Lyapunov. Dans ce cas on utilise la formule suivante :

$$\lambda(n) = \lim_{M_1 \rightarrow \infty} \sum_{k=1}^{l-1} [\mu(M_{k+1}, n+1) - \mu(M_k, n)] \quad (2.21)$$

La figure suivante présente la dynamique du paramètre de détection de chaos lorsque ces différents tests sont effectués sur une carte logistique :

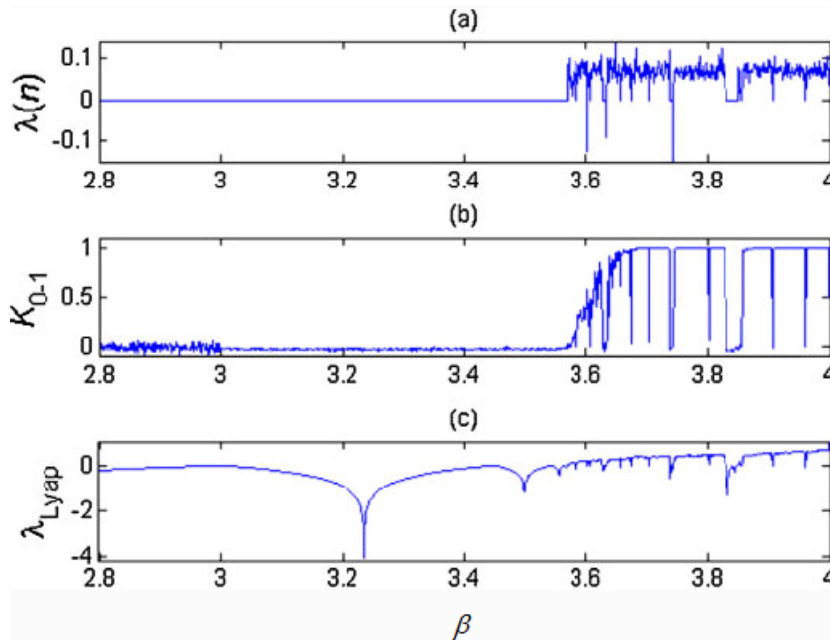


Figure 16 – Évolution de la dynamique du paramètre de détection du chaos sur la carte logistique. $x(k+1) = \beta x_k(1-x_k)$. (a) 3ST Test, (b) Test binaire 0-1, (c) Maximum de l'exposant de Lyapunov. Figure tirée de Eyebe et al. (2013)

Certains tests présentés ci-dessus ont subi quelques modifications et ont produit d'autres tests de chaos (Bandt et Pompe, 2002). Les conclusions de ses récents travaux montrent que, certains tests sont plus adaptés que d'autres, notamment dans l'étude des séries de données de type expérimentale.

2.2.2 Présentation des paramètres et génération de clef de cryptage

Dans la quasi-totalité des méthodes de cryptage d'images par chaos, les paramètres ainsi que les conditions initiales du système chaotique utilisé peuvent servir de clef de cryptage, il peut alors être nécessaire de pouvoir les générer. De nombreux processus de génération de clef sont présentés dans la littérature (Kunt, 1993 ; Pareek et al., 2003). Supposons que le cryptosystème utilise une clé de cryptage K de 256 bits, le système chaotique utilisé possède 6 conditions initiales $(x_0^1, y_0^1, z_0^1, x_0^2, y_0^2, z_0^2)$ et 3 paramètres (α, β, γ) à déterminer. La clé K est subdivisée en blocs d'octets K_i telle que $K = K_1, K_2, K_3, K_4, \dots, K_{32}$.

Les sous clés K_i sont générées aléatoirement et sont alors stockées sous forme d'octets (8bits). Chaque condition initiale est générée par les équations suivantes :

$$x_0^1 = [(K_1 \oplus K_2 \oplus K_3 \oplus K_4) + \sum_{i=1}^{i=32} K(i)]/2^8 \text{mod}(1) \quad (2.22)$$

$$y_0^1 = [(K_5 \oplus K_6 \oplus K_7 \oplus K_8) + \sum_{i=1}^{i=32} K(i)]/2^8 \text{mod}(1) \quad (2.23)$$

$$z_0^1 = [(K_9 \oplus K_{10} \oplus K_{11} \oplus K_{12}) + \sum_{i=1}^{i=32} K(i)]/2^8 \text{mod}(1) \quad (2.24)$$

$$x_0^2 = [(K_{13} \oplus K_{14} \oplus K_{15} \oplus K_{16}) + \sum_{i=1}^{i=32} K(i)]/2^8 \text{mod}(1) \quad (2.25)$$

$$y_0^2 = [(K_{17} \oplus K_{18} \oplus K_{19} \oplus K_{20}) + \sum_{i=1}^{i=32} K(i)]/2^8 \text{mod}(1) \quad (2.26)$$

$$z_0^2 = [(K_{21} \oplus K_{22} \oplus K_{23} \oplus K_{24}) + \sum_{i=1}^{i=32} K(i)]/2^8 \text{mod}(1) \quad (2.27)$$

$$\alpha = [(K_{25} \oplus K_{26} \oplus K_{27} \oplus K_{28}) + \sum_{i=1}^{i=32} K(i)]/2^8 \text{mod}(1) \quad (2.28)$$

$$\beta = [(K_{29} \oplus K_{30} \oplus K_{31} \oplus K_{32}) + \sum_{i=1}^{i=32} K(i)]/2^8 \text{mod}(1) \quad (2.29)$$

$$\gamma = (x_0^1 + y_0^1 + z_0^1 + x_0^2 + y_0^2 + z_0^2 + \alpha + \beta) \text{mod}(1) \quad (2.30)$$

L'opérateur \oplus représente le XOR (Ou exclusif) et la fonction *mod* représente le calcul modulaire, utile pour obtenir des valeurs dans l'intervalle $[0,1]$. Les équations (2.22) à (2.30) montrent que les conditions initiales et les paramètres du système chaotique utilisé sont très sensibles à la modification d'un seul bit de la clef secrète de 256 bits. Certains cryptosystèmes utilisent des fonctions non linéaires lors de la génération des clés de cryptage. Pour une clef de cryptage K de 128 bits, la clé K est subdivisé en sous clés de 8 bits.

$K = K_1, K_2, K_3, \dots, K_{16}$, le système chaotique utilisé contient 9 variables à déterminer, 7 conditions initiales et deux paramètres du système chaotique. Elles sont calculées par les équations suivantes :

$$T_i = \left[\frac{K_{2i-1} + (2i - 1)}{K_{2i} + 2i} \times \sum_{j=1}^{j=16} \frac{K_j \times 2^{8 \times (j-1)}}{2^{128}} \right] \text{mod}(1) \quad (2.31)$$

$$T_9 = \sum_{i=5}^{i=8} T(i) \text{mod}(1) \quad (2.32)$$

Avec $i = 1, 2, \dots, 8$.

Les auteurs proposent une analyse qui prouve qu'il est impossible de trouver des clés différentes qui ont le même effet sur les paramètres initiaux. Ainsi, la transformation algé-

brique proposée est très sensible à la clé secrète de sorte que même le changement d'un bit dans la clé secrète provoque des résultats complètement différents. Lors de la déclaration des sous clés, i représente l'index des sous clés de session, K_i représente la valeur ASCII ou la valeur binaire de la clef de session choisi au hasard. Ces clés de session sont générées soit par une bibliothèque standard du langage de programmation C++ (`rand()`), ou par une fonction standard de génération binaire décrite dans le logiciel Matlab (`randint()`).

2.2.3 Fonctions de cryptage et discrétisation

Le chaos n'est pas suffisant pour garantir la sécurité d'un cryptosystème. Conformément aux prescriptions de Shannon (1949), chaque algorithme de chiffrement doit posséder des propriétés de diffusion, et de confusion, de mélange et de sensibilités aux changements du texte en clair et de la clé secrète. Ainsi pour s'assurer de plus de sécurité on peut associer au chaos des fonctions de hachage (génération de clef), ou l'utilisation des fonctions de cryptage. Les techniques fondamentales pour crypter un bloc de symboles sont la confusion et la diffusion.

2.2.3.1 Diffusion de l'image

La diffusion peut propager un changement dans l'image originale sans aucun apport extérieur. La permutation qui change la séquence des symboles dans le bloc est la méthode la plus simple de diffusion. La diffusion peut s'effectuer sur les pixels (Congxu et al., 2015) ou sur les blocs de bits de pixels (Zhang et Wang, 2014). Pour une image en niveau de gris, dont la taille est donnée par $L = H \times V$. La séquence des pixels de longueur L , est donnée par l'équation (2.33) qui résulte de la transformation d'une image 2D en un vecteur 1D.

$$p(i) = [p(1), p(2), p(3), \dots, p(L)] \quad (2.33)$$

Où $p(i)$ est la valeur du niveau de gris du $i^{\text{ème}}$ pixel, $i \in [1, L]$.

La séquence correspondante permutée des pixels $p'(i)$ est telle que :

$$p'(i) = [p'(1), p'(2), p'(3), \dots, p'(L)] \quad (2.34)$$

$$\max(p_i) = \max(p'_i) \quad (2.35)$$

$$\sum_{i=1}^{i=n} p_i = \sum_{i=1}^{i=n} p'_i \quad (2.36)$$

Supposons que chaque pixel est représenté sur 8 bits, $p(i)$ la séquence binaire est représentée à l'équation suivante :

$$p(i) = [b_{i1}, b_{i2}, b_{i3}, b_{i4}, b_{i5}, b_{i6}, b_{i7}, b_{i8}] \quad (2.37)$$

Où b_{ij} est la valeur du $j^{\text{ème}}$ bit du $i^{\text{ème}}$ pixel. $b_{ij} = 0$ ou 1 , $j \in [1, 8]$.

Mathématiquement, pour une séquence binaire $[b_i]_{i=1}^n$, pour $i = 1, 2, 3, \dots, n$, sa séquence correspondante permutée $[b'_i]_{i=1}^n$ est telle que :

$$\max(b_i) = \max(b'_i) \quad (2.38)$$

$$\sum_{i=1}^{i=n} b_i = \sum_{i=1}^{i=n} b'_i \quad (2.39)$$

La diffusion chaotique

- Utilise la dynamique chaotique de certains phénomènes afin de pouvoir déplacer les pixels d'une image ou les bits de pixels d'une image (Congxu et al., 2015).
- Utilise des fonctions non linéaires, ou la résolution des équations non linéaires pour disposer les pixels ou les bits de pixels d'une image (Jianfeng et al., 2015; Eyebe et al., 2014).

2.2.3.2 Confusion de l'image

La confusion permet de rendre ambiguë la relation entre l'image originale et l'image chiffrée. La substitution, qui remplace un symbole par un autre, est le type le plus simple de confusion. La confusion chaotique utilise des fonctions linéaires ou des fonctions non linéaires pour substituer les pixels ou les bits de pixels par de nouvelles valeurs. En considérant la séquence binaire $[b_i]_{i=1}^n$ et les équations (2.33) et (2.37), leurs séquences correspondantes substituées $p'(i)$ et $[b'_i]_{i=1}^n$ sont telles que :

$$\max(p_i) \neq \max(p'_i) \quad (2.40)$$

$$\sum_{i=1}^{i=n} p_i \neq \sum_{i=1}^{i=n} p'_i \quad (2.41)$$

$$\max(b_i) \neq \max(b'_i) \quad (2.42)$$

$$\sum_{i=1}^{i=n} b_i \neq \sum_{i=1}^{i=n} b'_i \quad (2.43)$$

Nous allons présenter quelques fonctions de diffusion utilisées dans la littérature comme fonctions de cryptage.

2.2.4 Fonctions de hachage

Une fonction de hachage est une fonction rapide à calculer mais dont l'image réciproque est dans la classe des problèmes calculatoirement difficiles. C'est une fonction qui convertit une chaîne de longueur quelconque en une chaîne de taille inférieure et généralement fixe. L'intérêt est que cela permet l'usage de la cryptographie asymétrique sans engendrer trop de ralentissement, mais également d'assurer la provenance d'un fichier ainsi que son intégrité. Elles permettent notamment de diminuer la quantité d'information à crypter. Si l'image du texte en clair par la fonction de hachage est appelée l'empreinte de x , on peut par exemple ne crypter que l'empreinte tel qu'illustré à la figure 17. D'autre part, elles permettent de mettre au point des protocoles de signature électronique et d'authentification des messages ainsi que de vérifier l'intégrité d'un document. Elles sont beaucoup plus utilisées dans les algorithmes de chiffrement par blocs (Pareek et al., 2003 ; Congxu et al., 2015).

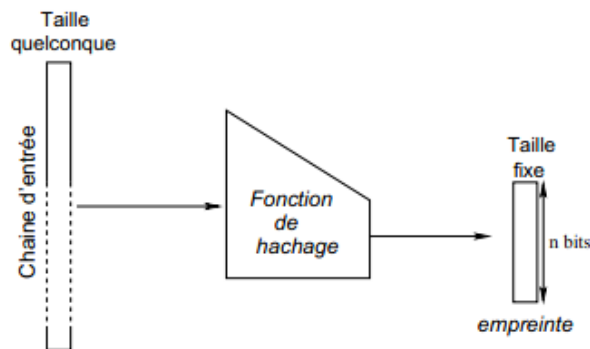


Figure 17 – Principe d'une fonction de hachage. Figure tirée de Dumas et al. (2006)

Les fonctions de hachage possèdent de nombreuses propriétés :

- Elles peuvent s'appliquer à n'importe quelle longueur de message M ;
- Elles produisent un résultat de longueur constante ;
- Il doit être facile de calculer $h = H(M)$ pour n'importe quel message M ;
- Pour un h donné, il est impossible de trouver x tel que $H(x) = h$. On parle de propriété à sens unique ;
- Pour un x donné, il est impossible de trouver y tel que $H(y) = H(x)$;

- Il est impossible de trouver x, y tels que $H(y) = H(x)$.

Pour les deux dernières propriétés on parle alors de résistance *faible* ou *forte* collision. Comme exemple de fonction de hachage nous présentons la fonction MD5 (Rivest, 1992) : c'est un modèle largement utilisé qui génère une valeur de hachage de 128 bits typiquement présentée sous la forme d'un nombre hexadécimal à 32 chiffres. En raison de sa bonne caractéristique de sécurité, même le changement d'un bit peut conduire à une différence significative entre deux images. Les valeurs initiales des cartes chaotiques sont générées par la valeur de hachage MD5, qui est calculée par l'équation suivante :

$$x_0 = \text{mod}(d_1 \otimes d_2 \otimes d_3 \otimes d_4 \otimes) / 255 \quad (2.44)$$

Où x_0 représente la valeur initiale de la carte chaotique. Sous certaines conditions cette valeur peut prendre la valeur 0 ou 1. Les valeurs d_1, d_2, d_3, d_4 sont extraites de la valeur de hachage MD5 de l'image à crypter. Cela signifie que la valeur de hachage MD5 de l'image originale étant représentée sur 128 bits soit : $b_1 b_2 b_3 \dots b_{127} b_{128}$, les 32 premiers bits sont utilisés pour générer les valeurs d_1, d_2, d_3, d_4 .

D'autres procédés de transformation aboutissent également à un message codé. De temps à autre, pour les images, les couleurs de deux pixels voisins ne sont pas indépendantes, et il sera plus judicieux de coder un ensemble de pixels comme un ensemble à la différence de coder chaque pixel indépendamment comme une valeur. On code alors par blocs de pixels, par une fonction de substitution. On peut citer alors : les chiffrements affines généraux, les chiffrements par fonction de décalage (les systèmes discrets), les chiffrements par transformée (transformée de Fourier, transformée de Fourier discrète, transformée en cosinus discret).

2.2.5 Chiffrements affines généraux

Ce sont des chiffrements dans lesquels le cryptosystème opère sur des blocs de caractères. Les chiffrements affines sont appelés fonctions affines par analogie avec les droites affines de la géométrie du plan réel. Elles peuvent être généralisées aux chiffrements polygrammes. Plutôt qu'une application $E(m, K) \rightarrow c = ma + b$, avec $K = (a, b)$. Nous pouvons appliquer une transformation linéaire des vecteurs u :

$$u = (m_1, m_2, \dots, m_n) \rightarrow (c_1, c_2, \dots, c_n) = u \times A + v \quad (2.45)$$

Par une certaine matrice $A_{ij} = (a_{ij})$ et un vecteur $v = (b_1, b_2, \dots, b_n)$

Chaque mot de caractères m_1, m_2, \dots, m_n est identifié avec le vecteur $u = (m_1, m_2, \dots, m_n)$. La multiplication de matrice est définie comme d'habitude, de sorte que :

$$c_{ij} = \sum_{i=1}^n m_{ij} a_{ij} + b_j \quad (2.46)$$

Avec le résultat interprété modulo p comme élément de $\mathbb{Z}/p\mathbb{Z}$, p représente le nombre de caractères ou le nombre de niveaux de gris dans le cas d'une image. La complexité des fonctions affines augmente exponentiellement quand le nombre p de caractères augmente. Ces cryptosystèmes rendent la cryptanalyse plus dure en détruisant les fréquences de caractère, préservées sous des chiffrements par substitution simples.

2.2.6 Courbes elliptiques

La théorie des courbes elliptiques a été proposée par deux chercheurs Miller (1985) et Koblitz (1987), de façon totalement indépendante. Ce type de cryptographie basé sur le modèle asymétrique, permet aussi bien de chiffrer les données sur le modèle symétrique en utilisant une courbe elliptique comme fonction de cryptage en utilisant l'arithmétique des courbes elliptiques. On utilise souvent l'abréviation ECC, pour *Elliptic Curve Cryptography*. Les clés utilisées sont plus courtes pour une sécurité égale ou supérieure. La théorie sous-jacente, ainsi que l'implémentation sont plus complexes, ce qui explique le fait que cette technologie soit moins répandue. Toutefois, de par la nécessité de traiter plus rapidement l'information, de gérer des quantités de données importantes et de miniaturiser au maximum, les avantages de cette technique poussent la recherche vers de nouveaux horizons. D'une manière générale, sur un ensemble donné \mathbb{R} par exemple, les courbes elliptiques seront considérées comme l'ensemble des couples (x,y) tels que :

$$y^2 = x^3 + ax + b \quad (2.47)$$

Les variables et coefficients prennent des valeurs dans un ensemble $[0, p - 1]$ pour un certain nombre premier p , et où toutes les opérations sont calculées *modulo* p . L'équation devient :

$$y^2 \text{ mod}(p) = x^3 + ax + b \text{ mod}(p) \quad (2.48)$$

Une particularité importante des EC (*elliptic curve*) est sa structure de groupe additif : Si 3 points sur une EC sont alignés, leur somme vaut O (point à l'infini).

- O est l'identité pour l'addition : $O = -O$;
- Pour n'importe quel point $P + O = P$;
- L'opposé d'un point $P(x, y)$ est $P(x, -y)$
- Pour additionner 2 points P et Q , on trace la droite les reliant. Cela nous donne un point d'intersection R obtenu en utilisant la droite qui passe par P et Q si $P \neq Q$ ou la tangente à la courbe en P , si $P = Q$ comme indiqué dans les figures ci-dessous.

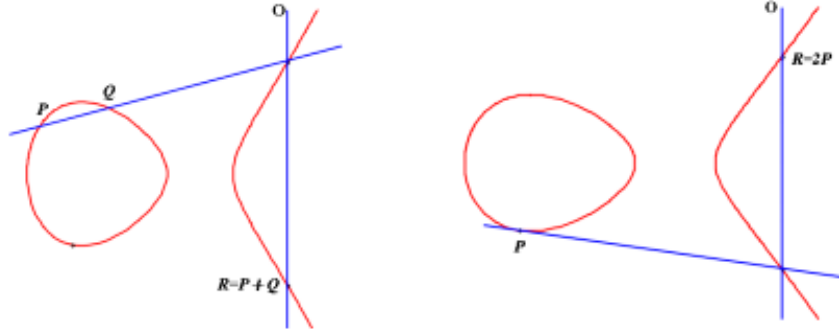


Figure 18 – Géométrie explicite des EC. Figure tirée de Nitaj (2011)

Pour les points $P, Q \in EC_p(a, b)$

- $P + O = P$;
- Si $P = (x_p, y_p)$ et $Q(x_Q, y_Q)$ avec $P \neq -Q$ alors $R = P + Q = (x_R, y_R)$ comme suit :

$$x_R = (\lambda^2 - y_R - x_Q) \bmod(p) \quad (2.49)$$

$$y_R = [\lambda(y_R - x_R) - y_p] \bmod(p) \quad (2.50)$$

où

$$\lambda = \begin{cases} \left(\frac{y_Q - y_P}{x_Q - x_P} \right) \bmod(p) & \text{si } P \neq Q \\ \left(\frac{3x_P^2 + a}{2y_P} \right) \bmod(p) & \text{si } P = Q \end{cases} \quad (2.51)$$

- La multiplication est définie comme une répétition d'additions (exemple : $5P = P + P + P + P + P$).

2.2.7 Transformée de Fourier et transformée de Fourier discrète

La transformée de Fourier est un procédé de codage permettant de passer d'une représentation à une autre notamment du domaine spatial au domaine fréquentiel. Comme tout procédé de codage, il est associé à son inverse, le décodage, et est exprimé par les formules suivantes :

$$\text{Codage : } H(f) = \int_{-\infty}^{+\infty} h(t)e^{-2\pi ift} dt \quad (2.52)$$

$$\text{Décodage : } h(t) = \int_{-\infty}^{+\infty} H(f)e^{2\pi ift} dt \quad (2.53)$$

Où $h(t)$ est le signal à coder et $H(f)$ le signal obtenu très utile pour le codage du son.

Pour une image dans le domaine spatial de taille $M \times N$, sa transformée de Fourier est donnée par l'équation :

$$\text{Codage : } F(l, c) = \sum_{i=0, j=0}^{M-1, N-1} f(m, n)e^{-i2\pi(\frac{im}{M} + \frac{cn}{N})} \quad (2.54)$$

$$\text{Décodage : } f(m, n) = \frac{1}{M \times N} \sum_{i=0, j=0}^{M-1, N-1} F(l, c)e^{i2\pi(\frac{im}{M} + \frac{cn}{N})} \quad (2.55)$$

Où $f(m, n)$ est la valeur du pixel de l'image se trouvant dans la ligne m et la colonne n , et $F(l, c)$ est la valeur du coefficient correspondant dans l'espace de Fourier se trouvant dans la ligne l et la colonne c . La transformée de Fourier dite discrète suit le même principe, mais avec des fonctions discrètes.

$$\text{Codage : } H_k = \frac{1}{\sqrt{n}} \sum_{j=0}^{n-1} h_j e^{-\frac{2\pi k j}{n}} \quad (2.56)$$

$$\text{Décodage : } h_j = \frac{1}{\sqrt{n}} \sum_{k=0}^{n-1} H_k e^{\frac{2\pi k j}{n}} \quad (2.57)$$

Où $h_0, \dots, h_j, \dots, h_{n-1}$ sont des informations discrètes et $H_0, \dots, H_j, \dots, H_{n-1}$ les valeurs discrètes obtenues.

La transformée discrète en cosinus (DCT) dérive directement de la transformée de Fourier discrète, pour une fonction h discrète, mais au lieu d'être une fonction du temps (ce qui est par exemple un bon modèle pour un son), est une fonction de l'espace (ce qui permet de coder une image). Les formules de codage et de décodage sont les suivantes :

$$\text{Codage : } H(i, j) = \frac{1}{\sqrt{n}} \sum_{x=0, y=0}^{n-1, n-1} h(x, y) \cos\left(\frac{(2x+1)i\pi}{2n}\right) \cos\left(\frac{(2y+1)j\pi}{2n}\right) \quad (2.58)$$

$$\text{Décodage : } h(x, y) = \frac{1}{\sqrt{n}} \sum_{i=0, j=0}^{n-1, n-1} H(i, j) \cos\left(\frac{(2x+1)i\pi}{2n}\right) \cos\left(\frac{(2y+1)j\pi}{2n}\right) \quad (2.59)$$

Où $H(i, j)$ sont les coefficients DCT de la ligne i et la colonne j et $h(x, y)$ sont les valeurs des pixels de l'image d'origine de la ligne x et la colonne y (Lala et al., 2009 ; Munir, 2012).

2.3 Outils d'analyse de sécurité

En appliquant un algorithme de chiffrement sur une image, les pixels de l'image chiffrée doivent être différents et indépendants (faible corrélation) de ceux de l'image originale. Ceci peut être visible par simple inspection de l'image cryptée. Toutefois, la simple inspection visuelle reste insuffisante pour juger le chiffrement d'une image, ainsi pour l'analyse des performances d'un algorithme de chiffrement on utilise plusieurs paramètres usuels de mesure tels que : Le PSNR (*Peak Signal to Noise Ratio*), SSIM (*Structural Similarity Index*), MAE (*Mean Absolute Error*), MSE (*Mean Squared Error*). Ces dernières métriques sont utilisées dans l'analyse statistique. D'autres ont trait à la robustesse : l'analyse différentielle, l'analyse de l'espace de la clef mais également l'analyse de la sensibilité de la clé.

2.3.1 Analyse statistique

L'analyse statistique permet de déceler quelques faiblesses dans un algorithme de cryptage en effectuant des calculs statistiques sur les différentes images utilisées et issues du cryptosystème, c'est-à-dire il analyse statistiquement les entrées et les sorties de ce système en calculant certaines métriques dont quelques-unes ont été citées ci-dessus, on pourra aussi parler de l'entropie. Dans un même objectif nous allons également présenter l'étude par histogramme et par corrélation de pixels adjacents sur les images.

- Le PSNR : est un paramètre qui évalue le degré de distorsion entre l'image originale et l'image cryptée la formule utilisée est la suivante :

$$PSNR = 10 \log \left\{ \frac{M \times N \times \max[I(m, n)]^2}{\sum_{m=0}^{M-1} \sum_{n=0}^{N-1} [I(m, n) - E(m, n)]^2} \right\} \quad (2.60)$$

$M \times N$ représente le nombre total de pixels dans l'image. I représente l'image originale et E l'image cryptée, et (m, n) représente la position du pixel sur l'image.

- Le MAE : est l'erreur absolue moyenne, c'est une mesure objective basée sur la qualité de l'image et désigne le degré de ressemblance entre l'image cryptée et l'image originale. Sa formule est indiquée ci-dessous :

$$MAE = \frac{1}{M \times N} \sum_{m,n} \frac{|I(m, n) - E(m, n)|}{\max[I(m, n)]} \quad (2.61)$$

- Le MSE : c'est l'erreur quadratique moyenne, lorsque le $MSE = 0$, cela indique que l'image testée est identique à celle d'origine et une valeur élevée de l'indice

MSE indique que les deux images testées sont différentes. La formule utilisée pour le calculer est la suivante :

$$MSE = \frac{1}{M \times N} \sum_{m,n}^{M-1,N-1} [I(m,n) - E(m,n)]^2 \quad (2.62)$$

- Le SSIM : mesure la similarité de structure entre les images originales et les images cryptées, certains travaux estiment que cet indicateur est de meilleure qualité que le PSNR (Wang et al., 2004). La formule utilisée pour la calculer est la suivante :

$$SSIM = \frac{(\mu_I \mu_E + C_1)(2\sigma_{IE} + C_2)}{(\mu_I^2 + \mu_E^2 + C_1)(\sigma_I^2 + \sigma_E^2 + C_2)} \quad (2.63)$$

μ_I représente la moyenne de l'image originale, et μ_E celle de l'image cryptée, σ_I est la variance de l'image originale, et σ_E celle de l'image cryptée, σ_{IE} représente la covariance de l'image originale et l'image cryptée. Les deux variables C_1 et C_2 sont utilisées pour stabiliser la division quand le dénominateur est très proche de zéro.

- L'entropie : l'entropie de l'information est une caractéristique définit pour exprimer le degré d'incertitude dans le système. L'entropie de l'information peut être exprimée comme suit :

$$H = \sum_{i=0}^{2^M-1} P(s_i) \ln \left(\frac{1}{P(s_i)} \right) \quad (2.64)$$

Où P représente la probabilité d'apparition du symbole s_i , dans le cas d'une image il correspond à la probabilité d'apparition des valeurs des pixels d'une image codée sur M bits. Pour une source d'émission de 2^M symboles purement aléatoire, possédant une même probabilité d'apparition $P(s_i) = 1/2^M$, l'entropie $H(s) = M$. Si l'image cryptée possède une valeur d'entropie inférieure à M , il existe un certain degré de prévisibilité qui menace sa sécurité.

2.3.2 Analyse par histogramme

Un histogramme est la distribution des intensités des pixels d'une image, c'est-à-dire le nombre de pixels pour chaque intensité lumineuse.

Cette analyse s'effectue par visualisation des différents histogrammes des images originales et cryptées afin d'observer la distribution des valeurs des pixels après passage dans le cryptosystème. Si la distribution des valeurs des pixels n'a pas été efficacement masquée par le chiffrement, elle peut conduire à une fuite d'informations. Un exemple de masquage faible est illustré à la figure 19.

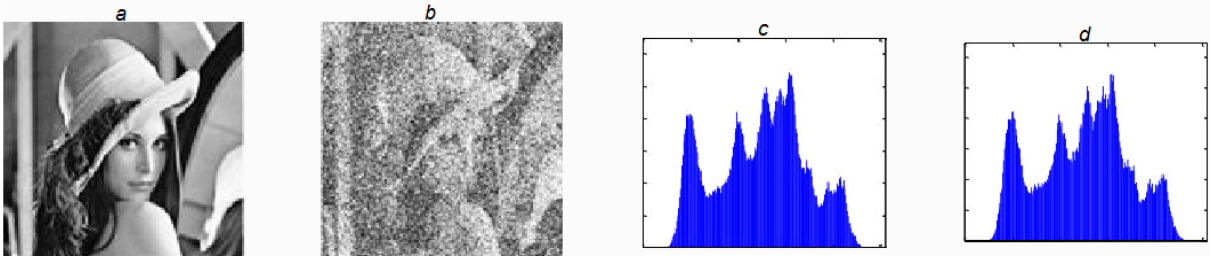


Figure 19 – Masquage par bruit additif de l'image. (a) Lena, (b) image cryptée de Lena, (c) histogramme de Lena, (d) histogramme de l'image cryptée de Lena

2.3.3 Analyse par Corrélacion

Le calcul du coefficient de corrélation entre les pixels de l'image cryptée permet l'évaluation de la qualité de cryptage des cryptosystèmes. Ce coefficient peut être représenté dans un graphique, il est mesuré entre -1 et 1 dans le cas où les valeurs des pixels sont proches, et très proche de zéro dans le cas où les valeurs des pixels ne sont pas liées. Un cryptanalyste peut utiliser cette information lors d'une attaque statistique pour trouver la clé secrète et de récupérer l'image originale, par conséquent, l'image cryptée doit avoir une corrélation proche de 0 . La formule utilisée pour calculer cette métrique est la suivante :

$$Cr = \frac{N \times \sum_{i=0}^N (x_i \times y_i) - \sum_{i=0}^N x_i \times \sum_{i=0}^N y_i}{\sqrt{\left[N \times \sum_{i=0}^N (x_i)^2 - \left(\sum_{i=0}^N x_i \right)^2 \right] \times \left[N \times \sum_{i=0}^N (y_i)^2 - \left(\sum_{i=0}^N y_i \right)^2 \right]}} \quad (2.65)$$

Où x, y sont les valeurs des niveaux de gris de deux pixels adjacents, et N le nombre de paires de pixels.

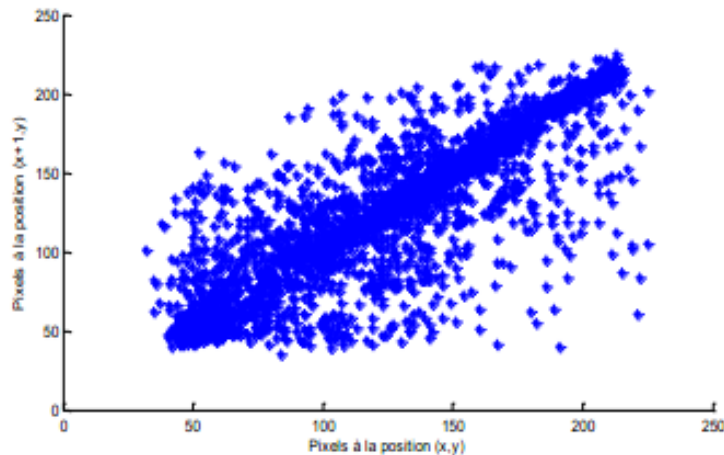


Figure 20 – Distribution des paires de pixels adjacents horizontalement dans l'image Lena

On peut étudier la corrélation des paires de pixels d'une image par la représentation

de la distribution géométrique (diagonale, anti diagonale, verticale et horizontale) de ces paires de pixels sur une image. Un exemple de représentation est illustré sur la figure 20.

2.3.4 Analyse différentielle

Dans un système cryptographique (symétrique ou asymétrique) d'image la haute sensibilité aux petits changements dans l'image originale devrait être une propriété fondamentale. Un adversaire peut faire une légère modification (un seul pixel par exemple) de l'image cryptée, et il observe le changement du résultat. De cette façon, il peut être en mesure de trouver une relation significative entre l'image claire et celle cryptée. Si un changement mineur dans l'image en clair peut provoquer un changement significatif dans l'image cryptée, à l'égard de la diffusion et la confusion, alors l'attaque différentielle devient inutile. Pour calculer l'influence d'un changement d'un seul pixel sur l'image cryptée par n'importe quel algorithme et par conséquent la résistance aux attaques différentielles, deux grandeurs communes peuvent être utilisées :

- Le NPCR (*Number of Pixel Change Rate*) : mesure le pourcentage du nombre de pixels différents par rapport au nombre total de pixels entre deux images ;
- L'UACI (*Unified Average Change Intensity*) : mesure la moyenne de différence d'intensité entre les deux images.

Ces deux métriques sont calculées par les formules suivantes :

$$NPCR = \frac{1}{M \times N} \sum_{i=1}^M \sum_{j=1}^N D(i, j) \quad (2.66)$$

Avec,

$$D(i, j) = \begin{cases} 0 & \text{si } I_{m0}(i, j) = I_{mc}(i, j) \\ 1 & \text{si } I_{m0}(i, j) \neq I_{mc}(i, j) \end{cases} \quad (2.67)$$

$$UACI = \frac{100}{M \times N} \sum_{i=1}^M \sum_{j=1}^N \frac{|I_{m0}(i, j) - I_{mc}(i, j)|}{255} \quad (2.68)$$

Où $M \times N$ représente la taille totale d'une image. $I_{m0}(i, j)$ et $I_{mc}(i, j)$ sont les valeurs des pixels à la position (i, j) des deux images cryptées dont les images originales ne diffèrent que d'un seul pixel.

2.3.5 Analyse de la sensibilité de la clé

La sensibilité de la clé d'un cryptosystème peut être observée par deux méthodes différentes :

- L'image cryptée doit être très sensible à la clé secrète ; i.e. si on utilise deux clés légèrement différentes pour crypter la même image, alors les deux images cryptées doivent être complètement indépendantes l'une par rapport à l'autre (faible corrélation) ;
- L'image cryptée ne peut pas être décryptée correctement si la clé secrète est légèrement modifiée à la phase de décryptage.

2.3.6 Analyse de l'espace des clefs

L'espace des clefs est le nombre total de clés différentes qui peuvent être utilisées dans le cryptage et le décryptage. Dans la littérature il est absolument nécessaire que la taille de l'espace de clé soit suffisamment large afin de pouvoir résister à une attaque exhaustive (Stinson, 2007).

2.3.7 Analyse de la complexité en temps

En dehors des considérations de sécurité, d'autres tests de performance pour l'analyse des cryptosystèmes sont également importants. Cela inclut la vitesse de chiffrement/déchiffrement, en particulier pour des applications Internet multimédia en temps réel : la télémédecine, les échanges de données militaires, les images personnelles, les conférences vidéo, les systèmes biométriques, etc. En fait, le temps d'exécution réelle d'un système de cryptage dépend de nombreux facteurs comme, le type d'ordinateur utilisé, la structure du processeur, la taille de la mémoire, le système d'exploitation, le langage de programmation, les options de compilation, les compétences de programmation et d'optimisation de code. Ainsi, il est plus judicieux de faire une comparaison explicite pour deux ou plusieurs systèmes de cryptage en utilisant le même environnement.

2.3.8 Cryptanalyse

Selon le principe de Kerckhoffs (1883), lorsque qu'on cryptanalyse un système de cryptage, une hypothèse générale est que : le cryptanalyste peut acquérir les informations sur la conception et le fonctionnement du cryptosystème étudié, à savoir, pour tout chercheur, il ou elle peut tout savoir sur le cryptosystème sauf les clés secrètes pour le chiffrement et le déchiffrement. De nos jours ce critère est une norme de base pour tout système de cryptage dans les réseaux de communication sécurisés. Par conséquent, il est important de faire une cryptanalyse du nouveau cryptosystème en effectuant une ou plusieurs des, cinq attaques

typiques (Rhouma et al., 2009) présenté dans le chapitre 1 ; dont on rappellera les noms : l'attaque sur texte chiffré seul, l'attaque à texte clair connu, l'attaque à texte clair choisi, l'attaque à texte chiffré connu, l'attaque adaptative à texte chiffré choisi.

2.3.9 Méthodes numériques

Pour les différentes simulations numériques, notamment pour l'évaluation des systèmes chaotiques, divers logiciels sont utilisés, pour les simulations expérimentales (Proteus, Spice, Multisim, etc.), pour les simulations assistées par ordinateur en temps réel (Matlab, Scilab, et bien d'autres) avec des compilateurs plus ou moins variés (C, Python, Java, C++, etc). L'une des méthodes les plus présentes dans la littérature pour le calcul numérique est la méthode de *Runge Kutta d'ordre 4*, dans cette section nous allons faire une présentation brève de cette méthode.

La méthode de *Runge Kutta d'ordre 4*, notée RK4 est une méthode explicite très populaire dans la résolution d'équations différentielles d'ordre supérieure (ordre de dérivation supérieure à 1).

On considère le problème différentiel du 1er ordre à valeur initiale suivant :

$$y' = f(x, y), y(x_0) = y_0 \quad (2.69)$$

Pour approcher la solution $y(x)$ de l'équation (2.69) sur l'intervalle $a \leq x \leq b$, on choisit $N + 1$ points distincts, x_0, x_1, \dots, x_N , tels que $a = x_0 < x_1 < x_2 < \dots < x_N = b$, et l'on construit les valeurs approchées $y_n \approx y(x_n)$ en $x_n, n = 0, 1, \dots, N$, on utilise le théorème de Taylor avec h qui représente le pas d'intégration, $h = x_{n+1} - x_n$:

$$y(x_{n+1}) = y(x_n) + y'(x_n)h + \frac{y''(x_n)}{2!}h^2 + \dots + \frac{y^{(5)}(x_n)}{5!}h^5 \quad (2.70)$$

Pour ζ_n une perturbation entre x_n et x_{n+1} , $n = 0, 1, \dots, N - 1$. On détermine les poids : b_1, b_2, b_3, b_4 les incréments en x : pour que l'équation (2.70) devienne :

$$y(x_{n+1}) = b_1k_1 + b_2k_2 + b_3k_3 + b_4k_4 + O(h^5) \quad (2.71)$$

avec,

$$\begin{aligned} k_1 &= hf(x_n, y_n) \\ k_2 &= hf\left(x_n + \frac{1}{2}h, y_n + \frac{1}{2}k_1\right) \\ k_3 &= hf\left(x_n + \frac{1}{2}h, y_n + \frac{1}{2}k_2\right) \end{aligned}$$

$$k_4 = hf(x_n + h, y_n + k_3)$$

Pour obtenir les incréments en y :

$$y_{n+1} = y_n + \frac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4) \quad (2.72)$$

Conclusion

Dans ce chapitre, nous avons détaillés les différents outils nécessaires à la construction d'un cryptosystème. Nous avons également abordé quelques fonctions mathématiques utilisées pour le brouillage des images. Enfin nous avons décrit quelques méthodes d'évaluation des algorithmes de chiffrement d'images. Dans le chapitre suivant, nous allons introduire un nouveau cryptosystème utilisant le mixage de cartes chaotiques et nous allons l'appliquer à des images.

Chapitre 3

Nouveaux cryptosystèmes basés sur le mixage et la fusion de cartes de données : Colpitts-Duffing, et Duffing-Hartley

Introduction

Malgré la pluralité de systèmes de cryptage dans la littérature (Li et al., 2007 ; Zhang et al., 2015 ; Radu et al., 2014 ; Xingyan et Chuanming, 2016 ; Jongseok et al., 2016), il semble toujours indispensable de proposer et de déployer des cryptosystèmes de plus en plus robustes, rapides et surtout plus efficaces. L'utilisation des séries de données provenant des systèmes électriques ou électroniques semble fournir un bon compromis dans l'élaboration des protocoles cryptographiques utilisant des séries de données pseudo aléatoires (Luby, 1996), ainsi que des fonctions mathématiques de plus en plus complexes (Boneh, 1998). Ces séries de données seront utiles dans la mesure où on peut déterminer leur caractère, régulier ou non (Gopal et al., 2013 ; Priya et Sankpal, 2014 ; Eyebe et Koepf, 2015). C'est dans cette optique que nous avons développé un cryptosystème à clef privée, basé d'abord sur le mixage de deux oscillateurs chaotiques : Colpitts, et Duffing, puis sur la fusion de cartes de données provenant de l'oscillateur de Hartley et de Duffing. Cette fusion va produire des séquences de données qui seront étudiées par de nouveaux tests (Eyebe et Koepf, 2015) afin de déterminer si elles seront chaotiques ou non, ses séquences de données seront enfin utilisées dans le cryptosystème proposé. Dans un premier temps, nous exposons les différents oscillateurs choisis afin de produire des séquences de données dont nous nous

servirons. Ensuite nous allons présenter un algorithme de cryptage d'image utilisant une carte chaotique, puis un second algorithme de chiffrement à deux cartes chaotiques dans le but de révéler les améliorations apportées au premier algorithme. Nous évoquerons aussi les similitudes observées. L'organigramme de cryptage, et la fonction de cryptage implémentés seront à chaque fois décrites. Nous allons utiliser plusieurs images (image en niveau de gris et image de couleur) afin de pouvoir généraliser l'application du cryptosystème à deux cartes chaotiques sur tout type d'image. Enfin, après avoir effectué plusieurs tests statistiques, nous allons effectuer une cryptanalyse du système de cryptage afin de s'assurer de sa sécurité. Les résultats numériques sont obtenus sous le logiciel Matlab.

3.1 Oscillateurs utilisés

3.1.1 Oscillateur de Colpitts

Le choix de cet oscillateur est motivé par la simplicité de sa structure et de son facteur non linéaire, car il comporte une non linéarité intrinsèque liée à la caractéristique exponentielle du transistor présent dans son montage. Nous allons nous limiter à la présentation du système différentiel de Colpitts, pour une étude plus approfondie une lecture de (Matsumoto, 1997; Effa et al., 2009; Kenfack et Tiedeu, 2014) apportera plus de détails. Le système de Colpitts est représenté par le système suivant :

$$\begin{cases} \dot{x} = -\sigma(-1 + e^{(-y-1)}) + \chi z \\ \dot{y} = -\gamma(-1 + e^{(-y-1)}) + \mu z \\ \dot{z} = -\rho(x + y) - \lambda z \end{cases} \quad (3.1)$$

C'est un ensemble d'équations non linéaires autonomes à coefficients constants. Pour avoir un comportement chaotique, les paramètres du système sont donnés comme suit : $\sigma = 1,915\eta$, $\chi = 1,92\eta$, $\gamma = 0,008\eta$, $\mu = 1,92\eta$, $\rho = 1,921\eta$ avec $\eta = 3,9$ qui est un paramètre de multiplication. Pour la résolution numérique, nous utilisons la méthode de RK4 avec un pas de calcul normalisé $\Delta t = 10^{-3}$, ainsi que le triplet $(x_0, y_0, z_0) = (0, 4; 0, 1; 0, 1)$ comme conditions initiales pour résoudre le système (3.1). Les variables x, y, z en fonction du temps sont obtenus à l'issue de la simulation, la représentation d'une variable ainsi que l'un des portraits de phase du système sont localisées sur la figure 21.

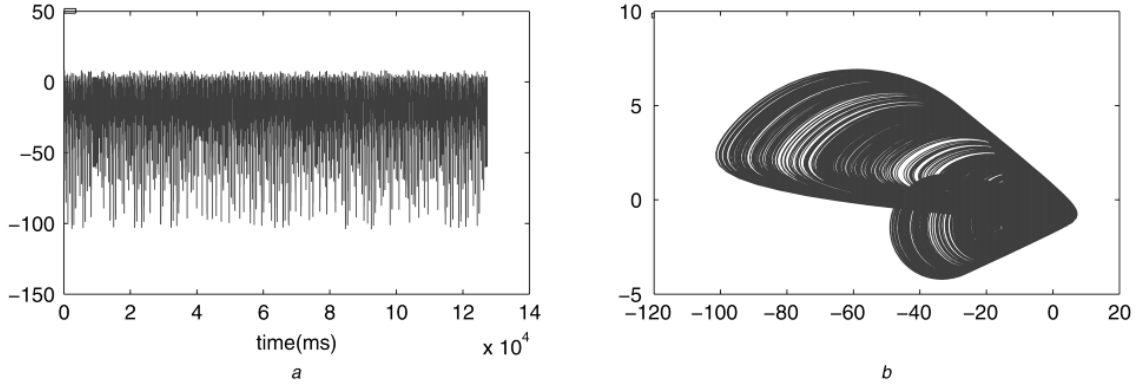


Figure 21 – Système de Colpitts. (a) Variable en fonction du temps, (b) portrait de phase du système

3.1.2 Oscillateur de Duffing

L'équation de Duffing fait partie des systèmes qui permettent d'étudier une dynamique non linéaire. Elle fut établie au début du siècle par l'ingénieur Georg Duffing, dans le but de modéliser les vibrations forcées d'une machine Industrielle. La forme la plus générale de l'équation de Duffing est présentée à l'équation (3.2) :

$$\ddot{x} + \varepsilon\dot{x} + bx^3 + \omega_0^2x = B \cos(\omega t + \varphi) \quad (3.2)$$

Dépendamment des paramètres choisis, l'équation peut prendre diverses formes. Par exemple, le système est n'est pas forcé et n'est pas amorti, $B = 0$, $\varepsilon = 0$. En adoptant des valeurs positives, l'équation (3.2) devient :

$$\ddot{x} + \varepsilon\dot{x} + bx^3 + \omega_0^2x = 0 \quad (3.3)$$

Les courbes de portrait de phase deviennent alors fermées, le portrait de phase obtenu est celui de la figure 22.

Si $\omega_0 = 0$, $\varphi = 0$, l'équation devient :

$$\ddot{x} + \varepsilon\dot{x} + bx^3 = B \cos(\omega t) \quad (3.4)$$

Si l'équation (3.2) contient six paramètres $\varepsilon, b, \omega_0, B, \omega, \varphi$ il n'en faut que trois parmi ceux-ci pour explorer les différents types d'évolution du système. En effet deux des six paramètres peuvent être considérés comme des facteurs d'échelle vis-à-vis du temps et de l'amplitude. Si nous posons : $\tilde{x} \rightarrow ax$, $\tilde{t} \rightarrow ct$ et en choisissant $a = B\omega^{-2}$ et $c = \omega^{-1}$, on obtient une équation qui ne contient que trois paramètres :

$$\ddot{\tilde{x}} + \tilde{r}\dot{\tilde{x}} + \tilde{\alpha}\tilde{x} + \tilde{\beta}\tilde{x}^3 = \cos \tilde{t} \quad (3.5)$$

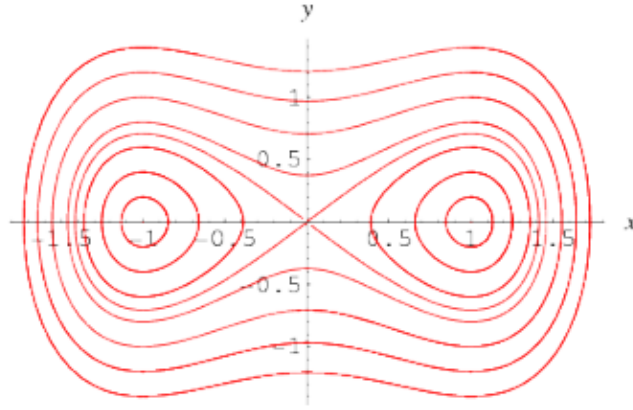


Figure 22 – Portait de phase de Duffing en l’absence du second membre

Avec $\tilde{r} = \varepsilon\omega^{-1}$, $\tilde{\alpha} = \omega_0^2\omega^{-2}$, $\tilde{\beta} = bB^2\omega^{-6}$.

Nous choisisons pour la suite de considérer l’influence de trois paramètres ε, b, B . L’équation différentielle du second ordre peut se ramener à un système équivalent de trois équations différentielles de premier ordre en introduisant d’abord une variable $z = t$, puis en posant l’équation de base telle que :

$$\dot{x} = y \quad (3.6)$$

et,

$$\dot{y} = -\varepsilon y - \omega_0^2 x - bx^3 + B \cos(\omega z) \quad (3.7)$$

On obtient alors le système suivant :

$$\begin{cases} \dot{x} = y \\ \dot{y} = -\varepsilon y - \omega_0^2 x - bx^3 + B \cos(\omega z) \\ \dot{z} = \omega \end{cases} \quad (3.8)$$

Ce système autonome dans lequel le temps n’apparaît pas explicitement fait intervenir trois variables x, y, z indépendantes. La résolution numérique du système différentiel est effectuée par la méthode de RK4 avec un pas de calcul normalisé $\Delta t = 10^{-3}$, les paramètres normalisés : $\varepsilon = 0,1$, $b = 0,25$, $B = 400$; $\omega = 0,8$, avec les conditions initiales choisies $(x_0, y_0, z_0) = (0; 1; 1)$. La représentation dans l’espace des phases (figure 23) du système peut être considérée comme la projection des trajectoires décrites dans l’espace tridimensionnel (x, y, z) . On utilise alors l’évolution d’une variable par rapport au temps, ou la coupe de Poincaré (figure 24) pour caractériser plus précisément les comportements observés.

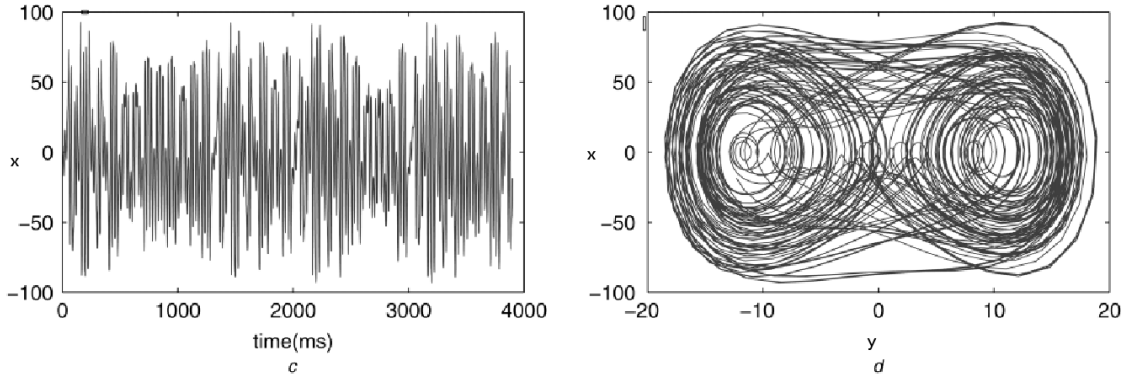


Figure 23 – Oscillateur de Duffing avec second membre. (c) Variable en fonction du temps. (d) Diagramme de phase de deux variables du système

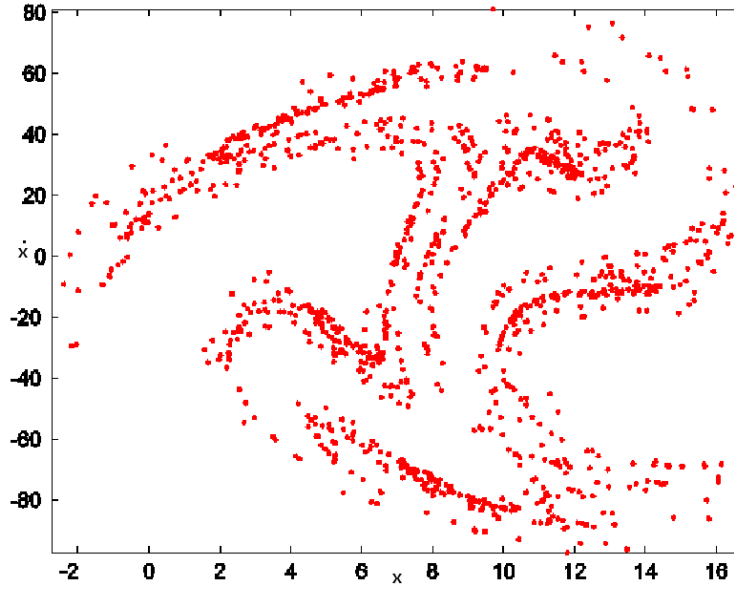


Figure 24 – Section de Poincaré de l'Oscillateur de Duffing

3.1.3 Oscillateur de Hartley

Les équations (3.9, 3.10 et 3.10) régissent le fonctionnement du système de Hartley.

$$\begin{cases} \dot{x}_1 = y_1 - z_1 - ax_1 \\ \dot{y}_1 = q - x_1 - b_1y_1 - F(z_1) \\ \dot{z}_1 = dx_1 - ez_1 + F^*(z_1) \end{cases} \quad (3.9)$$

$$F(z_1) = \begin{cases} -\frac{1}{V_{TH}}(h + m \times z_1) & si \ z_1 \geq \lambda \\ s & si \ z_1 < \lambda \end{cases} \quad (3.10)$$

$$F^*(z_1) = \begin{cases} 0 & \text{si } z_1 \geq \lambda \\ f(g + a_1 z_0) & \text{si } z_1 < \lambda \end{cases} \quad (3.11)$$

Les paramètres choisis pour notre oscillateur sont les suivants : $V_{TH} = 0,75$, $a = 0,706$, $a_1 = 0,0037$, $b_1 = 0,0222$, $d = 1$, $e = 1,1495$, $g = -0,013$, $h = -0,0018139$, $\lambda = 3,0554$, $m = 511,18$, $q = 5$, $s = 0,2667$, $\varepsilon = 0,1$, $b = 0,25$, $B = 400$, $\omega = 0,8$.

Les systèmes d'équations sont résolus en utilisant la méthode de RK4 avec un pas de calcul normalisé $\Delta t = 10^{-3}$. Pour des conditions initiales $(x_{10}, y_{10}, z_{10}) = (0, 2 : 0, 5; 0, 5)$, les portraits de phase du système de Hartley en 2D et 3D sont esquissés sur la figure 25.

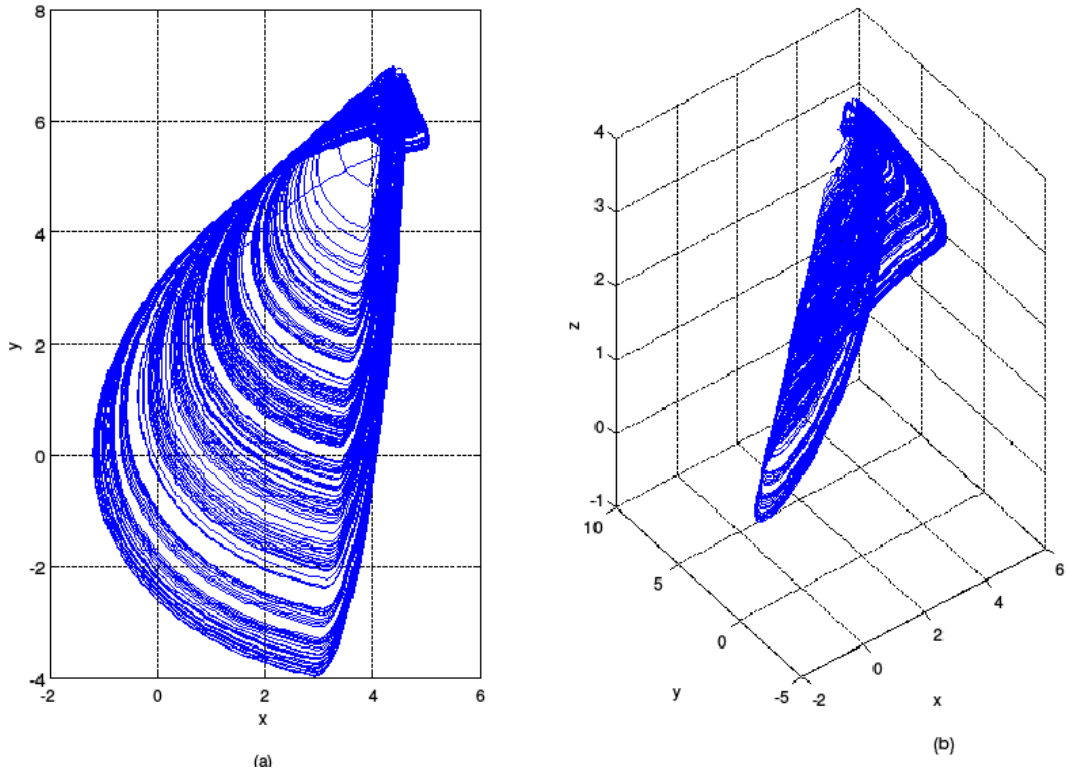


Figure 25 – Système de Hartley. (a) portrait de phase 2D, (b) portrait de phase 3D

Afin de s'assurer que nos données sont chaotiques nous représentons l'évolution de la dynamique des exposants de Lyapunov de nos systèmes sur la figure 26.

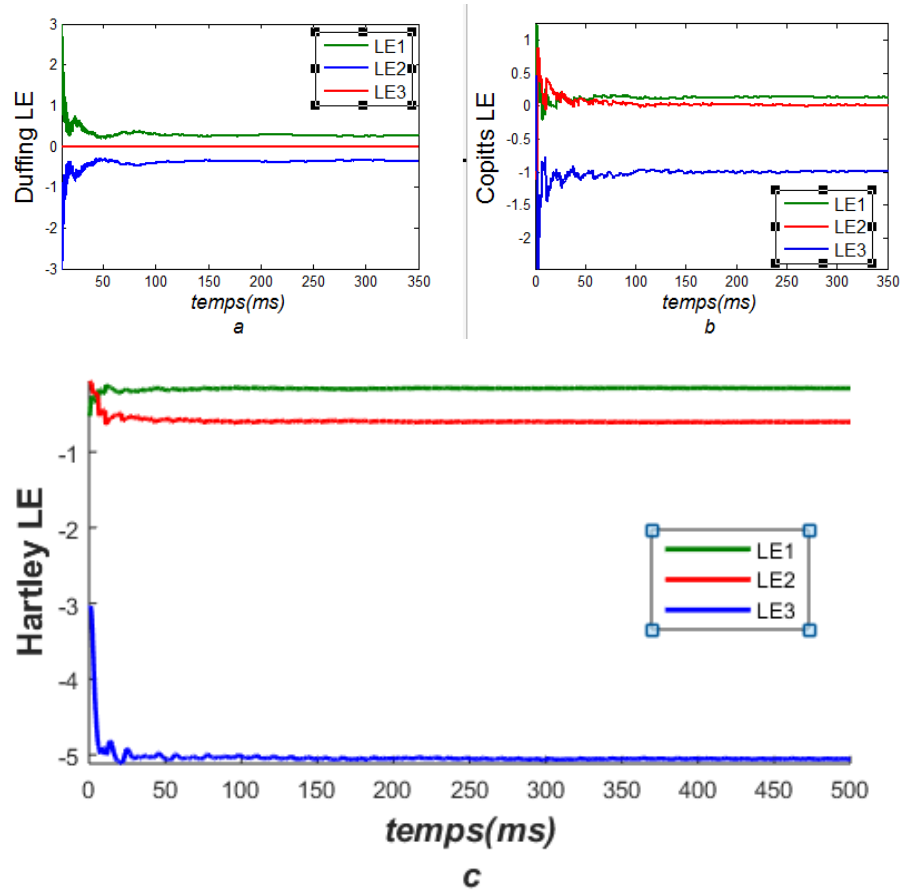


Figure 26 – Dynamique des exposants de Lyapunov. (a) système de Duffing, (b) système de Colpitts, (c) Système de Hartley

3.2 Description de la méthode de cryptage utilisant une carte chaotique

La procédure utilisée est la suivante :

1. Choisir une image sp de taille $M \times N$;
2. Choisir trois valeurs qui serviront de conditions initiales pour le système chaotique ;
3. Choisir le paramètre k_e , utilisé comme clé dans la fonction de cryptage ;
4. Créer une carte chaotique en additionnant les séquences chaotiques issues du système chaotique utilisé, et la stocker dans un vecteur K^* comme suit :

Pour i allant de 1 à $M \times N$

Faire

$$K^*(i) \leftarrow x_0(i) + y_0(i) + z_0(i)$$

Fin pour

5. Redimensionner le vecteur K^* en une matrice K de taille $M \times N$.
6. Le brouillage des pixels est effectué par la fonction de cryptage suivante :

$$cp \leftarrow [k_e \times (sp) + (1 - k_e) \times (10^{12}) \times \text{round}(K)] \text{mod}(2^8)$$

On obtient ainsi l'image cryptée. L'organigramme du schéma de cryptage est le suivant :

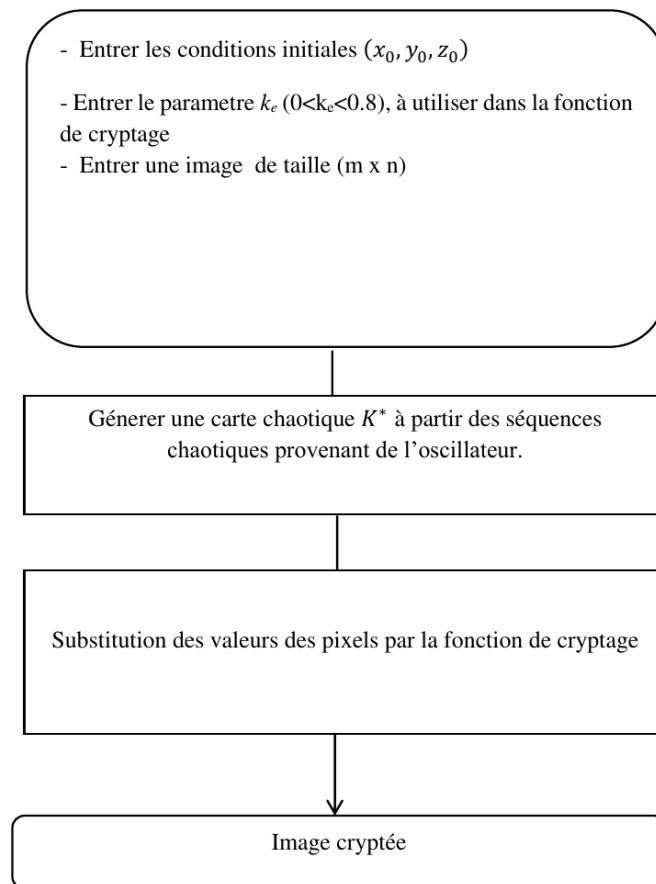


Figure 27 – Organigramme du schéma de cryptage

3.3 Analyse du cryptosystème

Certains tests cités au chapitre précédent seront effectués : analyse de la clé, analyse statistique, analyse différentielle. Ces tests seront explicités plus en détail dans la section 3.5. La carte chaotique est extraite de l'oscillateur de Colpitts.

3.3.1 Analyse de la clef secrète

La clef secrète est constitué de 4 nombres réels, trois conditions initiales et d'un paramètre à utiliser dans la fonction de cryptage. Pour une considération de 15 chiffres après

la virgule, les nombres étant convertis en binaire avec une double précision (IEEE, 2008), chaque paramètre sera stocké dans des cases mémoires de 64 bits ainsi notre clé aura une longueur de 4×64 bits soit un total de 256 bits ainsi l'espace des clés sera évalué à $2^{256} \approx 1,16 \times 10^{77}$ possibilités, ce qui implique que l'espace des clés possède un niveau de sécurité confortable pour résister à une attaque brute.

3.3.2 Analyse de la sensibilité de la clé

Le but ici est de montrer l'influence de chacun des paramètres utilisés dans la clé, la plus petite différence entre deux clés choisies pour crypter une même image va produire deux images totalement différentes après passage dans le cryptosystème, même résultat lors de la tentative de décryptage d'une image par deux clés relativement proches. Les résultats sont obtenus dans les figures 28 et 29. Comme image test, nous utilisons une image de couleur (figure 29a), nous fixons la marge de variation de chacun des paramètres impliqué dans la clé à 10^{-15} (pour les nombres réels).

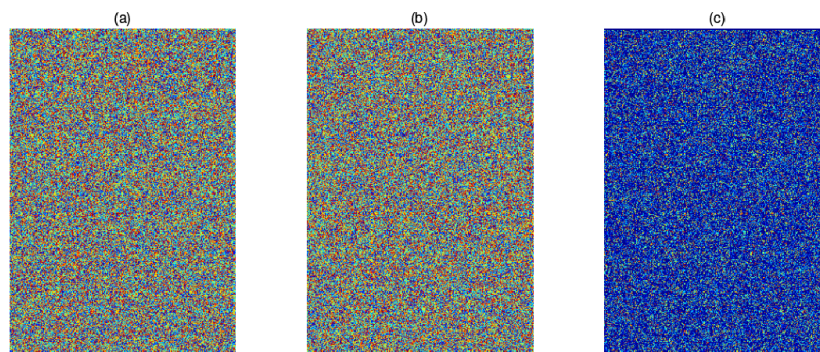


Figure 28 – Cryptage de l'image Lena avec la plus petite variation paramétrique 10^{-15} pour une conversion double précision. (a) image cryptée de Lena pour un paramètre de la clé $x_0 = 0.12345678912346$, (b) image cryptée de Lena pour un paramètre de la clé $x_0 = 0.12345678912345$

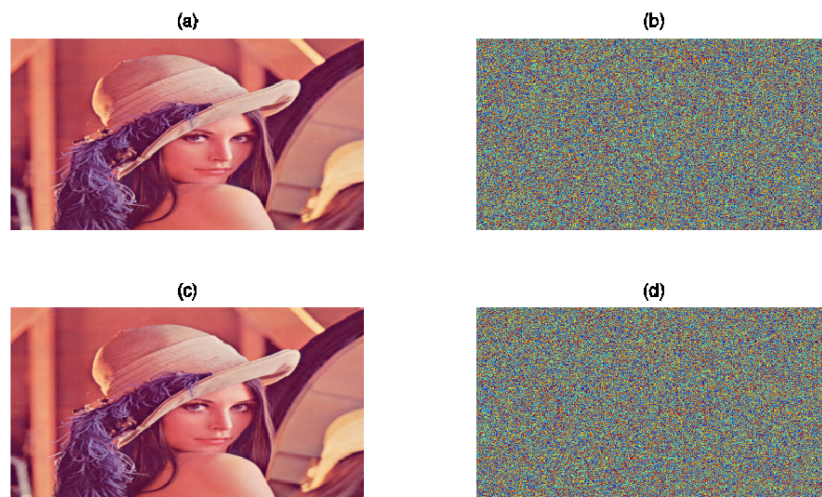


Figure 29 – Décryptage de l’image cryptée par deux clés relativement proches. (a) l’image de Lena, (b) image cryptée de Lena utilisant le paramètre de la clef $x_0 = 0,12345678912346$, (c) décryptage de l’image (a) en utilisant le paramètre $x_0 = 0,12345678912345$, (d) décryptage de (a)

3.3.3 Analyse différentielle

Les caractéristiques statistiques de l’image (image de Lena) sont : la taille (512×512), les différents coefficients de corrélation : horizontale, verticale, diagonale et anti diagonale ainsi que l’entropie H . Les coefficients, $NPCR$ et l’ $UACI$ sont calculés, les résultats obtenus sont inscrits dans le tableau 2.

Tableau 2 – Calcul du $NPCR$ et de l’ $UACI$ pour chaque séquence de données issues des oscillateurs de Colpitts et de Duffing

Métriques	Colpitts	Duffing
$NPCR$	99,4562	99,4251
$UACI$	32,7635	33,2762

3.3.4 Analyse de la vitesse

Pour le cryptage d’une image de taille 512×512 (Image de Lena), il faut en moyenne 9,77 millisecondes, satisfaisant pour le cryptage d’images en temps réel.

3.3.5 Histogramme des images

La figure 30 présente l’histogramme de l’image originale et de l’image cryptée de Lena.

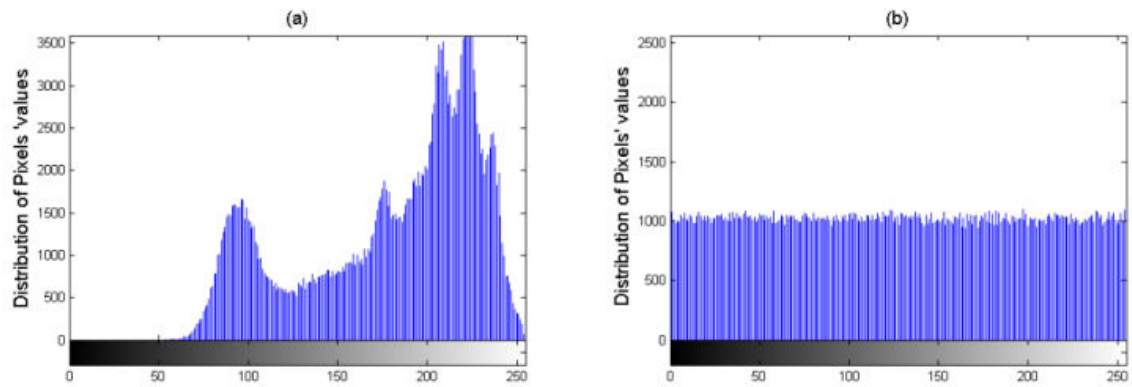


Figure 30 – Histogrammes. (a) Histogramme de l'image originale de Lena, (b) Histogramme de l'image cryptée de Lena.

Le calcul de la variance des histogrammes est effectué par la formule (3.13) les résultats sont stockés dans les tableaux 3 et 4.

Tableau 3 – Variances des histogrammes de l'image originale et de l'image cryptée de Lena

Cartes chaotiques	Image originale	Image cryptée
-	$1,018 \times 10^6$	-
Colpitts	-	914,6
Duffing	-	20004,8

Tableau 4 – Variances des histogrammes des images cryptées pour deux clés légèrement différentes

Cartes chaotiques	Clef 1	Clef 2
Colpitts	914,5	1005,5
Duffing	20004,8	19992,3

3.3.6 Analyse par corrélation

Nous sélectionnons 8000 paires de pixels adjacents dans cette image. Le résultat visualisé dans la figure 31 représente une comparaison entre les corrélations des pixels diagonales de l'image originale et de l'image cryptée. Les formules (3.14 et 3.15) sont utilisées pour calculer le coefficient de corrélation de l'image originale et de l'image cryptée. Le tableau 5 arbore les différentes valeurs.

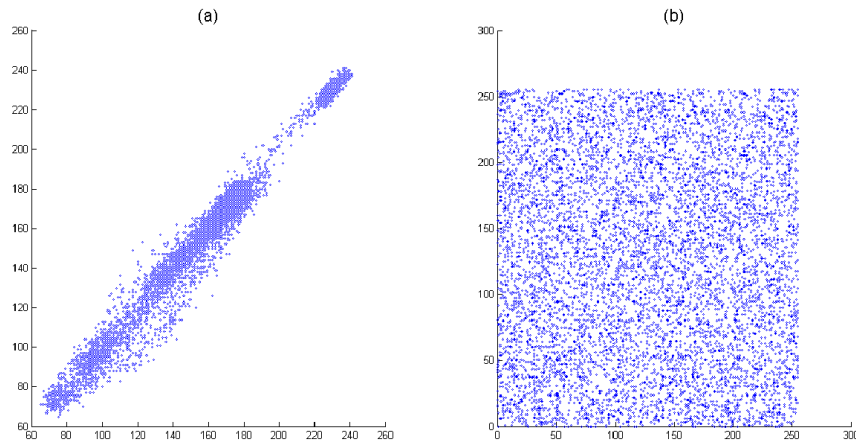


Figure 31 – Corrélation diagonale des pixels adjacents. (a) corrélation de l'image de Lena, (b) corrélation de l'image cryptée de l'image Lena

Tableau 5 – Coefficient de corrélation en fonction de chaque oscillateur. CH = corrélation horizontale, CV = corrélation verticale, CD = corrélation diagonale

Métriques	Image originale de Lena	Image cryptée de Lena	
Coefficient de corrélation	-	Colpitts	Duffing
CH	0,972	0,0014	0,0018
CV	0,985	0,002	0,0014
CD	0,9596	0,00017	0,0012

3.4 Cryptage par mixage des deux cartes

Les conditions initiales de chacun des deux oscillateurs ainsi que certains paramètres de la fonction de cryptage qu'on présentera plus loin, vont constituer la clef secrète du système de chiffrement et de déchiffrement. L'idée consiste à fabriquer une carte chaotique suffisamment robuste à partir de deux cartes chaotiques mentionnées plus haut, cette carte sera ensuite associée à une image de taille $M \times N$ dans un processus de cryptage qui utilisera une fonction de chiffrement. La description schématique du système de cryptage proposé est représentée à la figure 31. L'image chiffrée envoyée vers un algorithme de déchiffrement utilisant une séquence chaotique identique à celle utilisée par l'algorithme de cryptage, permettra de restaurer l'image originale.

3.4.1 Technique de combinaison pour la génération de la carte chaotique

Après avoir calculé les couples tridimensionnelles itérés (x_1^i, y_1^i, z_1^i) , et (x_2^i, y_2^i, z_2^i) des deux oscillateurs, où i représente le nombre d'itérations, pour une image originale O de taille $M \times N$ la combinaison des deux oscillateurs sera stockée dans une matrice K_{ij} qui est calculée par le pseudo code suivant :

Pour i allant de 1 à $M \times N$

Faire

$$K^*(i) \leftarrow x_1^i + y_1^i + z_2^i + x_2^i$$

Fin pour

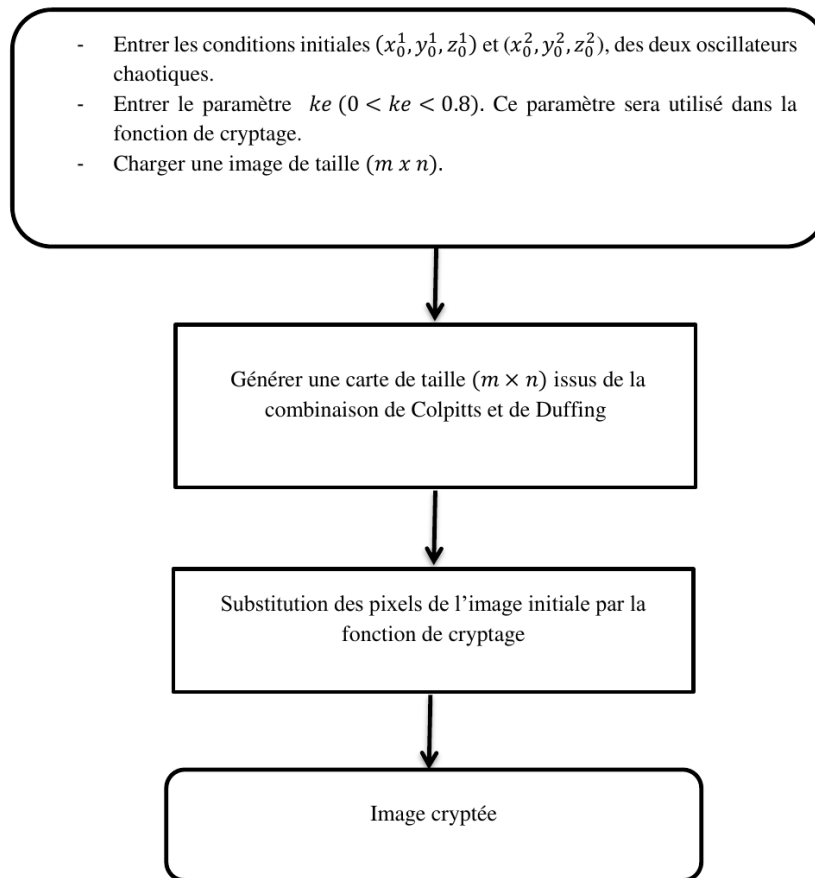


Figure 32 – Organigramme du système de cryptage proposé

3.4.2 Fonction de transformation ou de brouillage

Comme fonction de brouillage, nous utilisons une fonction mathématique linéaire utilisant la matrice K_{ij} , l'image initiale de départ ainsi que le paramètre précédemment déclaré comme variable, la formule utilisée est présentée à l'équation (3.12).

$$T_{ij} = \text{mod}(256)[k_e \times (O_{ij}) + (1 + k_e) \times (10^{12}) \times \text{round}(K_{ij})\text{mod}(2^8)] \quad (3.12)$$

Où O_{ij} représente l'image originale et T_{ij} représente l'image brouillée, k_e un paramètre à valeurs dans $]0;0,8[$.

Les fonctions *round* et *mod* sont des routines fonctionnant sous le logiciel Matlab, la première est une fonction d'arrondi, et la seconde réalise la fonction modulo afin d'obtenir des valeurs comprise entre 0 et 256 dans le cas d'une image comportant 256 niveaux de gris (Andrews et Pratt, 1963).

3.5 Analyses de la sécurité

Dans le but de tester le cryptosystème proposé, nous allons utiliser quatre images tests (Figure 33) de caractéristiques diverses : des images en noir et blanc (niveaux de gris), une image couleur (RGB), des images carrés de taille $(n \times n)$, des images de tailles différentes $(m \times n)$. Ces images sont les plus utilisées dans la littérature pour tester des cryptosystèmes (Chong et al., 2011 ; Eyebe et al., 2014 ; Zhang et al., 2015).

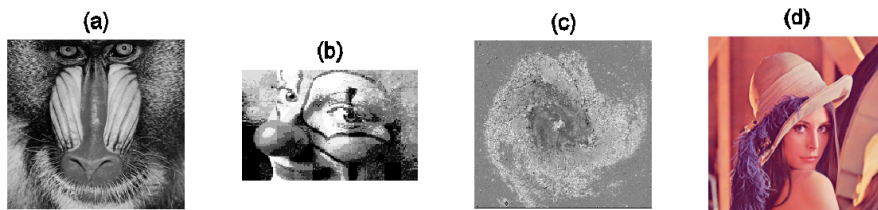


Figure 33 – Images tests. (a) Mandrill, (b) Clown, (c) Nebula m83, (d) Lena

Les différents tests cités au chapitre précédent seront effectués : Analyse de la clé, analyse statistique, analyse différentielle puis suivra une cryptanalyse du cryptosystème proposé.

3.5.1 Analyse de la clef secrète

La clef secrète est composé de 7 nombres réels, nous utilisons des nombres réels afin d'éviter des effets d'atténuations de données causée par la discrétisation. Les valeurs déci-

males sont à virgule fixe. Elles peuvent être convertis en binaire par le standard IEEE 754 (IEEE, 2008). Ce standard offre deux types de conversion : une conversion simple précision et une conversion double précision. Nous utiliserons la double précision. En considérant 15 chiffres après la virgule, les nombres étant convertis en binaire avec double précision (IEEE, 2008), chaque paramètre sera stocké dans des cases mémoires de 64 bits. Ainsi notre clé aura une longueur de 7×64 bits soit un total de 448 bits. Il s'en suit que l'espace des clés sera évalué à $2^{448} \approx 7,26 \times 10^{134}$ possibilités. L'espace des clés possède un niveau de sécurité largement suffisant pour résister à une attaque brute, car on considère généralement, pour la puissance des calculateurs actuels qu'un nombre de combinaisons égal à 10^{75} est suffisant pour résister à une attaque brute.

3.5.2 Sensibilité de la clef

Elle est testée en utilisant l'image 33a. Nous fixons la marge de variation de chacun des paramètres impliqués dans la clef à 10^{-15} (pour les nombres réels), les résultats sont obtenus dans les figures 34 et 35.

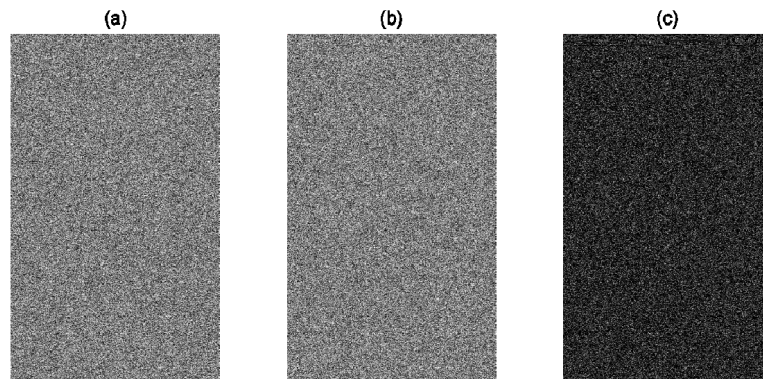


Figure 34 – Cryptage de l'image Mandrill avec la plus petite variation paramétrique 10^{-15} pour une conversion double précision. (a) image cryptée de Mandrill pour un paramètre de la clé $x_0^1 = .1234567891236$, (b) image cryptée de Mandrill pour un paramètre de la clé $x_0^1 = .1234567891237$, (c) Différence entre les deux images cryptées (a) et (b)

D'après les résultats obtenus le cryptosystème est extrêmement sensible à la plus petite modification de la clé (Shannon, 1949).

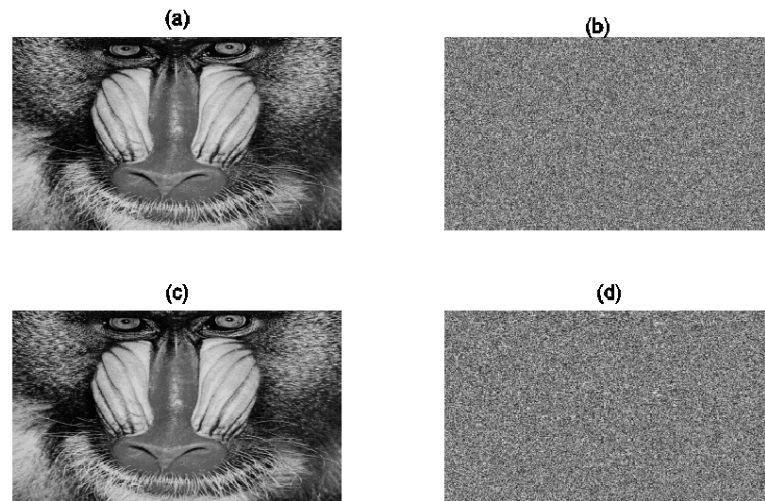


Figure 35 – Décryptage de l’image cryptée par deux clés relativement proches. (a) image de Mandrill, (b) image cryptée de Mandrill utilisant le paramètre de la clef $x_0^1 = .1234567891236$, (c) décryptage de l’image (a) en utilisant le paramètre $x_0^1 = .1234567891236$, (d) décryptage de l’image (a) en utilisant le paramètre de la clé augmenté de 10^{-15} , $x_0^1 = .1234567891237$

3.5.3 Analyse statistique

Pour les différents tests statistiques nous allons répertorier les caractéristiques intrinsèques des différentes images utilisées dans le tableau 6.

Tableau 6 – Propriétés statistiques des images. CH = corrélation horizontale, CV = corrélation verticale, CD = corrélation diagonale et H = entropie

Image	Size	CH	CV	CD	H
Lena	512 × 512	0,972	0,985	0,959	7,45
Mandrill	480 × 500	0,933	0,912	0,866	7,21
Clown	200 × 320	0,904	0,917	0,901	5,12
Nebula m83	400 × 378	0,459	0,487	0,369	4,90

3.5.4 Analyse par histogramme

Le test effectué ici opère en deux phases :

- Le tracé des histogrammes. Les tracés sont réalisés sur les figures 36 et 37 ;
- L’analyse de la variance des histogrammes (Zhang et Wang, 2014) des images originales et des images cryptées.

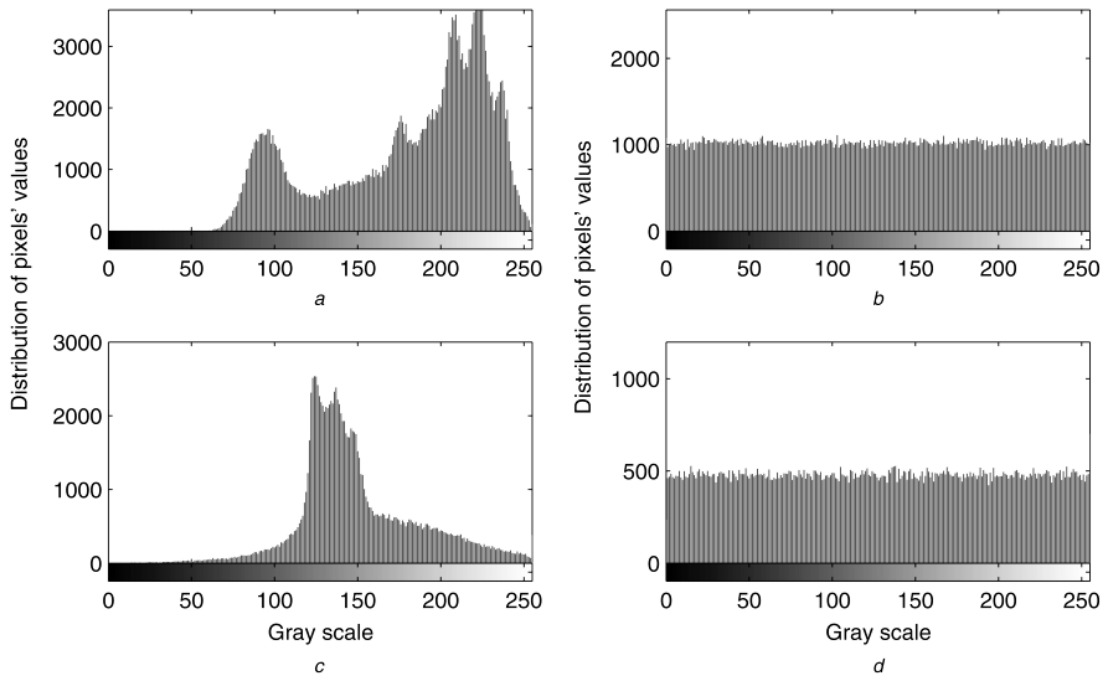


Figure 36 – Histogrammes des images. (a) Lena, (b) image cryptée de Lena, (c) Nebula m83, (d) image cryptée de Nebula m83

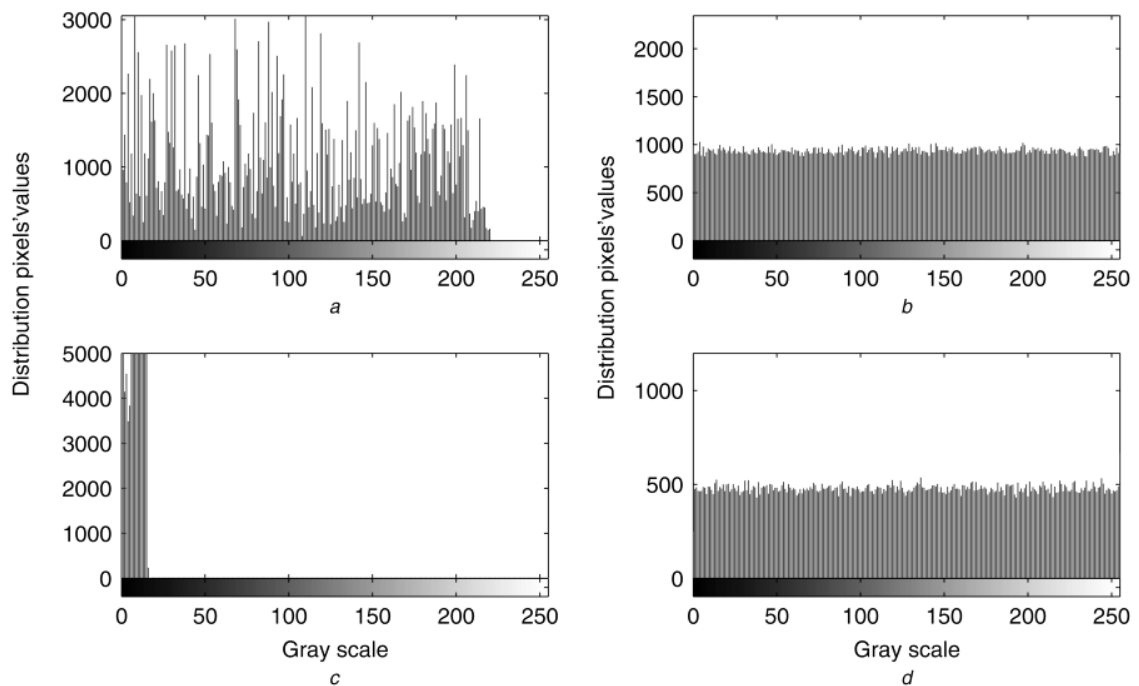


Figure 37 – Histogrammes des images. (a) Mandrill, (b) image cryptée de Mandrill, (c) Clown, (d) image cryptée de Clown

Dans la seconde hypothèse, De récents travaux (Zhang et Wang, 2014) révèlent qu'il est nécessaire de calculer la variance des histogrammes de l'image à crypter et de l'image

cryptée, mais aussi de calculer les variances des histogrammes de plusieurs images cryptées afin de s'assurer de l'uniformité des histogrammes des images cryptées. Ainsi après calcul on doit vérifier deux aspects :

1. Une faible valeur de la variance de l'image cryptée est synonyme de grande uniformité on peut l'observer dans le tableau 7 ;
2. Une modification légère d'un paramètre de la clé, entraîne une variation faible de la valeur des variances des images cryptées. On peut l'observer dans le tableau 8.

La formule utilisée pour calculer la variance des histogrammes des différentes images est donnée par l'équation 3.13.

$$Var(Z) = \frac{1}{n^2} \sum_{i=1}^n \sum_{j=1}^n \frac{1}{2} (z_i - z_j)^2 \quad (3.13)$$

$Z = (z_1, z_2, \dots, z_{256})$ pour une image représentée sur 256 niveaux de gris, Z est le vecteur de valeurs. z_i et z_j représentent le nombre de pixels dont les niveaux de gris aux indices i et j sont égales.

Tableau 7 – Variances des histogrammes des images en clair et des images chiffrées

Image	Image en clair	Image chiffrée
Mandrill	$6,13 \times 10^5$	971
Lena	$1,018 \times 10^6$	1077
Clown	$6,89 \times 10^5$	236
Nebula m83	$8,45 \times 10^6$	515

Tableau 8 – Variances des histogrammes des images cryptées pour deux clés légèrement différentes

Image	clé1	clé2
Mandrill	971	999
Lena	1077	996
Clown	236	269
Nebula m83	515	558

Le résultat obtenu montre que la distribution des pixels des images chiffrées est totalement uniforme contrairement à la distribution des pixels des images originales. Ceci peut être interprété par l'efficacité de la fonction de transformation dans l'algorithme de cryptage.

3.5.5 Analyse par corrélation

Nous avons également réalisé un test des propriétés de corrélations entre les pixels des différentes images originales et leurs images chiffrées. Nous sélectionnons 8000 paires de pixels adjacents dans chaque image. Cette sélection peut se faire de manière horizontale, verticale, diagonale et anti diagonale. Le résultat visualisé dans les figures 38 et 39 représente une comparaison entre les corrélations des pixels anti diagonales des différentes images. Les formules utilisées pour calculer le coefficient de corrélation des différentes images sont affichées ci-dessous :

$$r_{xy} = \frac{cov(x, y)}{\sqrt{Var(x)}\sqrt{Var(y)}} \quad (3.14)$$

$$cov(x, y) = \frac{1}{N} \sum_{i=1}^N [x_i - E(x)][y_i - E(y)] \quad (3.15)$$

avec,

$$E(x) = \frac{1}{N} \sum_{i=1}^N (x_i) \quad et \quad Var(x) = \frac{1}{N} \sum_{i=1}^N [x_i - E(x)]^2$$

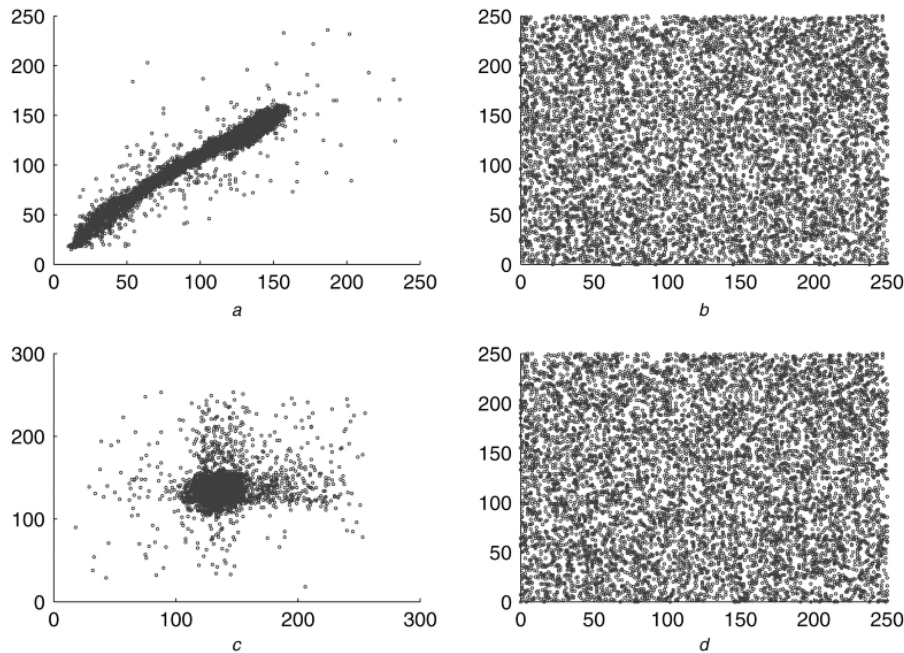


Figure 38 – Corrélation anti diagonale des pixels adjacents. (a) corrélation de l'image de Lena, (b) corrélation de l'image cryptée de l'image Lena, (c) corrélation de l'image Nebula m83, (d) image cryptée de l'image Nebula m83

Afin de vérifier toutes les orientations des pixels de l'image, nous calculons et nous présentons les valeurs des différents coefficients de corrélation (verticale, horizontal et diagonal)

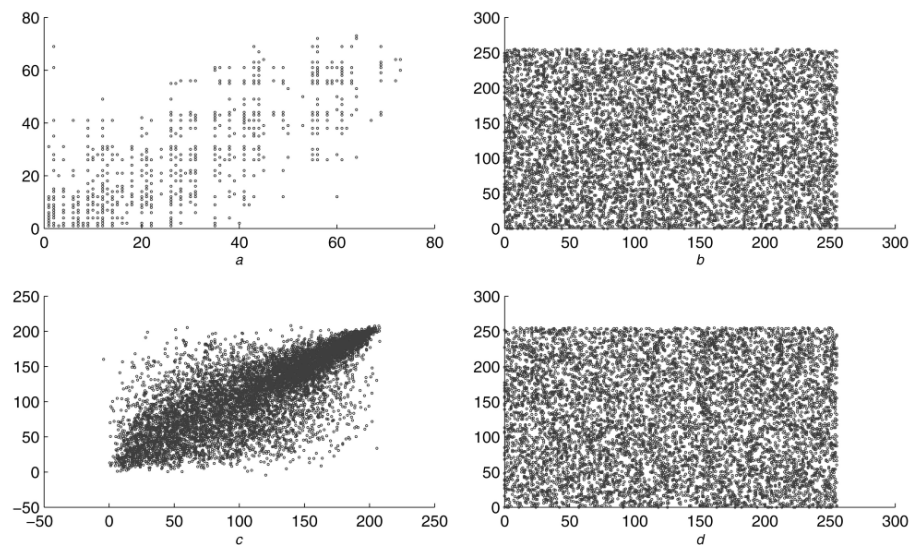


Figure 39 – Corrélation anti diagonale des pixels adjacents. (a) corrélation de l’image Mandrill, (b) corrélation de l’image cryptée de l’image Mandrill, (c) corrélation de l’image Clown, (d) corrélation de l’image cryptée de l’image Clown

de l’image de Mandrill, les résultats sont stockées dans le tableau 9. Le résultat est satisfaisant car on constate que la corrélation entre les pixels est très faible, et que la représentation des pixels adjacents expose une texture extrêmement désordonnée dans les images cryptées contrairement aux images initiales. Ces résultats sont confirmés par le calcul des valeurs de l’entropie H . Ce paramètre qui exprime le degré de désordre dans un système. Les résultats sont présentés dans le tableau 10.

Tableau 9 – Coefficient de corrélation des images. CH = corrélation horizontale, CV = corrélation verticale, CD = corrélation diagonale

	Lena	Mandrill	Clown	Nebula m83	Image chiffrée Lena	Image chiffrée Mandrill	Image chiffrée Clown	Image chiffrée Nebula m83	Valeur moyenne
CH	0,972	0,933	0,904	0,459	0,00145	-0,0015	0,0015	-0,0021	0,0033
VC	0,985	0,912	0,912	0,487	0,00237	-0,0012	-0,0032	0,00374	0,0056
CD	0,9596	0,866	0,866	0,369	0,00022	0,00035	-0,00078	-0,0013	0,0016

Tableau 10 – Entropie des images cryptées

	Lena	Mandrill	Clown	Nebula m83
H	7.996	7.999	7.997	7.998

3.5.6 Analyse différentielle

Notons que les coefficients *NPCR* qui permet d'évaluer le taux de changement des pixels dans l'image et l'*UACI* qui permet de quantifier la sensibilité par rapport au texte clair peuvent être utilisés de deux manières : Soit pour tester la sensibilité par rapport au texte clair en modifiant un seul pixel dans l'image originale et en analysant l'effet sur l'image chiffrée, soit pour tester la sensibilité à la clé secrète, mais au lieu de changer un pixel dans l'image originale, on change légèrement la clef secrète. Nous avons effectué la seconde méthode et les résultats obtenus sont inscrits dans le tableau 11.

Tableau 11 – Sensibilité à l'attaque différentielle par le calcul du NPCR et de l'UACI.

	Lena	Mandrill	Clown	Nebula m83
NPCR	99,5771	99,4712	99,384	99,3215
UACI	35,082	32,9123	32,3156	27,9054

Les valeurs de références (Patidar et al., 2009) sont fixées à 99,6 % pour le *NPCR* et 33,4 % pour l'*UACI* pour une image de taille 512×512 codée sur 8 bits (Image Lena). En comparaison avec notre cryptosystème, nos résultats sont très proches des valeurs de références.

3.5.7 Analyse de la vitesse

Notre avons développé ce cryptosystème dans l'environnement Matlab, pour cela nous avons utilisé un PC SAMSUNG Core Duo de 2.0 GHz de processeur et 2Go de mémoire, avec comme système d'exploitation, Windows 7. Pour le cryptage d'une image de taille 512×512 (Image Lena), il faut en moyenne 10 millisecondes. Pour le cryptage d'une image de taille inférieure, 400×380 (image Nebula m83) il faut en moyenne 6 millisecondes, ces durées sont suffisamment faibles pour envisager un cryptage en temps réel avec notre système.

3.5.8 Analyse du système par Cryptanalyse

Dans cette section, nous allons effectuer une analyse du cryptosystème proposé en utilisant une attaque en test clair choisi, la CPA (*chosen plaintext attack*) et une attaque en texte chiffrée choisi, la CCA (*chosen ciphertext attack*).

3.5.8.1 Attaque en texte clair choisi (CPA)

C'est un modèle d'attaque dans lequel un adversaire estime pouvoir venir à bout du cryptosystème dans la mesure où il peut en extraire la clé. Nous supposons dans un premier temps que l'adversaire possède une image cryptée provenant du cryptosystème. Il a totalement accès à la carte chaotique utilisé, mais il ne connaît pas le groupe de paramètres $[(x_0^1, y_0^1, z_0^1, x_0^2, y_0^2, z_0^2), k_e]$ utilisé pour générer les différentes valeurs. La particularité de cette attaque repose sur la faiblesse de l'algorithme dans le but de révéler la clé secrète (Cheddad et al., 2010). L'adversaire utilise une image extrême ou une matrice vide c'est-à-dire qu'il construit une matrice dont toutes les valeurs sont à 0 ou à 255. Il utilise donc l'algorithme de cryptage afin d'extraire la matrice utilisée pour crypter l'image qu'il possède. Une fois extraite l'image cryptée peut être décryptée en utilisant l'équation (3.16). Cette attaque a été développée par Cheddad et al. (2010), nous l'utilisons afin de tester notre cryptosystème.

$$A = A' \oplus (B' \oplus Map) \quad (3.16)$$

Avec A l'image décryptée, A' l'image cryptée de A , B' l'image cryptée obtenue à partir de l'image extrême, \oplus est la fonction de cryptage et Map est la carte présumée.

La procédure est effectuée en utilisant deux images tests de la figure 40 et les résultats sont observés dans la figure 41.



Figure 40 – Images tests. (a) Image de Barbara, (b) image neutre

On voit bien que l'adversaire n'arrive pas à obtenir la bonne image. L'attaque est donc un échec.

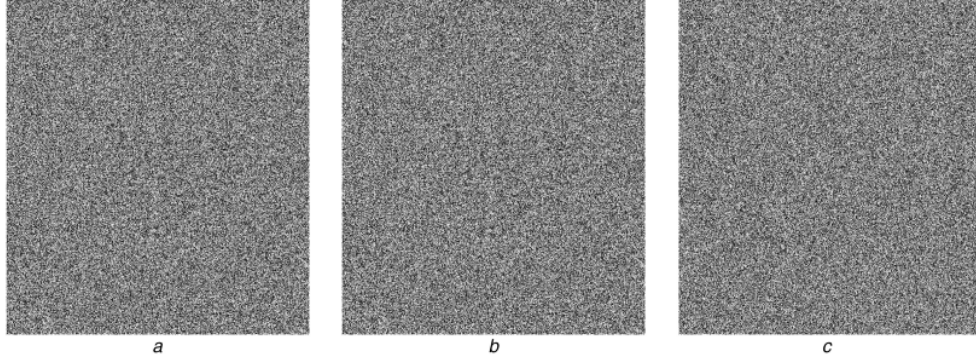


Figure 41 – Test par CPA. (a) Image cryptée de Barbara, (b) image extraite en cryptant l'image neutre, (c) décryptage de (a) en utilisant (b)

3.5.8.2 Attaque à texte chiffré choisi (CCA)

Dans ce type d'attaque, l'adversaire possède des images cryptées. Il a totalement accès à l'algorithme de décryptage, et a également accès à la matrice de cryptage/décryptage présumée, mais il ne connaît pas les paramètres utilisés pour générer la carte chaotique. En fonction de ces conditions, l'adversaire veut pouvoir décrypter une image cryptée C , pour cela on choisit une image extrême D qu'on utilise dans la fonction de décryptage \oplus suivant l'équation (3.17) :

$$B = Map \oplus D \quad (3.17)$$

Après avoir obtenu l'image intermédiaire B , l'image décryptée de C est obtenue par l'équation (3.18) :

$$A = C \oplus B \quad (3.18)$$

A est l'image décryptée, C l'image à crypter et B est l'image décryptée de l'image extrême D . Les images tests sont présentées dans la figure 42 et les résultats sont présentés dans la figure 43.

De nouveau, on se rend compte que l'attaque est un échec.

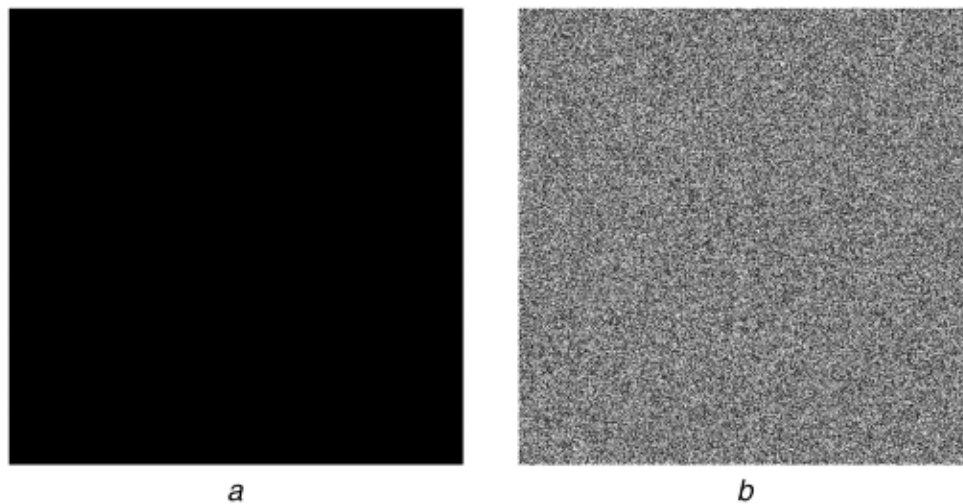


Figure 42 – Images tests. (a) image neutre, (b) image cryptée de Barbara

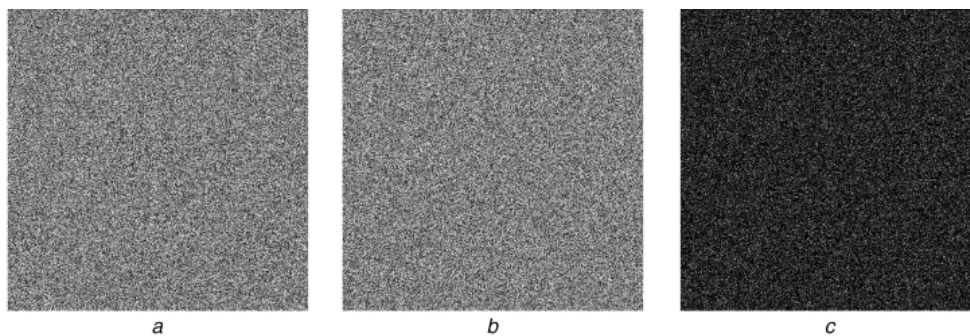


Figure 43 – Test CCA. (a) Image cryptée de Barbara, (b) image décryptée en utilisant l'image décryptée de l'image neutre, (c) différence entre (a) et (b)

3.5.9 Etude comparative des cryptosystèmes

3.5.9.1 Comparaison entre les cryptosystèmes, celui utilisant une carte, et l'autre utilisant deux cartes chaotiques

Pour présenter l'influence des cartes utilisées sur les cryptosystèmes, nous observons les paramètres statistiques obtenues à partir d'une image chiffrée. Nous utilisons l'image de Lena comme image de référence. Le tableau 12 répertorie les paramètres testés.

Le mixage de cartes nous assure un espace de clés plus large et de meilleurs paramètres statistiques.

Tableau 12 – Comparaison entre les cryptosystèmes. EC : Espace de la Clé, TR : Temps Requis (en millisecondes), VH : Variance des Histogrammes, CH : Corrélacion Horizontale, CV : Corrélacion Verticale, CD : Corrélacion Diagonale

	EC	NPCR	UACI	TR (ms)	VH	CH	CV	CD
Colpitts	$1,66 \times 10^{77}$	99,45	32,76	9,77	914,6	0,014	0,02	0,00017
Duffing	$1,66 \times 10^{77}$	99,42	33,57	9,77	20004,8	0,0018	0,0014	0,0012
Mixage Colpitts/ Duffing	$7,26 \times 10^{134}$	99,57	35,08	10	1077	0,00145	0,0023	0,00022

3.5.9.2 Comparaison avec d'autres méthodes

Depuis plusieurs décennies de nombreux cryptosystèmes basés sur le Chaos sont développés dans la littérature avec plus ou moins du succès, pour comparer notre procédé de cryptage à celles présentent dans la littérature, nous nous servons de quelques métriques calculées plus haut (*NPCR*, *UACI*, variance, entropie *H*, etc.) ainsi que d'autres paramètres tels que le temps requis pour le cryptage d'une image, le nombres de rondes effectuées, l'espace de la clef. Nous présentons un tableau comparatif de notre méthode à une liste de méthodes récentes rencontrées dans nos recherches, nous utilisons comme image test, l'image de Lena de taille 512×512 car elle est très utilisée dans la littérature. Le tableau 13 fait un résumé complet de cette comparaison. Comparaison des méthodes de cryptage utilisant l'image de Lena comme image test.

Tableau 13 – Comparaison des méthodes de cryptage utilisant l'image de Lena comme image test. EC : Espace de la Clé, CM : Corrélacion Moyenne, NTD : Nombre de Tours de Diffusion, TR : Temps Requis (en millisecondes), VH : Variances des Histogrammes, H : Entropie

Algorithme	NPCR (%)	UACI (%)	NTD	TR (ms)	VH	EC	H	CM
Schema proposé	99,57	35,08	1	9,77	1077	$7,2 \times 10^{134}$	7,996	0,0024
Eyebe et al., (2014)	99,62	33,40	1	210	-	$2,2 \times 10^{57}$	7,999	0,0047
Xiao et al., (2009)	99,60	33,30	2	-	-	-	-	-
Ahmed et al., (2013)	99,60	33,46	-	-	-	-	7,999	-
Li et al., (2007)	99,66	33,42	3	-	-	10^{120}	-	0,0025
Radu et al., (2014)	99,24	33,24	2	480	-	$3,1 \times 10^{144}$	7,902	0,0071
Zhang et Wang, (2014)	99,64	33,44	3	661,5	5335,8	10^{120}	-	0,0014

Notre méthode semble proposer de bonnes qualités par rapport à certaines et semble

bien adapté aux cryptages de données en temps réel.

3.6 Fusion de cartes

3.6.1 Algorithme de cryptage

Dans cet algorithme on procède en trois étapes ; on fusionne d'abord les deux oscillateurs pour produire une nouvelle carte chaotique puis on teste la régularité de cette nouvelle carte enfin on l'utilise dans le schéma de cryptage.

3.6.1.1 Technique de fusion

Les variables d'état du premier système (x_1, y_1, z_1) seront fusionnées aux variables d'état du second système (x_2, y_2, z_2) . Après i itérations chaque système génère des groupes de variables (x_{1i}, y_{1i}, z_{1i}) et (x_{2i}, y_{2i}, z_{2i}) qui seront ensuite utilisés dans un code pour générer la matrice K_{ij} comme suit :

Tant que $u \leq M \times N$

Pour i allant de 1 à M

Faire

Pour j allant de 1 à N

Faire

$$K_{ij} = x_{1u} + y_{1u} + z_{1u} + x_{2u}$$

Fin pour

Fin pour

$u = u + 1$

3.6.1.2 Etude de la carte chaotique obtenue

Malgré le fait que la série de données obtenue, provient de deux systèmes chaotiques, il est nécessaire de faire subir des tests à cette séquence, afin de s'assurer de la présence du chaos dans ses données. Quelques méthodes telles que le test binaire 0-1 (Gottwald et Melbourne, 2004 ; Gopal et al., 2013), la permutation d'entropie (Bandt et Pompe, 2002 ; Yinhe et al., 2004) ont été développées. Ces techniques présentent l'avantage, d'être applicables directement sur les séries de données temporelles sans nécessairement connaître le système d'où proviennent ces données, ce qui élimine la reconstitution du système de phase. Elles peuvent être utilisées sur des données provenant des systèmes analogiques ou numériques (Gopal et al., 2013 ; Eyebe et al., 2013 ; Eyebe et Koepf, 2015). Elles sont robustes au bruit,

l'étude d'un échantillon de données peut être étendue sur le système tout entier, et la nature des données est définie par des indicateurs de chaos. Parmi ces indicateurs, l'entropie est un paramètre très utile pour caractériser ou quantifier la complexité des séries de données. Il permet également de distinguer le caractère régulier (périodique ou non), aléatoire, ou chaotique de ces données. Récemment un nouveau test a été proposé par Eyebe et al. (2014) ; Eyebe (2017). Cette technique basée sur la permutation d'entropie, utilise les lignes de plus grandes pentes (PSLE) pour caractériser les séries de données. Ce nouveau test présente l'avantage d'être plus rapide que ceux cités précédemment, et donc plus adapté aux applications en temps réel. C'est ce qui explique l'utilisation de cette méthode.

3.6.1.3 Description de la méthode PLSE

Soit une série de données $[X_t]_{t=1,\dots,T}$ de taille T où t représente l'index des valeurs, la permutation d'entropie d'ordre n définit la mesure des probabilités de permutations d'ordre n . Les permutations d'ordre n sont obtenues par la comparaison des valeurs voisines dans un échantillon de données prélevé dans la série de données. Cette comparaison se fait par ordre croissant et est stocké dans le nouveau vecteur de données $X'_t = [X_{t+t_0}, X_{t+t_0+\tau}, \dots, X_{t+t_0+(l-1)\tau}]$ où l représente le nombre de valeurs de X'_t , t_0 l'index de la valeur initiale du vecteur X'_t et τ la distance entre deux valeurs du vecteur X'_t . En admettant que la permutation d'ordre n de la série X_t est une fonction linéaire par morceaux, on considère que la pente de chaque fonction linéaire s est la différence entre les différents paires de permutations voisines P_t .

$$s_i = P_t(i+1) - P_t(i), \quad 1 \leq i \leq n-1 \quad (3.19)$$

La plus grande pente de P_t est définie par :

$$S_t = \max(s_i) \quad (3.20)$$

En considérant $L = \lim_{n \rightarrow \infty} |S_t|$, les dynamiques régulières sont caractérisées par une seule et même valeur de L pour chaque ligne de plus grande pente si la dimension n est telle que $L < n$ (Eyebe et al., 2013). La valeur normalisée de l'entropie de plus grande pente (PLSE) d'ordre $n \geq 2$ est obtenue par l'équation suivante :

$$h_s(n) = \frac{-1}{\ln(n-1)} \sum p(s) \ln[p(s)] \quad (3.21)$$

Avec,

$$p(s) = \frac{Nb[t|t \leq T-n, S_t = s]}{T-N+1} \quad (3.22)$$

$P(s)$ est la probabilité d'apparition d'une pente s , et Nb est le nombre d'apparitions des pentes.

Pour des dynamiques régulières, $h_s(n) = 0$, avec une période $L < n$.

Pour des dynamiques non régulières $0 < h_s(n) < 1$.

Pour des dynamiques régulières, la plus grande pente $S_t = s$ quel que soit l'échantillon prélevé, $h_s = 0$ et $P(s) = 1$.

Pour des dynamiques non régulières la valeur S_t peut prendre différentes valeurs.

Afin de tester les données issues de la fusion des cartes chaotiques nous avons appliqué l'organigramme illustré sur la figure 44.

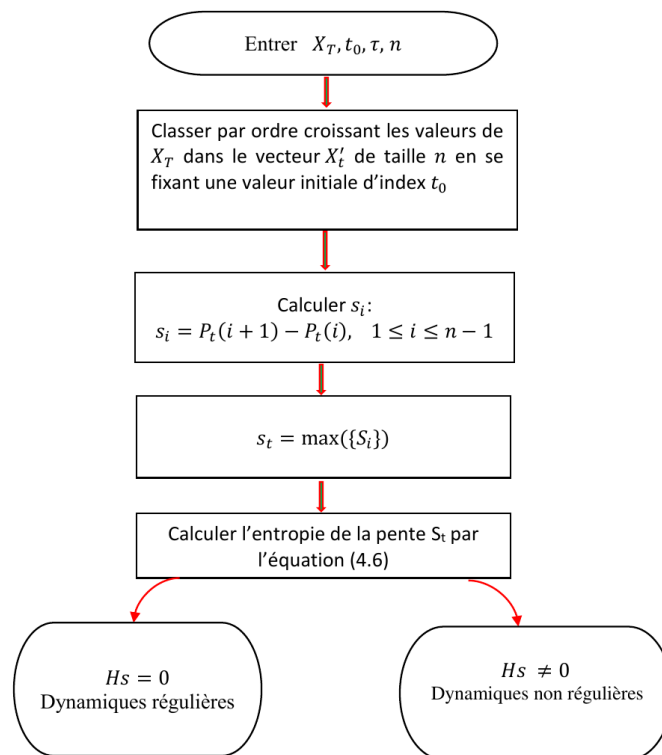


Figure 44 – Organigramme du test par la PLSE de la série de données X_t

La PLSE est appliquée sur la nouvelle série de données, nous avons retenu quelques échantillons de données dont l'entropie calculée est le maximum possible. Il est préférable de choisir $\tau \geq 1$ pour réduire la détection des erreurs (Eyebe et Koepf, 2015). En choisissant t_0 tel que $1 < t_0 < n$, cela permet de réduire le nombre d'échantillons de données à tester (Eyebe et Koepf, 2015). Les paramètres utilisés pour effectuer le test sont les suivants : $t_0 = 1, \tau = 1, T = 10000, n = 20, (x_{10}, y_{10}, z_{10}) = (0, 2; 0, 5; 0, 5), (x_{20}, y_{20}, z_{20}) = (0; 1; 1)$.

Le tableau 14 montre les différentes entropies obtenues sur les séquences de données utilisées dans l'algorithme de cryptage.

Tableau 14 – Détection de chaos par le calcul de l'entropie sur les données utilisées par la méthode de PLSE

Données	Séquence 1	Séquence 2	Séquence 3
Entropie H_s	0,3944	0,4892	0,7601

3.6.1.4 Fonction de transformation

La fonction de transformation utilisée est celle que nous avons déjà formulée à l'équation (3.12).

3.6.2 Résultats et performances du système de cryptage

Les images tests choisies sont les suivantes :

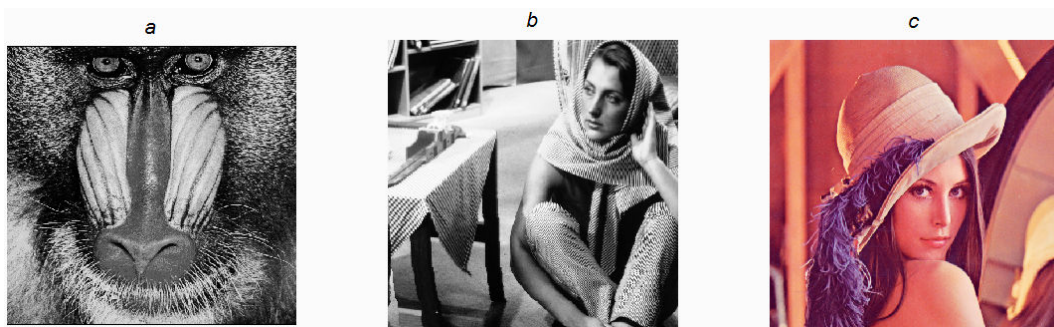


Figure 45 – Images tests. (a) Mandrill, (b) Barbara, (c) Lena

Tableau 15 – Propriétés statistiques des images. CH = corrélation horizontale, CV = corrélation verticale, CD = corrélation diagonale et H = entropie

Image	Size	CH	CV	CD	H
Lena	512 × 512	0,972	0,985	0,959	7,45
Mandrill	480 × 500	0,933	0,912	0,866	7,21
Barbara	512 × 512	0,954	0,892	0,884	7,34

La visualisation du cryptage d'une image est affichée sur la figure 46.

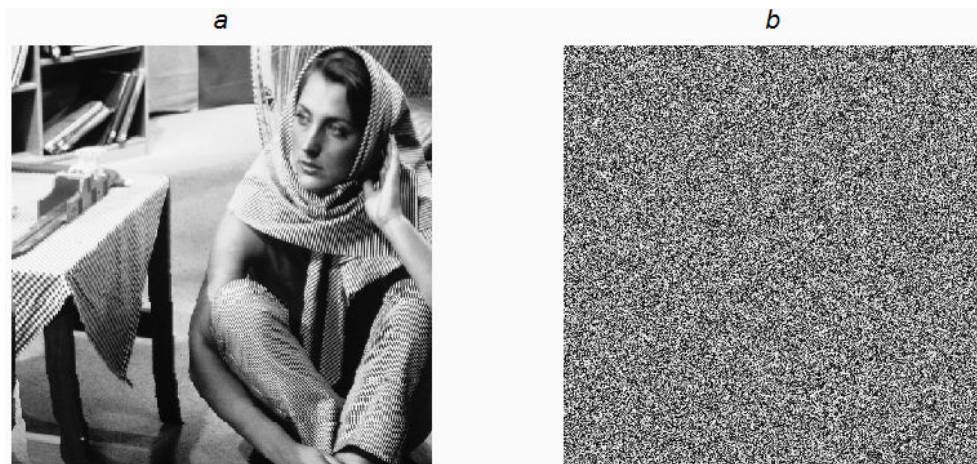


Figure 46 – Cryptage de l'image de Barbara. (a) image originale, (b) image cryptée

3.6.3 Histogrammes des images

Les histogrammes des différentes images ainsi que de leurs images cryptées sont représentées en figure 47.

Les différentes métriques utilisées pour tester l'algorithme utilisant cette nouvelle carte chaotique sont celles utilisées dans le chapitre précédent, à savoir : le NPCR, L'UACI, la variance d'histogramme, les coefficients de corrélation, ainsi que l'entropie. Les tableaux 16 et 17 résument ces différentes statistiques récoltées pour chaque image.

Tableau 16 – Propriétés statistiques des images. CH = corrélation horizontale, CV = corrélation verticale, CD = corrélation diagonale et H = entropie

	Lena	Mandrill	Barbara
NPCR	99,5771	99,4712	99,346
UACI	35,082	32,9123	33,406
H	7,996	7,995	7,997

3.6.4 Autres analyses

L'analyse de la vitesse, la taille de l'espace de la clé, ainsi que l'analyse de la corrélation font partie de ces analyses et ont été effectuées avec un certain succès. Pour un PC portable CORE DUO avec un processeur à 2.0 Ghz et une RAM à 4 Gigaoctet, le cryptage d'une image de résolution 512×512 prend en moyenne 10 ms et 9 ms pour une image de taille

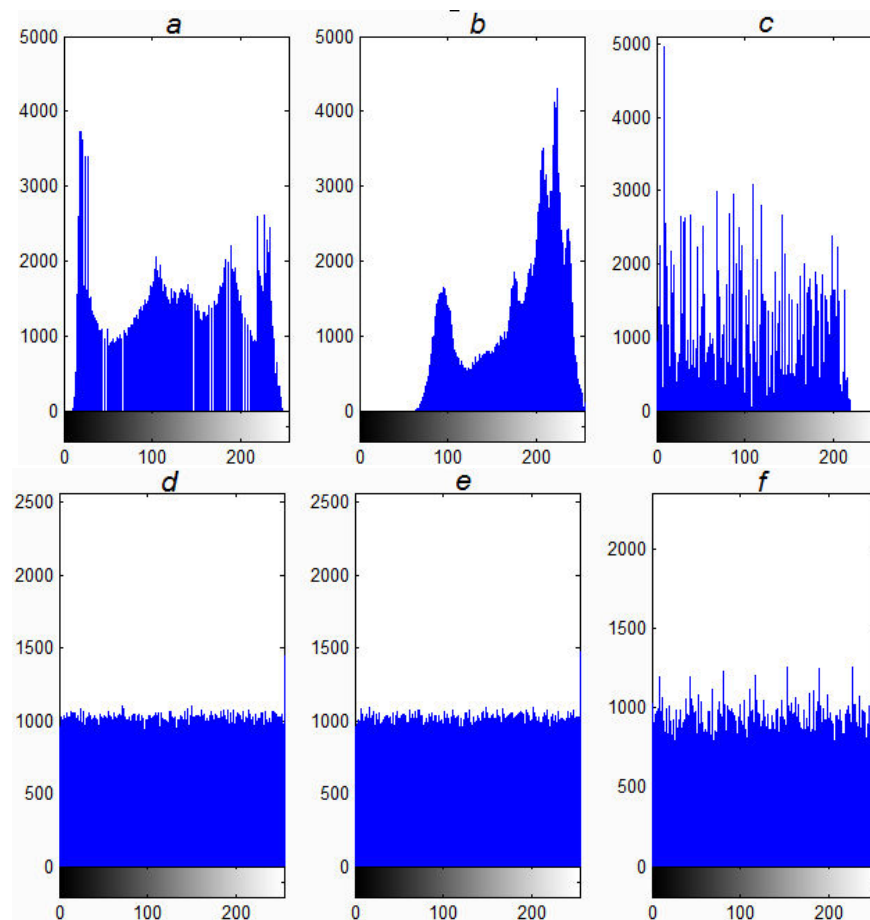


Figure 47 – Histogrammes des images. (a) Barbara, (b) Lena, (c) Mandrill, (d) image cryptée de Barbara, (e) Image cryptée de Lena, (f) Image cryptée de Mandrill

Tableau 17 – Propriétés statistiques des images. CH = corrélation horizontale, CV = corrélation verticale, CD = corrélation diagonale et H = entropie

Image	Initiale	Cryptée
Mandrill	$6,13 \times 10^5$	2080
Lena	$1,018 \times 10^6$	2216
Barbara	$7,32 \times 10^5$	2131

480×500 . La taille de la clé est de $2^{448} \approx 9,26 \times 10^{134}$, suffisamment large pour résister à une attaque par force brute, comme expliqué plus tôt.

L'analyse de la corrélation se fait sur deux pixels anti diagonales proches sur chacune des images, l'image originale et l'image cryptée. Le résultat est représentée dans la figure suivante.

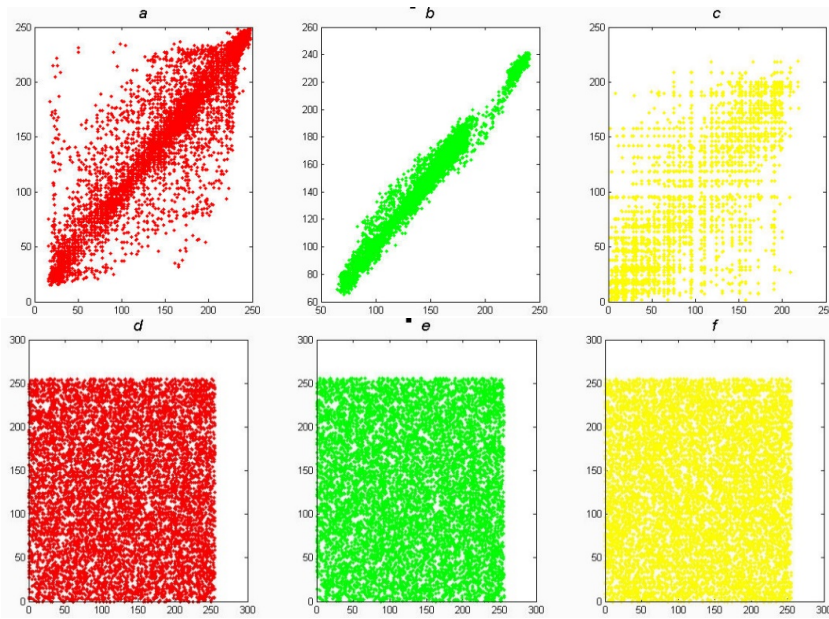


Figure 48 – Analyse par corrélation. (a) image de Barbara, (b) image de Lena, (c) image de Mandrill, (d) image cryptée de Barbara (e) image cryptée de Lena, (f) image cryptée de Mandrill

3.6.5 Cryptanalyse du cryptosystème

Pour la cryptanalyse de l’algorithme proposé, nous avons choisi d’effectuer le test par CPA (*Chosen Plain text Attack*). Nous l’avons décrit plus haut, on rappelle juste qu’une image extrême est utilisée dans le cryptosystème dans le but de révéler la clé secrète. L’équation 3.23 décrit la procédure.

$$A = A' \oplus (B' \oplus Map) \quad (3.23)$$

A représente l’image décryptée, A' son image cryptée, B' l’image cryptée de l’image extrême, le symbole \oplus représente le processus de cryptage et Map représente la clé présumée. Le résultat apparaît en figure 49.

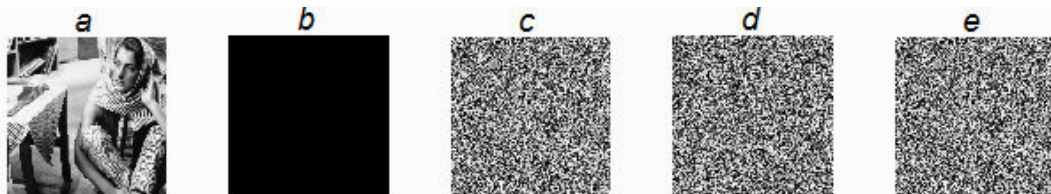


Figure 49 – Attaque par CPA. (a) image de Barbara, (b) image neutre, (c) image cryptée de Barbara, (d), image obtenue après extraction de la clé secrète, (e) décryptage de (c) en utilisant (d)

3.6.6 Comparaison avec de nouvelles méthodes

Comme effectué précédemment, nous présentons une liste comparative de quelques métriques de notre cryptosystème avec ceux de quelques cryptosystèmes récemment développés dans la littérature, l'image test choisie est celle de Lena. Le tableau 18 illustre cette comparaison.

Tableau 18 – Comparaison avec quelques cryptosystèmes. TR : Temps requis, NTD : Nombre de tours de Diffusion, CM : Corrélacion moyenne, EC : Espace de clé, H : Entropie, VH : Variance des Histogrammes

Algorithmes	TR (ms)	NTD	CM	EC	H	VH	UACI (%)	NPCR (%)
Cryptosystème proposé	9.77	1	0.0024	$7,2 \times 10^{134}$	7.996	2216	35.08	99.57
Ahmed et al., (2013)	-	-	-	-	7.999	-	33.46	99.60
Zhang et al., (2014)	661.5	3	0.0014	10^{120}	-	5335.8	33.44	99.64
Zhang et al., (2015)	-	3	0.0025	10^{120}	-	-	33.42	99.66
Xingya et al., (2016)	-	1	0.0112	10^{79}	7.997	-	33.45	99.6
Jongseo et al., (2016)	6.05	5	0.059	$7,2 \times 10^{134}$	7.819	-	31.15	99.63

Conclusion

Lors de l'élaboration de notre cryptosystème, nous avons d'abord proposé une procédure utilisant une carte chaotique, afin d'améliorer ce dernier, nous avons proposé un algorithme de chiffrement basé sur le mixage de deux cartes chaotiques. Ce mixage est utilisé ici comme générateur pseudoaléatoire de données utiles pour le cryptage d'images. Nous avons aussi présenté une nouvelle carte chaotique construite à partir de la fusion de deux autres oscillateurs à régime chaotique. Cela nous a permis de produire une nouvelle carte avec de bonnes caractéristiques de cryptage. Nous nous sommes servis d'un test de chaos développé récemment. Pour valider le choix de la carte obtenue puis intégrée dans le cryptosystème, nous avons effectué des tests statistiques qui ont été pour la plus part satisfaisants. Une fonction de cryptage est utilisée pour la diffusion des pixels de l'image. En dehors de la complexité structurelle, cet algorithme de cryptage se situe donc dans la classe de complexité de type calculatoire à l'instar du générateur pseudo aléatoire Blum blum shub. Nous avons testé ce procédé sur des images populaires telles que Lena, Mandrill et nous avons pu calculer certaines métriques d'évaluation afin de s'assurer que ce cryptosystème est efficient.

Conclusion générale et perspectives

Cette thèse traite du cryptage des images 2D en mettant au point des nouveaux algorithmes. Ces algorithmes sont bâtis autour de la fusion et du mixage des oscillateurs chaotiques et couplés à une fonction mathématique de diffusion. Ces cryptosystèmes possèdent des caractéristiques qui sont comparables avec ceux présents dans la littérature. En fonction des objectifs fixés nous avons procédé en plusieurs étapes. Nous avons pris le soin de choisir une carte chaotique dans chacun des différents oscillateurs électriques cités ci-dessus. Nous avons utilisé une fonction mathématique capable d'accentuer le brouillage dans le désordre engendré par le cryptage. L'une des contributions de cette thèse est la production de cartes de données suffisamment robustes pour résister à des attaques CPA et CCA. Pour cela, nous avons utilisé la fusion d'oscillateurs chaotiques afin d'élargir l'espace de clé de cryptage.

Impact des résultats obtenus

La quasi totalité de nos analyses ont révélé que le cryptosystème développé présentait des caractéristiques satisfaisantes en termes de sécurité. De manière quantitative, à travers l'analyse de l'entropie, le NPCR, L'UACI, il s'est avéré que les résultats obtenus par simulation numérique sur Matlab étaient aussi meilleurs ou mieux que ceux suggérées dans la littérature. Dans le chapitre 3, nous avons également montré l'avantage de l'usage de cartes fusionnées sur une carte unique pour ce faire, nous avons présenté un cryptosystème à base d'une carte chaotique issue d'un oscillateur puis nous avons extrait une carte chaotique provenant de deux oscillateurs chaotiques que nous avons aussi inséré dans notre cryptosystème. Nous avons constaté des similitudes entre les deux systèmes surtout du point de vue qualitatif, mais du point de vue quantitatif le système de cryptage utilisant deux oscillateurs chaotiques offre plus de sécurité. Après la fusion des oscillateurs chaotiques il est important de s'assurer que les séquences de données extraites de cette fusion soient chaotiques.

Perspectives

Le cryptosystème proposé dans cette thèse ne tient pas compte d'une ligne de transmission réelle (câble coaxial, fibre optique, radiofréquence), il serait nécessaire d'observer l'influence d'une chaîne de transmission (influence des bruits liés au composants, influence des bruits liés au codage canal ou au canal lui-même) sur les données à transmettre au moment du décryptage.

Le cryptage développé dans cette thèse fait partie de la famille de cryptages à clé secrète ce qui sous-entend que le contenu de la clé est gardée secrète et ne sera transmise au destinataire que de manière secrète. Il serait intéressant de développer une technique de transmission parallèle de la clé de manière simple et sécuritaire afin de ne pas utiliser le même canal de transmission. La nature dynamique des systèmes électriques à caractère chaotique ouvre un horizon vers des applications à temps réel de plus en plus variées et de plus en plus rapides, des applications telles que la transmission du son et de la vidéo, les transactions bancaires, le commerce en ligne, ceci peut également nous permettre de nous intéresser aux systèmes optoélectroniques à des fins de cryptage, car ces systèmes sont capables de produire des séquences chaotiques à des fréquences très élevées, de l'ordre de plusieurs dizaines de Gigahertz. Enfin, ce cryptosystème pourrait être implémenté et déployé dans un réseau local.

Références bibliographiques

- Abuturab, M.R.**, 2012 : Securing color information using Arnold transform in gyrator transform domain. *Optics and Lasers in Engineering*, **50**, 772–779.
- Ahmad, M., and F. Ahmad**, 2014 : Cryptanalysis of Image Encryption Based on Permutation-Substitution Using Chaotic Map and Latin Square Image Cipher. *Adv. Intell. Syst. Comput.*, **327**, 481–488.
- Andrews, H., and W. Pratt**, 1963 : Transform image coding. *Processing communication*, 63–84.
- Arroyo, D., C. Li, S. Li, G. Alvarez, and W. Halang**, 2009 : Cryptanalysis of an image encryption scheme based on a new total shuffling algorithm. *Chaos Solitons Fractals*, **41**, 2613–2616.
- Bandt, C., and B. Pompe**, 2002 : Permutation entropy : a natural complexity measure for time series. *Phys. Rev. Lett.*, **88**.
- Barenghi, C.**, 2010 : Introduction to chaos : Theoretical and numerical methods.
- Bechikh, R., H. Hermassi, A.A.A. El-Latif, R. Rhouma, and S. Belghith**, 2015 : Breaking an image encryption scheme based on a spatiotemporal chaotic system. *Signal Processing : Image Commun.*, **39**, 151–158.
- Belazi, A., A. El-Latif, and S. Belghiht**, 2016 : A novel image encryption scheme based on substitution-permutation network and chaos. *Signal Process.*, **128**, 155–170.
- Belazi, A., H. Hermassi, R. Rhouma, and S. Belghith**, 2014 : Algebraic analysis of a RGB image encryption algorithm based on DNA encoding and chaotic map. *J. NonLinear Dyn.*, **76**, 1989–2009.
- Benjeddou, A., A. Taha, D. Fournier-Prunaret, and R. Bouallegue**, 2008 : A New Cryptographic Hash Function Based on Chaotic S-box. *CSNDSP, Austria*.
- Boneh, D.**, 1998 : The decision Diffie-Hellman problem. In Algorithmic Number Theory. *3rd Intl. Symposium, volume 1423 of Lecture Notes in Computer Science*, 48–63.
- Bresson, E.**, 2015 : Cours de cryptographie. *Cours de Master à l'université de Paris XII, Val de marne, Laboratoire de cryptographie*.

- Cheddad, A., J. Condell, K. Curran, and P. McKeivitt**, 2010 : A hash-based image encryption algorithm. *Optics Communications*, **283**, 879–893.
- Chen, G., Y. Mao, and C.K. Chui**, 2004 : A symmetric image encryption scheme based on 3D chaotic cat maps. *Chaos, Solitons and Fractals*, **21**, 749–761.
- Chen, J.X., Z.L. Zhu, C. Fu, H. Yu, and Y. Zhang**, 2015 : Reusing the permutation matrix dynamically for efficient image cryptographic algorithm. *Signal Process.*, **111**, 294–307.
- Chen, M., D. Zhou, and Y. Shang**, 2005 : A sliding mode observer based secure communication scheme. *Chaos, Solitons and Fractals*, **25**, 573–578.
- Chong, F., B. Lin, Y. Miao, and authers.**, 2011 : A novel chaos-based bit-level permutation scheme for digital image encryption. *Opt. Commun.*, **284**, 5415–5423.
- Congxu, Z., X. Siyuan, H. Yuping, and S. Kehui**, 2015 : Breaking a novel image encryption scheme based on Brownian motion and PWLCM chaotic system. *Nonlinear Dyn*, **79**, 1511–1518.
- de Oliveira, L.P.L., and M. Sobottka**, 2008 : Cryptography with chaotic mixing. *Chaos, Solitons and Fractals*, **35**, 466–471.
- Dedieu, H., and M. Kennedy**, 1995 : Chaos Shift Keying : Modulation and Demodulation of a Chaotic Carrier Using Self-Synchronising Chua’s Circuits. *IEEE*.
- Dedieu, H., M.P. Kennedy, and M. Hasler**, 1993 : Chaos shift keying : modulation and demodulation of a chaotic carrier using self-synchronizing chua’s circuits. *IEEE Trans. on Circ. Syst. I*, **40**, 634–642.
- Deng, S., L. Zhang, D. Xiao, X. Wang, Liao, and Z. Yi**, 2005 : Image encryption scheme based on chaotic neural system. *In J. Advances in neural networks ISNN 2005, LNCS*, **3497**, 868–872.
- Dumas, J., J. Roch, H. Tannier, and S. Varette**, 2006 : Théorie des codes : Compression, cryptage, correction. *Support de cours de LMD à l’Institut National Polytechnique de Grenoble (INPG)*.
- Dumas, J., J. Rosch, E. Tannier, and S. Varette**, 2007 : Théorie des codes, Compression, cryptage, correction. *Cours de Master à l’université de Luxembourg*.
- Dumont, R.**, 2009 : Cours de cryptographie et sécurité. *Faculté des sciences appliquées de l’université de Liège*.
- Effa, J.Y., B.Z. Essimbi, and J.M. Ngundam**, 2009 : Synchronization of improved chaotic Colpitts oscillators using nonlinear feedback control. *Nonlinear Dynamics*, DOI : 10. 1007/s11071–008–9459–7.

- El-Latif, A., and X. Niu**, 2013 : A hybrid chaotic system and cyclic elliptic curve for image encryption. *Int. J. Electron. Commun. (AEÜ)*, **67**, 136–143.
- Enayatifar, R., H.J. Sadaei, A.H. Abdullah, M. Lee, and I.F. Isnin**, 2015 : A novel chaotic based image encryption using a hybrid model of deoxyribonucleic acid and cellular automata. *Optics and Lasers in Engineering*, **71**, 33–41.
- Eyebe, F.J.S.A.**, 2017 : Applicability of the permutation largest slope entropy to strange nonchaotic attractors. *Nonlinear Dyn.*, **87**, 1859. doi :10.1007/s11071-016-3158-6.
- Eyebe, F.J.S.A., B. Bodo, G.M. Djeufa, and S.L. Sabat**, 2016 : Experimental chaos detection in the Duffing oscillator. *Commun Nonlinear Sci. Numer. Simulat.*, **33**, 259–269.
- Eyebe, F.J.S.A., J. Effa, S. Sabat, and authers**, 2014 : Highly secured chaotic block cipher for fast image encryption. *Appl. Soft Comput.*, **25**, 435–444.
- Eyebe, F.J.S.A., J.Y. Effa, M. Kom, and M. Ali**, 2013 : The three-state test for chaos detection in discrete maps. *Applied Soft Computing*, **13**, 4731–4737.
- Eyebe, F.J.S.A., and W. Koepf**, 2015 : Detecting regular dynamics from time series using permutations slopes. *Commun Nonlinear Sci Numer Simul.*, **27**, 216–227.
- Feki, M.**, 2003 : An adaptive chaos synchronization scheme applied to secure communication. *Chaos, Solitons and Fractals.*, **18**, 141–148.
- Ferrenberg, A., D. Landau, and Y. Wong**, 2001 : Monte Carlo simulations : hidden errors from good random number generators. *Phys. Rev. Lett.*, **69**, 3382–3384.
- Fu, C., W. Meng, Y. Zhan, and authers**, 2013 : An efficient and secure medical image protection scheme based on chaotic maps. *Comput. Biol. Med.*, **43**, 1000–1010.
- Gao, T., and Z. Chen**, 2008a : Image encryption based on a new total shuffling algorithm. *Chaos Solitons and fractals*, **38**, 213–220.
- Gao, T., and Z. Chen**, 2008b : New image encryption algorithm based on hyper-chaos. *Phys. Lett. A.*, **372**, 394–400.
- Gonzalo, A., and L. Shujun**, 2006 : Some Basic Cryptographic Requirements for Chaos-Based Cryptosystems. *International Journal of Bifurcation and Chaos*, **16**, 2129–2151.
- Gopal, R., A. Venkatesan, and M. Lakshmanan**, 2013 : Applicability of 0-1 test for strange nonchaotic attractors. *CHAOS*, **23**, 023123.
- Gottwald, G., and I. Melbourne**, 2004 : A new test for chaos in deterministic systems. *Proc. R. Soc. A*, **460**, 603–611.
- Gottwald, G., and I. Melbourne**, 2009 : On the implementation of the 0–1 test for chaos. *SIAM J. Appl. Dyn. Syst*, **8**, 129–45.

- Guan, Z.H., F. Huang, and W. Guan**, 2005 : Chaos-based image encryption algorithm. *Physics Letters A*, **346**, 153–157.
- Guosheng, G., and H. Guoqiang**, 2006 : An Enhanced Chaos Based Image Encryption Algorithm. *IEEE Proceedings of the First International Conference on Innovative Computing, Information and Control (ICICIC'06)*.
- Hasler, M.**, 1995 : Engineering chaos for encryption and broadband communication. *Phil. Trans. Royal Soc. London*, 115–126.
- Hermassi, H., A. Belazi, R. Rhouma, and S. Belghith**, 2014 : Security analysis of an image encryption algorithm based on a DNA addition combining With chaotic maps. *Multimed. Tools. Appl.*, **72**, 2211–2224.
- Huang, X.**, 2012 : Image encryption algorithm using chaotic Chebyshev generator. *Nonlinear Dyn.*, **67**, 2411–2417.
- IEEE**, 2008 : Std 754. *IEEE standard for floating-point arithmetic*, 1–70.
- Jain, A., and N. Rajpal**, 2015 : A robust image encryption algorithm resistant to attacks using DNA and chaotic logistic maps. *Multimed. Tools. Appl.*, **75**, 5455–5472.
- Jianfeng, Z., W. Shuying, C. Yingxiang, and L. Xianfeng**, 2015 : A novel image encryption scheme based on an improper fractional-order chaotic system. *Nonlinear Dyn.*, DOI 10.1007/s11071-015-1911-x.
- Jing, H., T. Wen-wen, G. Jianbo, and authers**. Reliability of the 0-1 test for chaos.
- Jongseok, C., S. Seonhee, S. Hwajeong, and K. Howon**, 2016 : Fast ARX model-based image encryption scheme. *Multimedia Tools Appl.*, doi : 10.1007/s11042-016-3274-9.
- Kenfack, G., and A. Tiedeu**, 2014 : Chaos-based encryption of ECG signals : experimental results. *J. Biomed. Sci. Eng.*, **7**, 368–379.
- Kerckhoffs, A.**, 1883 : La cryptographie militaire. *Journal des sciences militaires*, **9**, 5–38.
- Koblitz, N.**, 1987 : Elliptic curve cryptosystems. *Mathematics of Computation*, **48**, 203–209.
- Kocarev, L.**, 2001 : Chaos-based cryptography : A brief overview. *IEEE Circuits and Systems Magazine*, **1**, 6–21.
- Kostelich, E.J., and D.P. Lathrop**, 1992 : The Prediction of Chaotic Time Series : a Variation on the Method of Analogues. *WAS and G.N.A (Eds.), Time series prediction : Forecasting the futur and understanding the past, Addison-Wesley*, 283–295.
- Kumar, M., A. Iqbal, and P. Kumar**, 2016 : A new RGB image encryption algorithm based on DNA encoding and elliptic curve Diffie-Hellman cryptography. *Signal*

- Process.*, **125**, 187–202.
- Kunt, M.**, 1993 : Traitement numérique des images. , **2**.
- Kwok, H., and W. Tang**, 2005 : A chaos-based cryptographic hash function for message authentication. *Int J Bifurcation Chaos*, **15**.
- Lala, K., B. Sami, A. Thawar, and Z. Shaaban**, 2009 : Image encryption using DCT and stream cipher. *European Journal of Scientific Research*, **32**, 48–58.
- L’Ecuyer, P.**, 2001 : Software for uniform random number generation : distinguishing the good and the bad. *Proceedings of the 2001 Winter Simulation Conference, IEEE Press*, 95–105.
- L’Ecuyer, P.**, 2004 : Random Number Generation. *chapter 2 of the Handbook of Computational Statistics, J. E. Gentle, W. Haerdle, and Y. Mori, eds., Springer-Verlag*, 35–0.
- Li, C., S. Li, and K.T. Lo**, 2011 : Breaking a modified substitution–diffusion image cipher based on chaotic standard and logistic maps. *Commun. Nonlinear Sci. Numer. Simul.*, **16**, 837–843.
- Li, C., S. Li, and A. Muhammad**, 2009 : On the security defects of an image encryption Scheme. *Image Vis. Comput.*, **27**, 1371–1381.
- Li, S., G. Alvarez, Z. Li, and W. Halang**, 2007 : Analog Chaos-based Secure Communications and Cryptanalysis : A Brief Survey. *In Proceedings of 3rd International IEEE Scientific Conference on Physics and Control (PhysCon’ 2007), Potsdam, Germany*, 1–6.
- Li, S., X. Mou, Z. Ji, and J. Zhang**, 2003 : Cryptanalysis of a class of chaotic stream ciphers. *J. Electronics and Information Technology*, **25**, 473–478.
- Li, T., and J. York**, 1975 : Period three implies chaos. *Amer. Math.*, **82**, 985–992.
- Lian, S., J. Sun, and Z. Wang**, 2005 : A block cipher based on a suitable use of the chaotic standard map. *Chaos, Solitons and Fractals*, **26**, 117–129.
- Liu, H., and Y. Liu**, 2014 : Cryptanalyzing an image encryption scheme based on hybrid chaotic system and cyclic elliptic curve. *Optics and Laser Technology*, **56**, 15–19.
- Liu, L., Q. Zhang, and X. Wei**, 2012 : A RGB image encryption algorithm based on DNA encoding and chaotic map. *J. Comput. Electric. Eng.*, **28**, 1240–1248.
- Liu, Y., J. Tang, and T. Xie**, 2014 : Cryptanalyzing a RGB image encryption algorithm based on DNA encoding and chaos map. *Optics and Laser Technology*, **60**, 111–115.
- Lorenz, E.**, 1963 : Deterministic non-periodic flows. *Journal of the Atmospheric Sciences*, **20**, 130.

- Luby, M.**, 1996 : Pseudorandomness and Cryptographic Applications. *Princeton University Press*.
- Mao, Y., G. Chen, and S. Lian**, 2004 : A novel fast image encryption scheme based on 3D chaotic baker maps. *International Journal of Bifurcation and Chaos*, **14**, 3613–3624.
- Masuda, N., and K. Aihara**, 2002 : Cryptosystems with discretized chaotic maps. *IEEE Trans. Circuits Syst.I*, **49**, 28–40.
- Matsumoto, M., and T. Nishimura**, 1998 : Mersenne twister, A 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Transactions on Modelling and Computer Simulation*.
- Matsumoto, T.**, 1997 : Chaos in electronic circuits. *Proc. IEEE*, **75**, 1033–1057.
- Matthews, R.A.J.**, 1989 : On the derivation of a chaotic encryption algorithm. *Cryptologia XIII*, 29–42.
- Memon, Q.**, 2003 : Synchronized chaos for network security. *Comp. Comm.*, **26**, 498–505.
- Miller, V.**, 1985 : Use of elliptic curves in cryptography. *CRYPTO*, **85**.
- Munir, R.**, 2012 : Robustness analysis of selective image encryption algorithm based on arnold cat map permutation. *In Proceedings of 3rd Makassar International Conference on Electrical Engineering and Informatics*, 1–5.
- Nien, H.H., W.T. Huang, C.M. Hung, S.C. Chen, S.Y. Wu, C.K. Huang, and authers**, 2009 : Hybrid image encryption using multi-chaos-system. *In 7th International Conference on Information, Communications and Signal Processing (ICICIS)*, 1–5.
- Nitaj, A.**, 2011 : La cryptographie du futur. *Cours de Cryptographie. Laboratoire de Mathématiques Nicolas Oresme Université de Caen, France*. <http://www.math.unicaen.fr/~nitaj>.
- Norouzi, B., and S. Mirzakuchaki**, 2016 : Breaking an Image Encryption Algorithm based on the New Substitution Stage with Chaotic Functions. *Optik - Int. J. Light Electron.*, **127**, 5695–5701.
- Oppenheim, A.V., K.M. Cuomo, and S.H. Strogatz**, 1993 : Synchronization of lorenz-based chaotic circuits with applications to communications. *IEEE Trans. on Circ. Syst. II*, **40**, 626–633.
- Panduranga, H., N. Kumar, and S. Kiran**, 2014 : Image encryption based on permutation-substitution using chaotic map and Latin square image cipher. *The European Physical J.-Spec. Topics*, **223**, 1663–1677.

- Pareek, N.K., V. Patidar, and K.K. Sud**, 2003 : Discrete chaotic cryptography using external key. *Phys. Lett. A*, **309**, 75–82.
- Pareek, N.K., V. Patidar, and K.K. Sud**, 2006 : Image encryption using chaotic logistic map. *Image Vis. Comput.*, **24**, 926–934.
- Parker, T.S., and L.O. Chua**, 1987 : Chaos : A tutorial for engineers. *In Proc. IEEE*, **75**, 982–1008.
- Parlitz, U., L.O. Chua, L. Kocarev, K.S. Halle, and A. Shang**, 1993 : Transmission of digital signals by chaotic synchronization. *Int. J. of Bifurcat. and Chaos*, **2**.
- Parvin, Z., H. Seyedarabi, and M. Shamsi**, 2014 : A new secure and sensitive image encryption scheme based on new substitution with chaotic function. *Multimed. Tools Appl.*, 1–18.
- Patidar, V., N.K. Pareek, G. Purohit, and K.K. Sud**, 2010 : Modified substitution–diffusion image cipher using chaotic standard and logistic maps. *Commun. Nonlinear Sci. Numer. Simul.*, **15**, 2755–2765.
- Patidar, V., N.K. Pareek, and K.K. Sud**, 2009 : A new substitution-diffusion based image cipher using chaotic standard and logistic maps. *Commun. Nonlinear Sci. Numer. Simul.*, **14**, 3056–3075.
- Pecora, L.M., and T.L. Carroll**, 1990 : Synchronization in chaotic systems. *Phys. Lett. A.*, **64**, 821–824.
- Pisarchik, A.N., and M. Zanin**, 2008 : Image encryption with chaotically coupled chaotic maps. *Physica D*, **237**, 2638–2648.
- Priya, R., and Sankpal**, 2014 : Image encryption using chaotic map : a survey. *In Proc of Fifth International Conference on Signal and Image Processing (ICSIP), Jeju, Island*, 102–107.
- Radu, B., C. Ana, and P. Iustin**, 2014 : A new hyperchaotic map and its application in an image encryption scheme, Signal Process. *Image Commun.*, **29**, 887–901.
- Rhouma, R., D. Arroyo, and S. Belghith**, 2009 : A new color image cryptosystem based on a piecewise linear chaotic map. *In 6th International Multi-Conference on Systems, Signals and Devices*, 1–6.
- Rhouma, R., and S. Belghith**, 2008 : Cryptanalysis of a new image encryption algorithm based on hyper-chaos. *Phys Lett. A.*, **372**, 5973–5978.
- Rivest, R.**, 1992 : The MD5 message-digest algorithm.
- Shannon, C.**, 1949 : Communication Theory of Secrecy Systems. *Bell Systems Technical Journal*, **28**, 656–715.

- Sinha, A., and K. Singh**, 2013 : Image encryption using fractional Fourier transform and 3D Jigsaw transform. Retrieved from <http://pdf-world.net/pdf-2013/Image-encryption-using-fractional-Fourier-transform-and-3D-Jigsaw-transform-pdf.pdf>.
- Sobhy, M., and A. Shehata**, 2001a : Chaotic Algorithms for Data Encryption. *IEEE Proceeding of ICASSP*, **2**, 997–1000.
- Sobhy, M., and A. Shehata**, 2001b : Methods of Attacking Chaotic Encryption and Countermeasures. *IEEE Proceeding of ICASSP*, **2**, 1001–1004.
- Song, C.Y., and X.Z. Qia, Y.-L. and Zhang**, 2013 : An image encryption scheme based on new spatiotemporal chaos. *Optik*, **124**, 3329–3334.
- Stinson, D.**, 2007 : Cryptography : Theory and Practice. *CRC press*.
- Sui, L., and B. Gao**, 2013 : Single-channel color image encryption based on iterative fractional Fourier transform and chaos. *Optics and Laser Technology*, **48**, 117–127.
- Takougang Kingni, S., J. Hervé Talla Mbé, and P. Wofo**, 2012 : Semiconductor lasers driven by self-sustained chaotic electronic oscillators and applications to optical chaos cryptography. *Chaos : An Interdisciplinary Journal of Nonlinear Science*, **22**, 033108.
- Tchitnga, R., P. Louodop, H. Fotsin, P. Wofo, and A. Fomethe**, 2013 : Synchronization of simplest two-component Hartley’s chaotic circuits : influence of channel. *Nonlinear Dynamics*, **74**, 1065–1075.
- Vashisth, S., H. Singh, A. Yadav, and K. Singh**, 2014 : Image encryption using fractional Mellin transform, structured phase filters, and phase retrieval. *Optik-International Journal for Light and Electron Optics*, **125**, 5309–5315.
- Wang, X., and W. Qiang**, 2014 : A Novel image encryption algorithm based on dynamic S-boxes constructed by chaos. *Nonlinear Dyn.*, **75**, 567–576.
- Wang, Z., A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli**, 2004 : Image quality assessment : From error visibility to structural similarity. *IEEE transactions on Image Processing*, **13**, 600–612.
- Wua, X., H. Kan, and J. Kurths**, 2015 : A new color image encryption scheme based on DNA sequences and multiple improved 1D chaotic maps. *Applied Soft Comput.*, **37**, 24–39.
- Xiang, T.**, 2007 : A novel symmetrical cryptosystem based on discretized two-dimensional chaotic map. *Physics letters. A*, **364**, 252–258.
- Xiao, D., X. Liao, and K. Wong**, 2005 : An efficient entire chaos-based scheme for

- deniable authentication. *Chaos Solitons Fractals*, **23**, 1327–1331.
- Xiao, H.P., and G.J. Zhang**, 2006 : An image encryption scheme based on chaotic systems. *IEEE Proceedings of the Fifth International Conference on Machine Learning and Cybernetics, Dalian*.
- Xingyan, W., and L. Chuanming**, 2016 : A novel and effective encryption algorithm based on chaos and DNA encoding. *Multimedia Tools Appl.*, doi :10.1007/s11042-016-3311-8.
- Xu, L., Z. Li, J. Li, and W. Hua**, 2016 : A novel bit-level image encryption algorithm based on chaotic maps. *Optics and Lasers in Engineering.*, **78**, 17–25.
- Yang, T.**, 2004 : A survey of chaotic secure communication systems. *International Journal of Computational Cognition*, **2**, 81–130.
- Ying, W., Z. DeLing, J. Lei, and authers**, 2004 : The spatial-domain encryption of digital images based on high-dimension chaotic system. *Proceedings of 2004 IEEE Conference on Cybernetics and Intelligent Systems, Sin-gapore*, 1172–1176.
- Yinhe, C., T. Wen-wen, J.B. Gao, V.A. Protopopescu, and L.M. Hively**, 2004 : Detecting dynamical changes in time series using the permutation entropy. *Physical Review E*, **70**, 046217.
- Zhang, B., and C. Jin**, 2008 : Cryptanalysis of a Chaos-based Stream Cipher. *In Proceedings of the 9th International Conference for Young Computer Scientists, Hunan, China*.
- Zhang, L., Z. Zhuand, B. Yang, W.Y. Liu, and M. Zhug, H.-F.and Zou**, 2015 : Cryptanalysis and improvement of an efficient and secure medical image protection scheme. *Math. Probl. Eng.*, 11.
- Zhang, Q., L. Guo, and X. Wei**, 2010 : Image encryption using DNA addition combining with chaotic maps. *J. Math. Comput. Modeling*, **52**, 2028–2035.
- Zhang, W., H. Yu, Y.I. Zhao, and Z.L. Zhu**, 2016 : Image encryption based on three-dimensional bit matrix permutation. *Signal process.*, **118**, 36–50.
- Zhang, Y.Q., and X.Y. Wang**, 2014 : Analysis and improvement of a chaotic-based symmetric image encryption scheme using a bit-level permutation. *Nonlinear Dyn.*, **77**, 687–698.
- Zhengjun, L., C. Hang, L. Ting, L. Pengfei, X. Lie, D. Jingmin, and L. Shutian**, 2011 : Image encryption by using gyration transform and Arnold transform. *Journal of Electronic Imaging*, **1**, 3020–3026.
- Zhou, Q., K.W. Wong, X. Liao, T. Xiang, and Y. Hu**, 2008 : Parallel image encryp-

tion algorithm based on discretized chaotic map. *Chaos, Solitons and Fractals*, **38**, 1081–1092.

Zhou, Y., L. Bao, and C.L.P. Chen, 2014 : A new 1D chaotic system for image encryption. *Signal Process*, **97**, 172–182.

Zhu, Z., W.W. Zhang, K. Wong, and H. Yu, 2011 : A chaos-based symmetric image encryption scheme using a bit-level permutation. *Info. Sci.*, **181**, 1171–1186.

Liste des publications

1. **Yannick Abanda**, Alain Tiedeu (2016) : **Image encryption by chaos mixing**. *IET Image Processing*. **10** :742-750. DOI : 10.1007/s00382-017-3547-7.
2. Alain Tiedeu, **Yannick Abanda** and Gutenbert Kenfack (2018) : **E-medicine : A Secure Transmission of Electrocardiograms Using Chaos Oscillators Synchronization**. *In : C. M. F. Kebe et al. (Eds.) : Intersol 2018, LNICST. Springer*.1-10. DOI : 10.1007/978-3-319-98878-8_8.

Conférences et ateliers

1. Alain Tiedeu, **Yannick Abanda : Image Processing : Some methods for Chaos-based 1D and 2D signal encryption**. Lam 11 International workshop to celebrate international year of light : optics photonics and lasers in science and technology for sustainable development and AFSIN international workshop on spectral imaging launching of the Africans optics and photonics society. 23-27 November 2015, Saly, Senegal ;
2. **Yannick Abanda**, Alain Tiedeu : **Image encryption Using Optics devices**. AFSIN international workshop on Optical simulation and Optical system design and contribution, 18-30 November 2017, Ouagadougou, Burkina Faso ;
3. **Yannick Abanda**, Alain Tiedeu : **Medical image encryption using a mix of Colpitts and Duffing maps**. Cameroon physical society (SCP) Physics for medicine Conference, 04-08 December 2017 Yaoundé, Cameroun ;
4. **Yannick Abanda**, Alain Tiedeu : **E-medicine : a secure transmission of electrocardiograms using chaos synchronization**. EAI International conference on innovations and interdisciplinary solutions for underserved areas. 24-25 March, 2018, Kigali, Rwanda.

The IET Premium Awards 2018

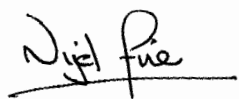
This is to certify that

Yannick Abanda and Alan Tiedeu

are awarded

the IET Image Processing Premium Award for the paper 'Image encryption by chaos mixing',
Volume 10, Issue 10, 2016, p. 742-750

June 2018



Nigel Fine
IET Chief Executive and Secretary



Nick Winser
IET President

www.ietidl.org/journals