République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

UNIVERSITÉ YAHIA FARÈS DE MÉDÉA

Faculté de Technologie

Département de Génie Electrique

Laboratoire des Systèmes Electroniques Avancés (LSEA)

THESE DE DOCTORAT

Présentée par :

Mr. BENZEGANE SAMIR

Ingénieur d'Etat en Electronique : Université de Tizi-Ouzou, Algérie

Magistère en Télécommunication : EMP. Alger, Algérie

En vue de l'obtention du titre de Doctorat en Sciences en Génie Electrique

Thème

Contribution à la Sécurité dans la Transmission des Données des Habitations Intelligentes

Soutenue le, 10 /10 /2024 devant le jury composé de :

Prof. HOUASSINE Hamza	Professeur (U. Bouira)	Président
Dr. BETTACHE Farouk	MCA (U. Médéa)	Examinateur
Dr. ABIDINE M'hamed Bilal	MCA (U. USTHB)	Examinateur
Dr. REZKI Mohamed	MCA (U. Bouira)	Examinateur
Prof. BATEL Noureddine	Professeur (U. Médéa)	Directeur de thèse
Prof. BOUHEDDA Mounir	Professeur (U. Médéa)	Co-directeur de thèse

Université de Médéa, Pôle universitaire d'Ouzera, Algérie

Remerciements

Ce travail a été réalisé au sein du Laboratoire des Systèmes Electroniques Avancés (LSEA) de l'Université Yahia Farés de Médéa sous la direction du Pr. Batel Noureddine et du Pr. Bouhedda Mounir, qu'ils trouvent ici le témoignage de ma profonde gratitude pour leur attention, leurs conseils fructueux et leur contribution à la réussite de cette thèse.

Je tiens à exprimer ma parfaite considération aux membres de jury qui ont bien voulu examiner et juger mon travail : Prof. HOUASSINE Hamza, Dr. BETTACHE Farouk, Dr. ABIDINE M'hamed Bilal et le Dr. REZKI Mohamed.

Enfin, que tous ceux qui m'ont aidé, de près ou de loin, tout au long la réalisation de ce travail, trouvent ici mes sincères remerciements.

Dédicace

Je dédie ce travail à toute ma famille et aux personnes qui me sont chères.

Sommaire

Sommaire.....III
Resumé..... VII
Liste des figures X
Liste des tableaux XIII
Liste des acronymes.....XIV

Introduction Générale.....1

Chapitre I: L'Internet Des Objets et son Application dans les Maisons Intelligentes.....4

I.1 Introduction.....4
I.2 L'histoire de l'IoT.....4
I.3 Paradigme de l'IoT 5
I.4 Architectures de l'IoT.....6
I.4.1 Architecture à trois couches 6
I.4.2 Architecture à quatre couches 7
I.4.3 Architecture à cinq couches 8
I.5 Domaines d'applications de l'IoT.....9
I.6 Technologies et protocoles de communication de l'Internet des Objets 12
I.6.1 Protocoles de Communication 12
I.6.2 Protocoles de Données 17
I.7 Les défis de l'IoT 19
I.8 Les maisons intelligentes (smart homes) 21
I.8.1 Applications des IoT dans les maisons intelligentes.....21
I.8.2 Composition d'une maison intelligente 22
I.8.3 Les réseaux de capteurs sans fil dans les maisons intelligentes.....24
I.9 Menaces et Attaques dans les maisons intelligentes.....25
I.9.1 Écoute clandestine (Eavesdropping).....25
I.9.2 Denial of Service (DoS).....26
I.9.3 Hijacking (Détournement) 26
I.9.4 Sinkhole and WormholeAttacks 27
I.10 Stratégies de sécurisation des maisons intelligentes 28
I.10.1 L'authentification.....28
I.10.2 La fraicheur des données.....29
I.10.3 La confidentialité 30
I.10.4 L'anonymat 31
I.10.5 L'intégrité.....32
I.10.6 La disponibilité 32
I.10.7 La non répudiation 32
I.11 Conclusion 33

Chapitre II : Chaos en Cryptographie 34

II.1 Introduction 34
II.2 Théorie du chaos 34
II.2.1 Définition du Chaos 34
II.2.2 Historique 35
II.2.3 Systèmes dynamiques 38
II.3 Systèmes chaotiques 38
II.3.1 Systèmes chaotiques à temps continu 39
II.3.2 Systèmes chaotiques à temps discret 41
II.3.3 Systèmes hyperchaotiques 41
II.4 Propriétés des systèmes chaotiques 42
II.4.1 Déterminisme 42
II.4.2 Sensibilité aux conditions initiales 42
II.4.3 Aspect aléatoire 43
II.4.4 Imprédictibilité 43
II.4.5 Apériodicité 44
II.5 Outils d'étude des systèmes chaotiques 44
II.5.1 Diagramme de bifurcation 45
II.5.2 Exposant de Lyapunov 46
II.5.3 Attracteur étrange 47
II.5.4 Section dePoincaré 48
II.6 Notion de base et primitives cryptographique 48
II.6.1 La cryptographie 49
II.6.2 La cryptographie symétrique 50
II.6.3 La cryptographie asymétrique 50
II.6.4 Le masque jetable (Chiffrement de Vernam) 51
II.6.5 La cryptanalyse 51
II.7 Cryptographie basée-chaos 53
II.7.1 Chaos vs Cryptographie 54
II.7.2 Type decrypto-systèmes chaotiques 55
II.8 Outils d'évaluation de la qualité des clés de chiffrement 56
II.9 Métriques et analyse de sécurité des images chiffrées 57
II.9.1 Analyse différentielle 59
II.9.2 Entropie 58
II.9.3 Espace des clés 59
II.9.4 Sensibilité des clés 59
II.9.5 Histogramme 59
II.9.6 Test de chicarré 60
II.9.7 Corrélation des pixels adjacents 60
II.10 Conclusion 61

Chapitre III: Cryptosysteme Hyperchaotique pour Reseau de Capteurs Sans Fil 62

III.1 Introduction.....62
III.2 Implémentation du générateur hyperchaotique de Chen.....62
III .2.1 Signaux temporels et attracteurs hyperchaotiques.....63
III.3 Génération des clés de chiffrement.....65
III .3.1 Signaux temporels et attracteurs du GHCC.....67
III .3.2 Section Poincaré.....69
III .3.3 Choix du générateur de Chen.....70
III .3.4 Tests statistiques.....71
III.4 Structure du cryptosystème proposé.....74
III.5 Analyse de sécurité d'images chiffrées.....75
III .5.1 Analyse différentielle.....75
III .5.2 Entropie.....76
III .5.3 Espace des clés.....77
III .5.4 Sensibilité des clés.....77
III .5.5 Histogramme.....78
III .5.6 Corrélation des pixels adjacents.....80
III.6 La synchronisation chaotique dans un réseau informatique.....82
III .6.1 Principe de la synchronisation chaotique.....83
III .6.2 Problème de la synchronisation chaotique.....85
III .6.3 Solutions proposées.....84
III .6.4 Fonction de permutation.....85
III.7 Approche et déploiement de l'architecture proposée.....86
III .7.1 Description de l'IHM.....89
III .7.2 Algorithme de Cryptage Hyperchaotique.....91
III .7.3 Analyse et Performance du Schéma Cryptographique proposé.....95
III.8 Conclusion.....98

Chapitre IV: Sécurisation d'un Réseau de Capteurs Sans Fil Heterogene Basé Chaos.....99

IV.1 Introduction.....99
IV.2 Le système chaotique unifié.....100
IV.3 Système Chaotique Unifié Discrétisé(SCUD).....100
IV.4 Les signaux temporels et les attracteurs.....101
IV.5 Génération des clés de chiffrement chaotiques.....106
IV.5.1 Principe du générateur chaotique.....107
IV.5.2 Signaux temporels et attracteurs du GSCUD.....108
IV.5.3 Tests statistiques.....109
IV.6 Structure du cryptosystème proposé.....109
IV.7 Analyse de sécurité d'images chiffrées.....110

IV.7.1 Analyse différentielle 110
IV.7.2 Test d'entropie 111
IV.7.3 Espace des clés 111
IV.7.4 Sensibilité des clés 111
IV.7.5 Histogramme 112
IV.7.6 Corrélation des pixels adjacents 113
IV.8 Analyse de sécurité du signal audio 115
IV.8.1 Analyse d'histogramme 115
IV.8.2 Analyse de spectrogramme 116
IV.8.3 Fonction d'autocorrélation 116
IV.9 Structure et déploiement de l'architecture proposée 117
IV.9.1 Solution proposée pour le problème de la synchronisation 118
IV.9.2 Fonctions de Permutations 120
IV.10 Cryptosysteme pour la sécurisation des données du DHT22 122
IV.10.1 Au niveau de l'émission 122
IV.10.2 Au niveau de la réception 123
IV.11 Cryptosystème pour la sécurisation des données du smartphone 124
IV.11.1 Sécurisation du flux vidéo 125
IV.11.2 Sécurisation du flux audio 126
IV.12 Procédure d'initialisation du générateur GSCUD 127
IV.13 Réseau de capteur réalisé 127
IV.13.1 Maquette expérimentale 128
IV.13.2 Description des différents nœuds 128
IV.13.3 Description de la carte UDOO QUAD (Serveur) 131
IV.13.4 IHM développée 132
IV.14 Test de validation et résultat 133
IV.15 Analyse et Performance du Schéma Cryptographique proposé 134
IV.16 Conclusion 136

Conclusion Générale et Perspectives **138**

Bibliographie **140**

Résumé

L'objectif principal de cette thèse est le développement de deux cryptosystèmes basés sur des systèmes chaotiques pour sécuriser les données dans le contexte des Maisons Intelligentes utilisant la technologie de l'Internet des Objets(IdO). Le premier cryptosysteme utilise le Générateur Hyperchaotique de Clé de Chiffrement (GHCC) à base du système de Chen pour générer de clés de chiffrement de données avant leur transmission. Tandis que le deuxième cryptosysteme exploite le Système Chaotique Unifié Discrétisé (SCUD), couplé avec une carte logistique, comme générateur de clés de chiffrement avec de bonnes performances statistiques. Le critère aléatoire des clés générées a été exhaustivement évalué au moyen de la suite de tests du NIST. Tous les tests du NIST ont été passés avec succès ce qui confirme l'excellente qualité des clés ainsi produites par les deux générateurs GHCC et SCUD, ce qui permet de les utiliser comme des clés de chiffrement dans le processus de cryptage et de décryptage. De plus, une analyse statistique et différentielle a été réalisée incluant des tests d'histogramme, d'entropie, de sensibilité des clés et de l'espace des clés.

La première plateforme matérielle, basée sur un microcontrôleur ARM, réalisée est constituée de deux nœuds et d'un serveur communicant à travers le réseau Wifi. Quant à la deuxième plateforme elle inclue trois nœuds et un serveur. Les données de température, d'humidité et de flux audio/vidéo issues des capteurs sont transmises vers le serveur à travers les technologies de transmission sans fils XBee, Wifi et Bluetooth. Il a été vérifié et démontré dans cette thèse que les plateformes développées, à base de Réseau de Capteur San fil (RCSF), répondent à toutes les principales exigences de sécurité des données (confidentialité, authentification et non-répudiation, intégrité, résilience et fraîcheur des données) ainsi que le temps de traitement et de la consommation d'énergie des algorithmes proposés. Les différentes applications développées ont étés testées et validées expérimentalement.

Les résultats expérimentaux obtenus dans cette thèse sont très satisfaisants et concluants démontant ainsi l'efficacité de notre approche.

Mots clés **:** Cryptosysteme, IdO, Maison Intelligente, NIST, RCSF, Système de Chen, Système Chaotique Unifié.

Abstract

The main purpose of this thesis is to develop and design two secure and efficient cryptosystems based on chaos system to secure data within the context of Smart Home automation applications based on Internet of Thing (IoT) technology. The first cryptosystem use a Hyperchaos-based Key Stream Generator (H-KSG) based on hyperchaotic Chen system to generate keys for a encrypt data before transmission. While, the second cryptosystem exploits a Discrete Unified Chaotic System (DUCS), coupled with a logistic map, to generate encryption keys with good statistical performances. The randomness of the generated encryption keys, for the two generators, has been extensively tested using NIST testing strategy. By successfully passing all NIST tests, the proposed H-KSG and DUCS generators provides reliable random encryption keys that serve as keystream for both encryption and decryption process. Furthermore, a statistic and differential cryptanalysis was made studying different aspects of our cryptosystem such as randomness, histogram, entropy, sensitivity dependence on the key and the plaintext, key space. The experimental implementation of the first design was implemented on microcontroller-based ARM architecture with two sensor nodes boards which are connected to the server via the Wifi link, when the second design includes gateway server board and three sensor nodes boards. The temperature, humidity and video/audio data from the sensors are sent to the central receiver unit by using the XBee, Wifi and Bluetooth wireless technologies.

It has been also verified in this thesis that a developed Wireless Sensors Network(WSN) platforms fulfill all major security requirements data security (confidentiality, authentication and non-Repudiation, integrity, resilience, and data freshness). In addition, the proposed algorithms have a good running speed and less energy consumption. The software and hardware design of the proposed systems was implemented and tested successfully.

The results of cryptanalysis and the necessary security performance tests show the efficiency and the high level of security of the two proposed systems.

Keywords: Cryptosystem, IoT, Smart Home, NIST, WSN, Chen System, Unified Chaotic System.

الملخص

الغرض الرئيسي من هذه الأطروحة هو تطوير وتصميم نظامين تشفيريين آمنين وفعالين يعتمدان على نظام الفوضى لتأمين البيانات في سياق تطبيقات أتمتة المنزل الذكي القائمة على تقنية إنترنت الأشياء (IoT). يستخدم نظام التشفير الأول مولد دفق المفاتيح (H-KSG) القائم على استنادًا إلى نظام شان (Chen) مفرط الفوضى لإنشاء مفاتيح لتشفير البيانات قبل الإرسال. بينما يستغل نظام التشفير الثاني النظام الفوضوي الموحد (DUCS)، مقترنًا بخريطة لوجستية، لإنشاء مفاتيح تشفير ذات أداء إحصائي جيد. تم اختبار عشوائية مفاتيح التشفير المولدة للمولدين على نطاق واسع باستخدام استراتيجية اختبارنست(NIST). من خلال اجتياز جميع اختبارات نست بنجاح، توفر مولدات H-KSG وDUCS المقترحة مفاتيح تشفير عشوائية موثوقة تستعمل في عملية التشفير وفك التشفير. علاوة على ذلك، تم إجراء تحليل تشفير إحصائي وتفاضلي لدراسة جوانب مختلفة من نظام التشفير لدينا مثلا لرسم البياني، والإنتروبيا، والاعتماد على حساسية المفاتيح، ومساحة المفاتيح. تم تنفيذ التنفيذ التجريبي للتصميم الأول على معمارية ARM المبنية على المتحكم الدقيق مع لوحتي عقد استشعار متصلتين بالخادم عبر وصلة Wifi، بينما يتضمن التصميم الثاني لوحة خادم البوابة وثلاث لوحات عقد استشعار. يتم إرسال بيانات درجة الحرارة، الرطوبة والفيديو/الصوت من أجهزة الاستشعار إلى وحدة الاستقبال المركزية باستخدام تقنيات XBee وWifi وBluetooth اللاسلكية.

تم التحقق أيضًا في هذه الأطروحة من أن منصات شبكة الاستشعار اللاسلكية (WSN) المطورة تفي بجميع متطلبات الأمان الرئيسية لأمن البيانات (السرية، والمصادقة وعدم الإنكار، والنزاهة، والمرونة، ونضارة البيانات). بالإضافة إلى ذلك، تتمتع الخوارزميات المقترحة بسرعة تشغيل جيدة واستهلاك أقل للطاقة. وقد تم تنفيذ واختبار تصميم البرامج والأجهزة للأنظمة المقترحة بنجاح. تظهر نتائج تحليل الشفرات واختبارات الأداء الأمني اللازمة كفاءة ومستوى الأمان العالي للنظامين المقترحين.

الكلمات المفتاحية: نظام التشفير، إنترنت الأشياء، المنزل الذكي،NIST ، شبكة الاستشعار اللاسلكية (WSN)، نظام Chen، النظام الفوضوي الموحد.

LISTE DES FIGURES

I.1 Architecture à trois couches..... 6
I.2 Architecture à quatre couches..... 7
I.3 Architecture à cinq couches..... 8
I.4 Domaines d'application de l'IoT..... 9
I.5 Application de l'IoT dans le secteur de transport..... 10
I.6 Application de l'IoT dans les villes intelligentes..... 11
I.7 Application de l'IoT dans le domaine médical..... 11
I.8 Application de l'IoT dans une maison intelligente..... 12
I.9 Quelques protocoles de communication..... 13
I.10 Composition d'une maison intelligente..... 22
II.1 Système chaotique de Rössler …….…… 40
II.2 Evolution de la fonction Logistique 41
II.3 Sensibilité aux conditions initiales dans le système de Chen pour des conditions initiales voisines..... 43
II.4 Aspect aléatoire du système chaotique de Lorenz 44
II.5 Le signal chaotique de Chua..... 44
II.6 Diagramme de bifurcation de la fonction Logistique..... 45
II.7 Divergence de deux trajectoires dans le plan de phase..... 47
II.8 Exemples d'attracteurs étranges : attracteurs étranges de Chen..... 47
II.9 Exemple de sections de Poincaré d'un attracteur chaotique 48
II.10 Principe du chiffrement symétrique 50
II.11 Principe du chiffrement asymétrique 51
II.12 Principe de la cryptographie basée-chaos..... 53
II.13 Cryptographie basée-chaos de Vidal..... 54
III.1 Signaux temporels du système hyperchaotique de Chen..... 64
III.2 Attracteurs hyperchaotiques 2D et 3D du système de Chen..... 64
III.3 Exemple de séquences binaires (16Q16) générées par le system hyperchaotique de Lorenz..... 65
III.4 Schéma Bloc du GHCC..... 67
III.5 Signaux temporelsdes signaux hyperchaotiquesX1(t), Y1(t), Z1(t)et W1(t)..... 68
III.6 Attracteurs 2D et 3D de la partie fractionnaire de Chen..... 68
III.7 Attracteur 3D du générateur de Chen (à gauche) et du GHCC (à droite)..... 69
III.8 Section Poincaré du plan de phase (y, z) et (Y1,Z1)..... 69
III.9 La fonction d'autocorrélation des signaux X1(t), Y1(t), Z1(t) et W1(t)..... 70
III.10 La fonction d'intercorrélation des signaux X1(t), Y1(t), Z1(t) et W1(t)..... 70
III.11 La distribution des P_value..... 72
III.12 Proportions des séquences..... 73
III.13 Structure générale du cryptosystème..... 74
III.14 Chiffrement d'image : Sensibilité aux conditions initiales (top) et aux paramètres (bas)..... 78
III.15 Chiffrement d'image : Analyse d'histogrammes d'image en couleur (Airplane)..... 78
III.16 Chiffrement d'image : Analyse d'histogrammes d'image en couleur (Lena)..... 79
III.17 Chiffrement d'image : Analyse d'histogrammes d'image en couleur (Pepper)..... 79
III.18 Chiffrement d'image Airplane: corrélation des pixels adjacents dans l'image originale et chiffrée..... 81
III.19 Chiffrement d'image Peppers: corrélation des pixels adjacents dans l'image originale et chiffrée..... 81

III.20 Chiffrement d'image Mandrill : corrélation des pixels adjacents dans l'image originale et chiffrée..... 82
III.21 Schéma de principe de la SCI..... 83
III.22 Phase de transfert de données par TCP..... 85
III.23 Illustration du procédé de permutation pour un tableau de taille L=5..... 86
III.24 Composants des nœuds capteurs..... 87
III.25 Application typique du capteur DTH22..... 87
III.26 L'architecture de système de réseaux de capteurs sans fil..... 88
III.27 Photo de la maquette expérimentale..... 89
III.28 IHM développée..... 89
III.29 Zone de gestion des nœuds..... 90
III.30 Zone de gestion de l'animation..... 91
III.31 Principe du chiffrement par flot 92
III.32 Processus de cryptage..... 93
III.33 Organigramme de décryptage..... 94
III.34 Principe de mesure du courant consommé..... 97
IV.1 Système chaotique unifié en fonction de k..... 100
IV.2 Les signaux temporels pour k = 0..... 101
IV.3 Les signaux temporels pour k = 0.8..... 102
IV.4 Les signaux temporels pour k = 1..... 102
IV.5 Les attracteurs 2D pour k=0..... 103
IV.6 Les attracteurs 2D pour k=0.8..... 103
IV.7 Les attracteurs 2D pour k=1..... 104
IV.8 Les attracteurs 3D pour k=0..... 104
IV.9 Les attracteurs 3D pour k=0.8..... 105
IV.10 Les attracteurs 3D pour k=1..... 105
IV.11 Fichier contenant la représentation binaire de la clé X..... 106
IV.12 Schéma Bloc du GSCUD..... 107
IV.13 Signaux temporels des signaux X(t), Y(t) et Z1(t) du GSCUD..... 108
IV.14 Attracteurs 2D et 3D du GSCUD..... 108
IV.15 Sensibilité aux conditions initiales (haut) et aux paramètres (bas)..... 112
IV.16 Analyse d'histogrammes d'image (Lena)..... 112
IV.17 Analyse d'histogrammes d'image (Mandrill)..... 113
IV.18 Corrélation des pixels adjacents dans l'image Originale et chiffrée (Peppers)..... 114
IV.19 Corrélation des pixels adjacents dans l'image Originale et chiffrée (Airplane)..... 114
IV.20 Chiffrement de la parole : (a) signal parole original et (b) signal chiffré..... 115
IV.21 Analyse d'histogramme du chiffrement de la parole : (a) l'histogramme du signal parole original et (b) l'histogramme du signal chiffré..... 116
IV.22 Spectrogramme : (a) signal parole original et (b) signal chiffré..... 116
IV.23 Autocorrélation : (a) signal audio original et (b) signal audio chiffré..... 117
IV.24 Bloc diagramme du système..... 118
IV.25 Processus de chiffrement/déchiffrement..... 118
IV.26 Principe de la synchronisation coté émission..... 119
IV.27 Principe de la synchronisation coté réception..... 120
IV.28 Exemple de permutationFP1 d'un tableau 121
IV.29 Exemple de permutation FP2 d'un tableau..... 122
IV.30 Processus de cryptage..... 123
IV.31 Processus de décryptage..... 124
IV.32 Processus de chiffrement de la vidéo..... 125
IV.33 Processus de déchiffrement de la vidéo..... 126
IV.34 Maquette expérimentale..... 128
IV.35 L'ensemble des nœuds 129
IV.36 Composants du nœud 1..... 129
IV.37 Réalisation du nœud 1..... 130
IV.38 Réalisation du nœud 2..... 130
IV.39 L'application Android développée..... 131

IV.40 La carte UDOO QUAD..... 131
IV.41 Montage du serveur..... 132
IV.42 IHM serveur..... 132
IV.43 IHM complète..... 133
IV.44 Donnée des nœuds 1 et 2..... 133
IV.45 Donnée du nœud 3 audio/vidéo temps réel..... 134

LISTE DES TABLEAUX

I.1 Comparaison entre différentes technologies de communication sans fil utilisées dans l'IoT 19
II.1 Comparaison des propriétés chaotiques et cryptographiques ……..……..…… 55
III.1 Résultats des tests statistiques du NIST, la p_value_T des différents tests pour chaque séquence ……. 71
III.2 Résultats des tests du NIST, proportion des clés qui passent chaque test avec succès…… 73
III.3 Résultats d'analyse de l'attaque différentielle suivant les métriques NPCR, UACI et DH..... 75
III.4 Comparaison du NPRC, UACI et DH avec quelques références..... 76
III.5 Résultats de test d'entropie……..……...…….. 76
III.6 Test chi-carré sur différentes images……..……. 80
III.7 Coefficients de corrélation des pixels adjacents……..……. 82
III.8 Temps d'exécution en millisecondes..... 96
III.9 Consommation d'énergie en milli Joule..... 97
IV.1 Résultats des tests statistiques du NIST..... 109
IV.2 Résultats d'analyse du NPCR, l'UACI et DH..... 110
IV.3 Comparaison du NPRC, UACI et DH avec quelques références..... 110
IV.4 Résultats du test d'entropie..... 111
IV.5 Test de chi-carré..... 113
IV.6 Les coefficients de corrélation des pixels adjacents..... 115
IV.7 Temps d'exécution en millisecondes..... 136
IV.8 Consommation d'énergie en milli Joule..... 136

LISTE DES ACRONYMES

ACK	Acknowledgement
AES	Advanced Encryption Standard
AMQP	Advanced Message Queuing Protocol
BLE	Bluetooth Low Energy
CI	Conditions Initiales
CoAP	Constrained Application Protocol
CPL	Courant Porteur en Ligne
CRC	Cyclic Redundancy Check
CTR	Compteur
DCC	Direct Chaotique Communication
DES	Data Encryption Standard
DH	Distance de Hamming
DNS	Domain Name System
DoS	Denial of Service
FPGA	Field Programmable Gate Array
GHCC	Générateur basé Hyperchaos des Clés de Chiffrement
GSCUD	Générateur basé-Système Chaotique Unifié Discrétisé
H	Humidité
HTTP	Hyper Text Transfert Protocol
ID	Identifiant
IDEA	International Data Encryption Algorithm
IdO	Internet des Objets
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IHM	Interface Homme Machine
IoT	Internet of Things
IP	Internet Protocol
JPEG	Joint Picture Expert Group
LoRaWAN	Long Range Wide Area Network
MAC	Message Authentication Code
MCA	Masquage Chaotique par Addition
MD5	Message Digest 5
MEMS	Micro ElectroMechanical Systems
MIMO	Multiple Input Multiple Output
MIT	Massachusetts Institut of Technologies
MITM	Man-In-The-Middle

MQTT	Message Queuing Telemetry Transport
M2M	Machine To Machine
NFC	Near Field Communication
NIST	National Institut of Standards and Technologies
NPCR	Number of Pixel Change Rate
NTP	Network Time Protocol
PRNG	Pseudo Random Number Generator
QoS	Quality Of Service
RC4	Rivest Cipher 4
RCSF	Réseaux de Capteurs Sans Fil
RFID	Radio Frequency Identification
RGPD	Règlement General sur la Protection des Données
RK-4	Range Kutta d'ordre 4
RNG	Random Numberb Generator
RSA	Rivest -Shamir-Adleman
SCI	Synchronisation Chaotique Identique
SCUD	Système Chaotique Unifié Discrétisé
SHA	Secure Hash Algorithm
SIG	Special Interest Group
SMS	Short Message Service
SNMP	Simple Network Management Protocol
SQL	Structured Query Language
SSL/TLS	Secure Sockets Layer/Transport Layer Security
T	Température
TCP	Transmission Control Protocol
TRNG	True Random Number Generator
UACI	Unified Average Changing Intensity
UDP	User Datagram Protocol
UIT	Union Internationale des Télécommunications
URI	Uniform Resource Identifiers
VI	Vecteur d'Initialisation
XML	eXtensible Markup Language
XMPP	Extensible Messaging and Presence Protocol
Wi-Fi	Wireless Fidelity
6LoWPAN	IPv6 over Low-Power Wireless Personal Area Networks

INTRODUCTION GENERALE

L'habitat connecté ou la maison intelligente (Smart Home en anglais) [1,2] représente une application clé de l'Internet des Objets ou *Internet of things* (IoT) en anglais, intégrant des technologies permettant une communication fluide entre les appareils domestiques pour automatiser diverses fonctions. Au cœur de ce concept, les réseaux de capteurs sans fil (RCSF) jouent un rôle crucial, facilitant l'automatisation et améliorant le quotidien des utilisateurs grâce à une gestion plus intuitive de l'habitat [3,4]. Ces réseaux, composés de capteurs et d'actuateurs interconnectés sans fil, forment un écosystème de contrôle avancé. Ils collectent et analysent des données provenant de divers appareils pour prendre des actions adaptatives visant à améliorer le confort, optimiser la consommation d'énergie et renforcer la sécurité domestique. Chaque capteur au sein d'un RCSF intègre trois niveaux fonctionnels : la collecte des informations environnementale (Température, Humidité, Luminosité...), le traitement des données et la transmission des informations vers l'unité de contrôle et de gestion [5-8]. Cependant, la sécurité demeure une préoccupation majeure dans les systèmes de maison intelligente, où la vulnérabilité peut exposer les données personnelles des utilisateurs à des risques externes. La protection de ces données est donc primordiale pour les utilisateurs, qui exigent des garanties de sécurité pour leurs informations personnelles [9,10]. Dans le contexte des maisons connectées, où capteurs et actionneurs échangent constamment des données, l'intégrité et la confidentialité de ces dernières sont essentielles. Ainsi, l'adoption de solutions cryptographiques s'avère nécessaire pour sécuriser le réseau contre les menaces potentielles. Plusieurs approches, incluant des techniques de cryptage classiques et des stratégies de sécurité spécifiques aux RCSF, ont été développées pour résoudre cette problématique complexe [11-17] et permettant ainsi un cryptage et un décryptage efficace des données au niveau de chaque entité du RCSF.

Dans ce contexte, la cryptographie basée-chaos s’avère une candidate très attractive et prometteuse car elle offre une flexibilité dans le choix et la personnalisation du cryptosystème à utiliser. Cette attractivité est due aux propriétés de déterminisme et d’extrême sensibilité aux conditions initiales des systèmes chaotiques [18-27]. Cette sensibilité aux conditions initiales rend le comportement chaotique de ces systèmes imprédictibles, ce qui peut être exploité pour « cacher » une information dans un signal chaotique. De plus, le déterminisme de ces systèmes permet de générer des signaux chaotiques identiques à partir des mêmes conditions initiales. En d’autres termes, il est possible d’utiliser deux systèmes chaotiques « identiques », un au niveau d’un émetteur et l’autre au niveau d’un récepteur, pour envisager ainsi une communication, exploitant le comportement chaotique de ces systèmes dans le but de masquer l’information à échanger. Par ailleurs, cette extrême sensibilité aux conditions

initiales entraine une grande difficulté à la mise en œuvre d'une synchronisation entre deux systèmes chaotiques, qui est considérée au départ comme impossible à réaliser et reste à ce jour l'épineux problème des communications basées chaos.

C'est dans ce contexte que s'inscrivent les objectifs de cette thèse portant sur "**Contribution à la Sécurité dans la Transmission des Données des Habitations Intelligentes**" et qui consiste à développer et à implémenter matériellement des cryptosystemes pour la sécurisation des données dans des RCSF homogènes et hétérogènes dans le contexte Smart Home. Pour réaliser ces objectifs, **dans un premier temps**, nous avons effectué une étude comparative de plusieurs systèmes chaotiques et hyperchaotiques dans le but de déterminer celui répondant au mieux aux exigences d'un chiffrement robuste. Notre choix s'est porté sur deux générateurs l'un hyperchaotique à base du système de Chen et l'autre chaotique basé sur le système chaotique unifié. Les deux générateurs sélectionnés génèrent des séquences binaires très intéressantes, pour les utiliser directement comme clés de chiffrement de données et qui possèdent de très bonnes propriétés statistiques confirmées par ailleurs par la batterie de tests du NIST (National Institute of Standards and Technology). **Dans un deuxième temps**, pour plus d'assurance et de validation, nous avons effectué une analyse de sécurité des données chiffrées en utilisant un masque jetable à base de l'opérateur XOR. Nous avons pris l'image et l'audio comme données à chiffrer. Sachant que le succès et la robustesse d'une méthode de cryptage se résume à sa résistance face aux diverses attaques telles que l'attaque par force brute, l'attaque statistique et l'attaque différentielle nous avons effectué des tests de sécurité les plus connus et utilisés tel que l'espace des clés, le calcul d'entropie, l'autocorrélation, les histogrammes et la sensibilité à la clé. Ces tests ont été appliqués sur différentes images de référence. **Dans un dernier temps**, après avoir présenté et expliqué la solution adoptée pour résoudre et contourner l'épineux problème de la synchronisation chaotique, qui se base essentiellement sur l'ajout d'indices de synchronisation, bien dissimilés, dans les données transmises, nous avons développé et implémenté des algorithmes, sur différentes plateformes matérielles (Microcontrôleur, Smartphone et PC embarqué), permettant la sécurisation de données échangées dans le RCSF. Les données à sécuriser englobent tout ce qui est mesures de température, d'humidité ainsi que le flux vidéo/audio temps réel.

En résumé, les principales contributions apportées dans cette thèse se résument à la conception, développement et validation expérimentale de :

- Deux cryptosystèmes symétriques basés sur deux différents générateurs robustes utilisant le chiffrement à masque jetable.
- Deux techniques de synchronisations fiables.

- Réalisation de deux plateformes matérielles de RCSF.
- Des applications interactives pour PC, PC embarqué (sous linux) et Smartphones (tournant sous Android), sous forme d'Interface Homme Machine (IHM) simples d'utilisation.

La suite de cette thèse est organisée comme suit. La première partie du premier chapitre aborde et décrit les définitions et les généralités sur l'Internet Des Objet ainsi que ses différents domaines d'applications notamment dans le contexte des maisons intelligentes. La deuxième partie présente les différentes technologies de communication ainsi que les protocoles associés. La troisième partie est consacrée aux maisons intelligentes en termes de menaces et de protection des données collectées par les capteurs ainsi que la stratégie de sécurité adoptée.

La première partie du deuxième chapitre aborde et décrit les définitions et les généralités sur la théorie du chaos, les principales propriétés des systèmes chaotiques et les différents outils utilisés pour l'étude et l'analyse de ces systèmes. La deuxième partie expose brièvement des notions de base sur le cryptage et la cryptanalyse. Dans la troisième partie, une attention particulière est réservée aux différents outils et métriques les plus utilisés pour l'évaluation des cryptosystemes.

Dans le troisième chapitre, consacré au premier cryptosysteme proposé, on commence par présenter l'architecture du générateur des clés de chiffrement hyperchaotiques proposé ainsi que les résultats de l'analyse statistique des clés hyperchaotiques générées en utilisant la batterie de test du NIST. Pour mieux valider la robustesse des clés, une analyse de sécurité différentielle et statistique est effectuée sur des données chiffrées (image) par masque jetable. Puis nous présentons l'architecture de la plateforme matérielle développée ainsi qu'une description et une analyse des performances de l'algorithme du cryptage des données proposé.

Le dernier chapitre présente et détaille notre deuxième contribution qui consiste à développer et à réaliser un cryptosystème basé sur le système chaotique unifié discrétisé pour sécuriser différentes données (température, humidité et flux audio/vidéo) au sein d'un réseau de capteur sans fil hétérogène utilisant plusieurs technologies de communication. Une analyse de sécurité des données chiffrées est effectuée. Puis une description détaillée de la plateforme matérielle réalisée ainsi qu'une analyse des performances de l'algorithme du cryptage des données proposé sont données. Enfin, la thèse est clôturée par une conclusion.

Chapitre I

L'INTERNET DES OBJETS ET SON APPLICATION DANS LES MAISONS INTELLIGENTES

I.1 Introduction

L'internet des objets (IdO) ou l'IoT désigne un réseau interconnecté de dispositifs physiques, tels que les appareils domestiques, les capteurs, les véhicules et autres objets, qui sont capable de communiquer et d'échanger des données entre eux et avec des systèmes informatiques à travers l'internet. Dans ce premier chapitre nous présentons une vue générale de l'Internet des Objets et son application dans les maisons intelligentes. En premier lieu, nous commençons par donner l'historique de l'IoT, son évolution, ses paradigmes et ses différentes architectures ainsi que ses technologies et protocoles de communication associés. En deuxième lieu, nous abordons les domaines d'application de l'IoT et plus particulièrement dans les maisons intelligentes, suivi des différents obstacles et défis auxquels l'IoT doit faire face. En fin, nous citons les différentes menaces et attaques ainsi que les stratégies de sécurisation permettant d'atténuer ces risques et assurer ainsi la protection des données privées et sensibles des utilisateurs.

I.2 L'histoire de l'IoT

L'internet des objets, initialement conceptualisé dans les années 1990, s'est rapidement développé grâce aux avancées en matière de connectivité d'internet, de capteurs bas couts, et de puissance de calcul. Cette évolution a permis une intégration plus profonde et plus large des dispositifs IoT dans la vie quotidienne, transformant des concepts autrefois futuristes en réalités pratiques au sein des foyers modernes.

En 1999, le terme internet des objets a été utilisé pour la première fois par Kevin Ashton lors d'une présentation pour Procter & Gamble (une multinationale américaine qui opère principalement dans le secteur des biens de consommation) dans le contexte de la gestion de la chaîne d'approvisionnement en utilisant la technologie RFID (Radio Frequency Identification) [29]. Au début des années 2000 et avec le développement des technologies de réseaux sans fil et les systèmes MEMs (Micro Electro Mechanical Systems), l'IoT a commencé à prendre forme. Dans les années 2010, Google a annoncé la collecte de données sur le flux de trafic à partir des capteurs des Smartphones, illustrant ainsi une application

pratique de l'IoT [30]. Depuis 2020 l'internet des objets a continué à se développer, intégrant les nouvelles technologies telles que la 5G, l'intelligence artificielle et l'apprentissage automatique, et elle s'est largement répandu dans plusieurs domaines comme la santé, l'industrie, les villes et les maisons intelligentes. Cependant la collecte des données à grande échelle soulève des préoccupations en matière de sécurité et de vie privée, incitant ainsi à la création de réglementations spécifiques telles que le Règlement General sur la Protection des Données(RGPD) en Europe [31].

I.3 Paradigme de l'IoT

Le paradigme de l'Internet des Objets repose fondamentalement sur l'idée de connecter les objets de la vie courante à Internet, permettant ainsi une interaction et une collecte de données automatisées. Les principaux concepts qui définissent ce paradigme sont les suivants:

a. Connectivité Ubiquitaire

L'IoT vise à doter une multitude d'objets de capacités de communication, leur permettant d'être connectés à Internet en permanence ou à la demande. Cela inclut des appareils allant des capteurs environnementaux aux appareils électroménagers, en passant par les véhicules et les équipements industriels [32].

b. Interopérabilité et Standards

L'internet des objets se caractérise par l'emploi de plusieurs objets hétérogènes de technologies différentes comme LoraWAN, SigFox, Bluetooth, etc. Ces objets doivent fonctionner ensemble et partager des données d'une façon cohérente, cela nécessite l'introduction d'un standard permettant l'interopérabilité entre les différents protocoles de communication et plateformes [33][34].

c. Intelligence et Automatisation

Les objets connectés sont dotés de capacités de traitement qui leur permettent de prendre des décisions autonomes basées sur les données collectées. L'introduction de l'intelligence artificielle permet de ce fait de créer un modèle prédictif, de détecter des anomalies ainsi que d'optimiser les opérations des appareils connectés. Cela conduit à des niveaux plus élevés d'automatisation. Par exemple, l'ajustement de l'éclairage et la température en fonction de la présence humaine et des conditions environnementales ou bien le déclenchement d'enregistrement vidéo en cas de détection de mouvement.

d. Sécurité et Vie Privée

Avec l'augmentation du nombre d'objets connectés, la sécurité et la protection de la vie privée deviennent un enjeu majeur [34]. Il est alors essentiel et nécessaire de mettre en place des mécanismes robustes pour protéger les données contre les accès non autorisés et garantir ainsi la confidentialité des données des utilisateurs [35][36].

I.4 Architectures de l'IoT

L'architecture de l'IoT fournit la structure nécessaire à la connectivité, à la communication et à la gestion des objets connectés. Elle permet de concevoir un modèle qui assure la sécurité, l'évolutivité et l'efficacité opérationnelle du réseau IoT. Une architecture bien conçue facilite l'intégration de différentes couches et composants, garantissant un fonctionnement harmonieux de l'IoT. L'architecture de l'Internet des Objets peut être structurée en différentes couches selon les besoins de l'application, de l'intégration des dispositifs et de la gestion des données [41]. Les architectures à trois, quatre et cinq couches sont les plus citées dans la littérature.

I.4.1. Architecture à trois couches

La figure (I.1) illustre une architecture basique de l'IoT composée de trois couches [37].

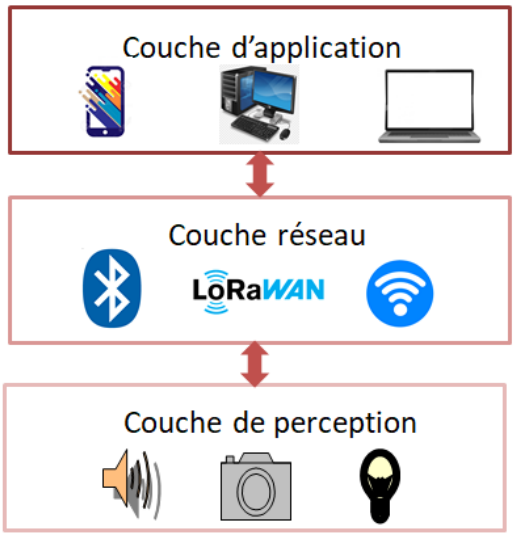


Figure I.1. Architecture à trois couches.

Cette architecture est constituée d'une :

1. Couche de perception : Comprend les capteurs et les dispositifs de collecte des données qui interagissent avec l'environnement. Cette couche est responsable de la perception physique des données.

2. Couche de réseau : représente le cœur de l'IoT, Elle assure une transmission sécurisée des données recueillies par la couche de perception vers la couche d'application et vice versa, en utilisant divers protocoles de communication (wifi, bluetooth, zigbee… etc).

3. Couche d'application : cette couche est responsable de la gestion globale de l'application. Elle traite et analyse les données collectées pour fournir des services spécifiques aux utilisateurs finaux.

I.4.2. Architecture à quatre couches

L'architecture à trois couches n'est pas suffisante vu le développement fulgurant de l'IoT et le grand nombre de données à gérer conduisant ainsi à la proposition d'une autre architecture à quatre couches [38] comme représenté sur la figure (I.2).

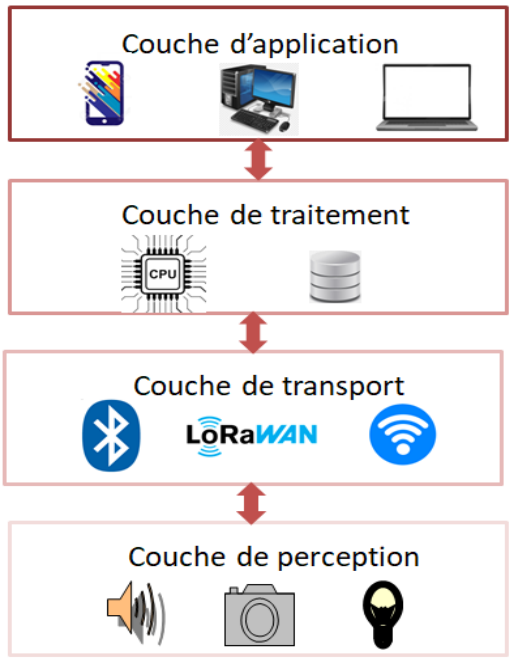


Figure I.2. Architecture à quatre couches.

Cette architecture comprend :

1. Couche de perception : Identique à celle de l'architecture à trois couches.

2. Couche de transport : Route les données des capteurs vers le processeur de la gestion des données. Elle inclut les technologies de communication filaire et sans fil.

3. Couche de traitement (aussi appelée Middleware) : Fournit une capacité de traitement des données, de gestion des dispositifs et de support logiciel pour les applications IoT.

4. Couche d'application : Identique à celle de l'architecture à trois couches.

I.4.3. Architecture à cinq couches

L'architecture à cinq couches représentée sur la figure (I.3) est constitué d'une :

1. Couche de perception : Identique aux architectures précédentes.

2. Couche de transport : Identique à l'architecture à quatre couches.

3. Couche de traitement : Identique à l'architecture à quatre couches.

4. Couche d'affaires : Analyse les données de l'application pour prendre des décisions basées sur des informations. Elle gère également l'ensemble du système IoT, y compris les applications et les services utilisateurs [42].

5. Couche d'application : Se concentre sur la livraison des applications IoT spécifiques aux exigences des utilisateurs finaux.

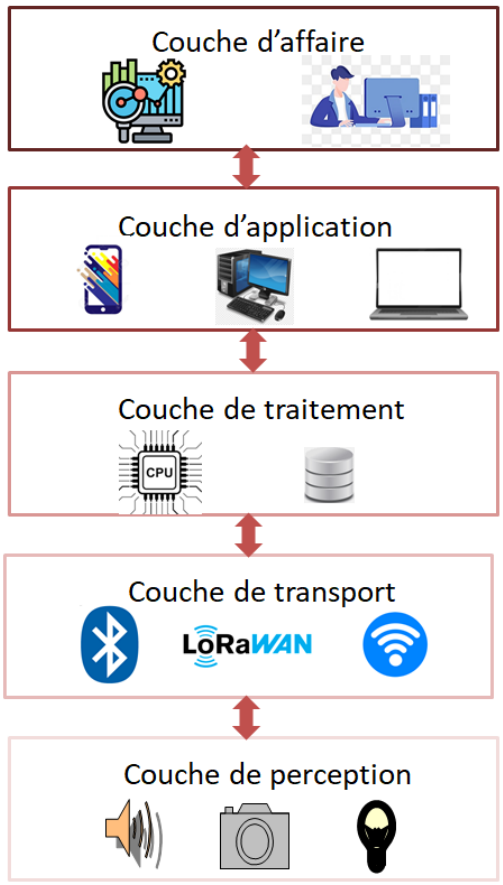


Figure I.3. Architecture à cinq couches.

Ces architectures offrent un cadre pour comprendre les composants fonctionnels de l'IoT et la manière dont ils interagissent. Elles peuvent être adaptées ou étendues en fonction des besoins spécifiques des projets IoT.

La différence fondamentale entre les architectures IoT à trois, quatre et cinq couches réside dans la manière dont les responsabilités et les fonctionnalités sont réparties à travers le système. L'architecture à trois couches est la plus simple et la plus basique, comprenant seulement les couches de perception, de réseau, et d'application. En raison de sa simplicité, elle peut être limitée en termes de gestion avancée des données, d'analyse, et de sécurité, ce qui la rend moins adaptée aux systèmes IoT complexes. Quant à l'architecture à quatre couches, qui ajoute une couche de traitement (ou middleware) entre les couches de réseau et d'application, permet une gestion des données plus avancée, une meilleure intégration des services, et une abstraction supplémentaire pour les développeurs d'applications. Cette architecture offre une plus grande flexibilité et une meilleure gestion des dispositifs et des données ce qui facilite le développement d'applications IoT plus complexes. L'architecture à cinq couches introduit une couche d'affaires au-dessus de la couche d'application, permettant une analyse des données plus approfondie, la prise de décision basée sur les données, et la gestion stratégique du système IoT dans son ensemble. C'est l'architecture la plus complexe et la plus complète, qui offre une vision globale de l'écosystème IoT, depuis la collecte de données jusqu'à la prise de décision au niveau de l'entreprise.

I.5 Domaines d'applications de l'IoT

L'Internet des Objets joue un rôle important dans divers secteurs, notamment le transport, l'agriculture, les villes intelligentes, la santé et la domotique (figure I.4) en fournissant des solutions innovantes pour des défis contemporains.

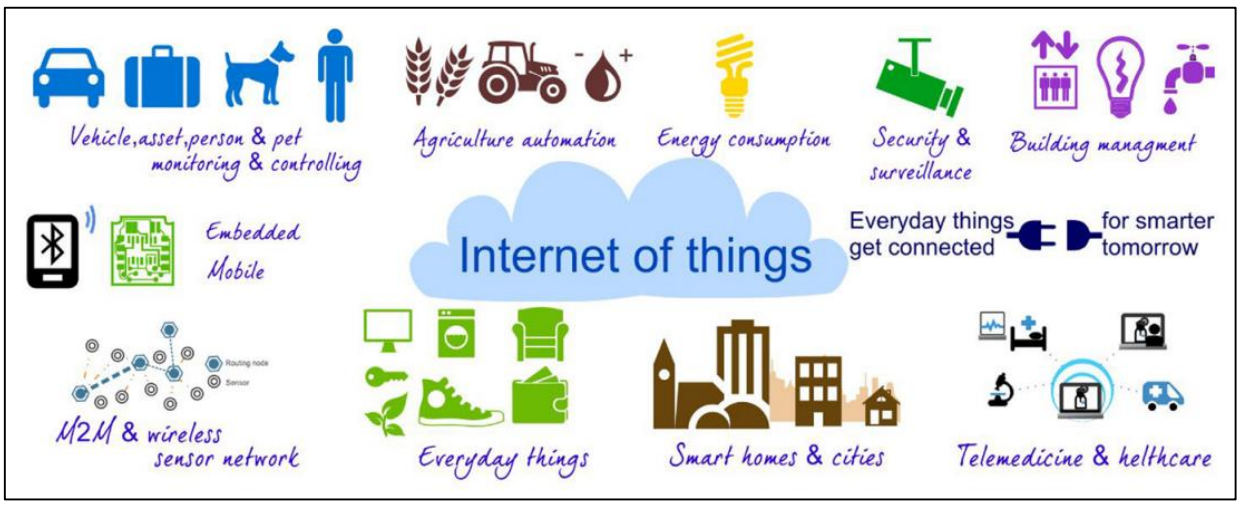


Figure I.4. Domaines d'application de l'IoT [40].

- **Transport**

Dans le secteur des transports, l'IoT offre des solutions intéressantes pour améliorer la gestion du trafic, la sécurité des véhicules et l'efficacité énergétique. Les systèmes intelligents de gestion du trafic utilisent des capteurs et des caméras pour optimiser les flux de circulation, réduire la congestion et améliorer la sécurité. Les systèmes de transport public profitent également de l'IoT pour le suivi en temps réel des véhicules, la gestion efficace des capacités et des ressources des entreprises de transport ainsi que la billetterie électronique.



Figure I.5. Application de l'IoT dans le secteur de transport [141].

- **Agriculture**

L'agriculture intelligente utilise et exploite l'IoT pour une surveillance précise des cultures, l'irrigation automatisée et la gestion du cheptel animalier, permettant une utilisation plus efficace des ressources et améliorant les rendements. Les capteurs mesurent l'humidité du sol, la température, la lumière et d'autres paramètres environnementaux nécessaires pour optimiser les conditions de croissance [40][43].

- **Villes intelligentes (Smart City)**

Les villes intelligentes intègrent l'IoT pour améliorer la qualité de vie urbaine, la durabilité et les services publics. Cela inclut l'éclairage public intelligent, la gestion des déchets, la qualité de l'air et la surveillance de l'eau, ainsi que des infrastructures de transport avancées pour une mobilité plus fluide et moins polluante.



Figure I.6. Application de l'IoT dans les villes intelligentes [142].

- **Santé**

Dans le domaine de la santé, l'IoT permet une surveillance à distance des patients, améliorant ainsi l'accès aux soins et la gestion des maladies chroniques. Les appareils portables et les capteurs à domicile collectent des données vitales en temps réel, permettant aux professionnels de la santé de suivre l'état des patients à distance et de réagir rapidement en cas de besoin ou d'urgence.



Figure I.7. Application de l'IoT dans le domaine médical [143].

- **Domotique**

La domotique utilise et exploite les avantages de l'IoT pour rendre les maisons plus confortables, sures et éco-énergétiques. Les dispositifs connectés permettent le contrôle à distance de l'éclairage, du chauffage, des appareils électroménagers et des systèmes de sécurité, offrant ainsi un confort accru tout en réduisant la consommation d'énergie.



Figure I.8. Application de l'IoT dans une maison intelligente [144].

Ces diverses applications montrent bien l'étendue et la diversité des possibilités offertes par l'IoT en transformant de manière significative la façon dont nous vivons, travaillons et interagissons avec notre environnement.

I.6 Technologies et protocoles de communication de l'Internet des Objets

I.6.1 Protocoles de Communication

Les protocoles de communication sont des ensembles de règles et de conventions qui définissent la manière dont les données sont échangées entre des dispositifs ou des systèmes informatiques. Ces protocoles déterminent la structure, le format et la séquence des messages échangés, ainsi que les actions à prendre en réponse à ces messages. Ils peuvent être utilisés pour diverses formes de communication, y compris les réseaux informatiques, les systèmes embarqués, les communications sans fil, et plus encore. La figure (I.9) illustre quelques protocoles de communication les plus connus et utilisés.

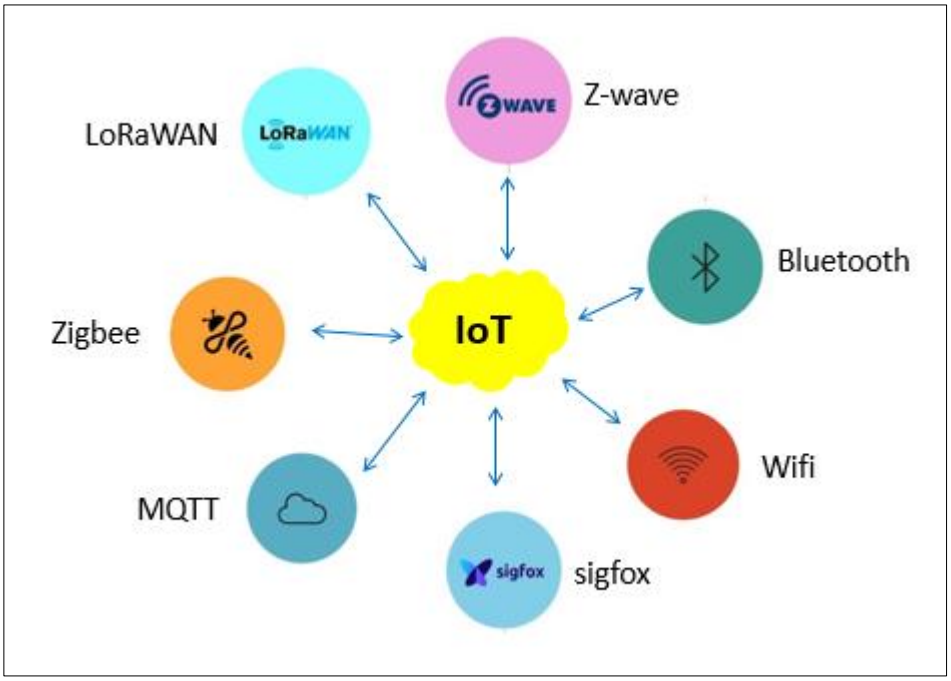


Figure I.9. Quelques protocoles de communication.

a) X-10 : Les protocoles de communication X-10 sont utilisés dans la domotique, pour contrôler différents appareils électriques dans une maison ou un bâtiment via le courant porteur en ligne (CPL). Initialement développé en 1975 par *Pico Electronics* (Écosse), le X-10 permet à des appareils électriques de communiquer entre eux en envoyant des signaux à travers le réseau électrique existant de la maison [45].

Les signaux X-10 sont transmis sous forme de paquets de données numériques qui sont superposés au courant alternatif standard. Ces signaux contrôlent l'allumage, l'extinction, la variation d'intensité des appareils électriques, et bien d'autres fonctions. Il existe différents types de modules X-10, y compris des modules récepteurs qui s'installent sur les appareils ou les prises électriques, et des modules émetteurs comme des télécommandes ou des interfaces informatiques. Ces modules peuvent être utilisés pour contrôler à distance des lumières, des appareils, des systèmes de chauffage, etc.

Les appareils X-10 sont contrôlés via un système d'adressage qui utilise des codes de maison (lettres de A à P) et des numéros d'unité (1 à 16). Cela permet de contrôler jusqu'à 256 appareils distincts dans une installation X-10.

b) Wi-Fi : Le protocole de communication Wi-Fi est un ensemble de normes définies par l'Institute of Electrical and Electronics Engineers (IEEE) sous la désignation 802.11. Ces normes régissent la communication sans fil entre appareils au sein des réseaux locaux sans fil. Le Wi-Fi permet à des appareils comme des ordinateurs, téléphones, tablettes,

imprimantes et autres équipements de se connecter à Internet ou de communiquer entre eux sans nécessiter de connexions filaires. Il existe plusieurs versions du protocole Wi-Fi, chacune avec ses propres spécifications techniques, telles que la portée, la bande de fréquence utilisée, et la vitesse de transmission de données [48].

- **802.11a** : Lancé en 1999, ce protocole opère sur la bande de fréquence de 5 GHz avec une vitesse de transmission maximale de 54 Mbps.
- **802.11b** : Également introduit en 1999, mais utilisant la bande de fréquence de 2,4 GHz pour une vitesse maximale de 11 Mbps. Cette version est largement considérée comme la première à avoir été adoptée massivement.
- **802.11g** : Introduit en 2003, il combine le meilleur des deux versions précédentes et opère sur 2,4 GHz pour des vitesses de transmission allant jusqu'à 54 Mbps.
- **802.11n (Wi-Fi 4)** : Adopté en 2009, ce protocole peut fonctionner sur les bandes de 2,4 GHz et 5 GHz, offrant des vitesses de transmission pouvant atteindre 600 Mbps grâce à la technologie MIMO (Multiple Input Multiple Output) qui utilise plusieurs antennes pour envoyer et recevoir des données.
- **802.11ac (Wi-Fi 5)** : Introduit en 2013, ce protocole opère exclusivement sur la bande de 5 GHz avec des vitesses théoriques pouvant atteindre plusieurs gigabits par seconde, grâce à l'amélioration de la technologie MIMO, connue sous le nom de MU-MIMO (Multi-User MIMO).
- **802.11ax (Wi-Fi 6)** : La dernière version en date, introduite en 2019, qui offre des améliorations significatives en termes d'efficacité, de vitesse et de couverture dans les zones à forte densité de réseaux Wi-Fi. Wi-Fi 6 est conçu pour mieux gérer les multiples connexions et offrir des vitesses plus élevées.

c) Bluetooth : Le protocole de communication Bluetooth est une technologie sans fil conçue pour le transfert de données à courte portée entre appareils. Contrairement au Wi-Fi, qui est destiné à des réseaux locaux sans fil à plus grande échelle, Bluetooth est souvent utilisé pour connecter des appareils personnels tels que téléphones, écouteurs, claviers et souris. Développé par le groupe Bluetooth Special Interest Group (SIG), le protocole Bluetooth vise à faciliter les communications entre appareils tout en maintenant une faible consommation d'énergie [28][48].Il existe plusieurs versions de Bluetooth, chacune offrant des améliorations en termes de vitesse de transfert, de portée, de sécurité, et de consommation d'énergie :

- Bluetooth 1.x: La première version de Bluetooth, relativement lente et moins sécurisée.

- Bluetooth 2.x: Offre une vitesse de transfert améliorée avec EDR (Enhanced Data Rate).

- Bluetooth 3.x + HS (High Speed) : Permet des transferts de données à haute vitesse allant jusqu'à 24Mbps.

- Bluetooth 4.x (Bluetooth Low Energy ou BLE) : Introduit le concept de Bluetooth Low Energy, offrant des communications à très faible consommation d'énergie, ainsi que la possibilité de connecter plusieurs appareils simultanément.

- Bluetooth 5.x: Augmente significativement la portée et la vitesse de transfert, et améliore la fonctionnalité BLE pour mieux servir les technologies de l'IoT.

- Bluetooth 5.1 et ultérieures : Introduisent des améliorations dans la localisation précise des appareils et la gestion de la connexion pour une communication encore plus efficace et économe en énergie.

d) Z-Wave: Le protocole Z-Wave est une technologie de communication sans fil spécialement conçue pour l'automatisation résidentielle et commerciale, notamment pour les applications de sécurité domestique, d'éclairage, de chauffage, de ventilation, de climatisation , et de gestion de l'énergie [49]. Développé à l'origine par la société danoise Zensys. Z-Wave opère sur des bandes de fréquence autour de 900 MHz. Cette bande de fréquence est moins encombrée que les bandes de 2,4 GHz utilisées par d'autres technologies sans fil comme le Wi-Fi et le Bluetooth, ce qui réduit les interférences. La portée typique d'un signal Z-Wave dans un environnement intérieur est d'environ 30 à 100 mètres, selon les obstacles et les matériaux de construction. En extérieur, cette portée peut être plus étendue. Z-Wave utilise une architecture de réseau maillé, où chaque dispositif Z-Wave agit comme un répéteur de signal. Cela permet aux signaux de s'affranchir des obstacles et d'étendre ainsi la portée du réseau.

e) LoRaWAN: Le protocole LoRaWAN (Long Range Wide Area Network) est une technologie de communication sans fil conçue pour permettre aux dispositifs à faible consommation d'énergie de communiquer sur de longues distances, ce qui la rend idéale pour les applications Internet des objets (IoT). LoRaWAN désigne l'architecture du système et la gestion du protocole de communication, tandis que LoRa (Long Range) fait référence à la technique de modulation radio sous-jacente qui permet cette communication à longue portée [44]. LoRaWAN offre une grande portée, permettant la communication

entre dispositifs et passerelles sur des distances allant de 2 à 5 km en milieu urbain dense et jusqu'à 15 km ou plus en milieu rural, avec une très faible consommation d'énergie. Ses dispositifs sont conçus pour fonctionner avec des batteries pendant plusieurs années, rendant la technologie idéale pour des applications où l'accès à l'alimentation électrique est limité ou inexistant. De plus LoRaWAN utilise une architecture de réseau en étoile, où les dispositifs communiquent avec une ou plusieurs passerelles qui retransmettent les messages à un serveur de réseau central. Cette structure supporte des millions de dispositifs sur un seul réseau.

f) 6LoWPAN: Le protocole 6LoWPAN (IPv6 over Low-Power Wireless Personal Area Networks) est un protocole de communication qui permet d'acheminer des paquets IPv6 sur des réseaux sans fil à faible consommation d'énergie, tels que les réseaux de capteurs sans fil (RCSF) et les réseaux de maillage à faible puissance (LPWAN) [47][48]. Puisque les trames dans les réseaux sans fil à faible puissance ont souvent des tailles de paquet limitées, 6LoWPAN utilise des techniques de compression d'en-tête pour réduire la surcharge de données et permettre l'envoi efficace de paquets IPv6 sur de tels réseaux. Le protocole 6LoWPAN supporte les topologies de réseau maillé, ce qui signifie que les appareils peuvent communiquer directement les uns avec les autres, même s'ils ne sont pas directement connectés à un point d'accès central.

g) Sigfox : Le Sigfox est une technologie de communication sans fil à faible débit, à faible consommation d'énergie, et à longue portée, conçue spécifiquement pour l'Internet des Objets (IoT)[40]. Le protocole sigfox est caractérisé par un faible débit et une longue portée. La portée de Sigfox est significativement plus grande que celle des technologies sans fil traditionnelles, ce qui permet aux appareils de communiquer sur de longues distances, souvent jusqu'à plusieurs kilomètres. Sigfox utilise un réseau à faible puissance et à large bande passante (LPWAN), dans lequel les appareils communiquent avec des passerelles Sigfox qui transfèrent les données vers le cloud. Cette architecture permet une couverture étendue avec relativement peu de points d'accès.

h) Zigbee : Le Zigbee est un protocole de communication sans fil à faible consommation d'énergie, à courte portée d'environ 10 à 100 mètres en intérieur et jusqu'à 1 000 mètres en extérieur, en fonction des conditions environnementales et des obstacles, et à haut débit, spécialement conçu pour les réseaux de capteurs sans fil et les applications de domotique [48].Le Zigbee fonctionne sur la bande de fréquences 2,4 GHz et utilise une topologie de réseau maillé, ce qui signifie que chaque appareil Zigbee peut agir comme un routeur pour

étendre la portée du réseau. Cela permet une couverture étendue sans dépendre d'une portée directe entre les appareils et la passerelle.

II.6.1 Protocoles de Données

a) MQTT (Message Queuing Telemetry Transport) : Le MQTT (Message Queuing Telemetry Transport) est un protocole de messagerie de type publish-subscribe où les appareils peuvent publier des messages sur des sujets, et d'autres appareils peuvent s'abonner à ces sujets pour recevoir les messages pertinents. Cela permet une communication décentralisée et évite le besoin de connexion directe entre les appareils. Ce protocole est léger et efficace en termes de bande passante et de ressources système, il est conçu pour les applications IoT et les systèmes de communication machine à machine (M2M) [62][64]. Le MQTT prend en charge trois niveaux de qualité de service (*Quality Of Service* QoS) pour garantir la fiabilité de la livraison des messages :

- QoS 0 : au plus une fois (pas de garantie de livraison).
- QoS 1 : au moins une fois (garantie de livraison mais risque de duplication).
- QoS 2 : exactement une fois (garantie de livraison sans duplication).

MQTT est utilisé dans une grande variété d'applications IoT, y compris la surveillance à distance, la télémétrie, la domotique, les villes intelligentes, l'agriculture connectée, et bien plus encore.

b) CoAP (Constrained Application Protocol) : Le protocole CoAP (Constrained Application Protocol) est un protocole de communication conçu spécifiquement pour les appareils avec des ressources limitées en termes de puissance de calcul, de mémoire et de bande passante tels que les capteurs et les actionneurs dans les réseaux IoT [63][64].CoAP adopte un modèle client-serveur similaire à HTTP, où les appareils CoAP interagissent en tant que clients ou serveurs. Il utilise UDP (User Datagram Protocol) comme protocole de transport par défaut, ce qui permet des échanges de données rapides et efficaces. Aussi il utilise des URIs (*Uniform Resource Identifiers*) pour identifier les ressources sur les serveurs CoAP, et il utilise des méthodes similaires à HTTP telles que GET, POST, PUT et DELETE pour interagir avec ces ressources.

c) AMQP (Advanced Message Queuing Protocol) : Le protocole AMQP (Advanced Message Queuing Protocol) est un protocole de messagerie ouvert basé sur un modèle de communication orienté message, où les applications envoient et reçoivent des messages de manière asynchrone via des files d'attente (queues) gérées par un courtier (broker) de messages, conçu pour la communication entre des systèmes de messagerie distants [65].

Il est indépendant du protocole de transport sous-jacent, ce qui signifie qu'il peut être implémenté sur différents protocoles, y compris TCP/IP, WebSocket et d'autres protocoles de transport.

d) M2M (Machine to Machine) : La communication M2M (Machine to Machine) fait référence à l'échange de données ou d'informations entre des dispositifs électroniques ou des machines sans nécessiter d'intervention humaine directe. Dans un réseau M2M, les machines interagissent entre elles pour effectuer des tâches spécifiques ou pour transmettre des informations [66]. Les dispositifs M2M comprennent souvent des capteurs qui collectent des données à partir de l'environnement et des actionneurs qui agissent en fonction de ces données. Par exemple, un capteur de température peut collecter des données et les transmettre à un thermostat qui ajuste ensuite la température en conséquence. Les dispositifs M2M utilisent différents protocoles de communication pour échanger des données. Cela peut inclure des protocoles sans fil comme Wi-Fi, Bluetooth, Zigbee, LoRa, ou des protocoles filaires comme Ethernet ou Modbus. Des protocoles spécifiques, comme MQTT, CoAP ou AMQP, peuvent également être utilisés pour des scénarios M2M spécifiques

e) XMPP (Extensible Messaging and Presence Protocol) : Le protocole XMPP (Extensible Messaging and Presence Protocol) est un protocole ouvert et extensible pour la messagerie instantanée et la présence en ligne. Il a été développé à l'origine sous le nom de Jabber au sein de la communauté de développeurs Open Source et a été normalisé par l'IETF (Internet Engineering Task Force). XMPP est basé sur XML (eXtensible Markup Language) et est conçu pour permettre la communication en temps réel entre différentes entités sur un réseau, comme des utilisateurs, des serveurs de messagerie, des services de présence, etc [67][68].

XMPP est largement utilisé dans une gamme d'applications, y compris les services de messagerie instantanée grand public, les solutions de collaboration en entreprise, les réseaux sociaux décentralisés, les systèmes de notification en temps réel, etc. Sa conception ouverte et extensible en fait un bon choix pour les développeurs qui cherchent à créer des applications de communication personnalisées.

Le tableau (1.1) présente une comparaison entre les différentes technologies de communication sans fil utilisées dans l'IoT.

Protocole	Bluetooth	ZigBee	Z-Wave	LoRaWAN	Sigfox	Wi-Fi (802.11ac)
Débits	2 Mbps	250 kbps	100 kbps	0,3-50 kbps	1 Mbps	433 Mbps - 1300 Mbps
Sécurité	AES 128 bits	AES 128 bits	AES 128 bits	AES 128 bits	Partiellement adressée	WEP- WPA (AES 128 bits)
Fréquences	2,4 GHz	2,4 GHz	868 MHz (EU) 908 MHz (USA)	868 MHz (EU) 915 MHz (USA)	868 MHz (EU) 902 MHz (USA)	5 GHz
Portée	10 - 100 m	10 - 100 m	30 m (à l'intérieur) et 100 m (à l'extérieur)	20 km (zone rurale) et 8 km (zone urbaine)	50 km (zone rurale) et 10 km (zone urbaine)	35 m (à l'intérieur) et 300 m (à l'extérieur)
Topologies	Réseau en étoile ou bus	Réseau en étoile ou mesh	Réseau mesh	Réseau en étoile	Réseau en étoile	Infrastructure ou Ad hoc

Tableau.1.1 Comparaison entre différentes technologies de communication sans fil utilisées dans l'IoT [50].

1.7 Les défis de l'IoT

Les défis de l'Internet des Objets (IoT) sont multiples et reflètent la complexité de ce domaine en rapide évolution. Dans ce qui suit on cite les principaux défis à relever ainsi que les obstacles à surmonter dans le développement et le déploiement de solutions IoT efficaces et sécurisées [48].

a) Standardisation

La standardisation est un sujet essentiel dans l'IoT. En effet, avec la présence de nombreux fabricants, de plateformes et de technologies, l'absence de standards universels pose un défi majeur. Cela peut mener à des problèmes de compatibilité et limiter l'intégration entre différents dispositifs et systèmes. Ainsi cette situation engendre une fragmentation du marché ce qui se traduit par une augmentation des coûts et de complexité pour les développeurs et les consommateurs.

b) Scalabilité

La scalabilité est une caractéristique clé de l'internet des objets, qui doit pouvoir prendre en charge un grand nombre d'appareils connecté simultanément. Les infrastructures actuelles, tant en termes de matériel que de logiciel, doivent être capables de s'adapter à ce grand volume de données sans compromettre les performances. Cette exigence entraine la nécessité de trouver des solutions adaptées pour gérer les données, les analyser et les stocker efficacement.

c) Interopérabilité

L'interopérabilité fait référence à la capacité de systèmes et dispositifs divers à communiquer et travailler ensemble de manière transparente. C'est un défi majeur en raison de la diversité des protocoles de communication et des standards. Cela nécessite le développent des interfaces et des protocoles communs qui facilitent l'intégration et la communication entre dispositifs.

d) Sécurité

La sécurité est un des défis les plus critiques de l'IoT, incluant la protection des données, la sécurisation des communications, et la gestion de l'identité et de l'accès. Les dispositifs IoT sont souvent la cible de cyberattaques en raison de leur sécurité insuffisante. Il est impératif de concevoir des dispositifs avec la sécurité en tête, en utilisant le chiffrement, les mises à jour sécurisées et les pratiques de développement sécurisé.

e) Hétérogénéité

L'IoT englobe une grande variété de technologies, dispositifs, systèmes d'exploitation et applications. Cette hétérogénéité rend difficile la gestion et l'analyse des données collectées des sources diverses. Ce qui nécessite l'utilisation de middleware et de plateformes capables de traduire et d'intégrer des données hétérogènes pour une utilisation et une analyse cohérentes.

f) Consommation d'Énergie

De nombreux dispositifs IoT sont alimentés généralement par des batteries et déployés dans des endroits difficiles d'accès. La consommation d'énergie est donc un défi majeur, surtout pour garantir une longue durée de vie sans maintenance fréquente. Ce qui mène à se concentrer sur l'optimisation de l'efficacité énergétique à travers des protocoles de communication à faible puissance et des technologies de récolte d'énergie.

Ces défis soulignent l'importance de la recherche et du développement continus dans le domaine de l'IoT pour surmonter ces obstacles. La collaboration entre les industries, les universités et les gouvernements est également cruciale pour établir des normes communes et des meilleures pratiques qui favorisent l'innovation et la sécurité dans l'écosystème IoT.

1.8 Les maisons intelligentes (Smart Homes)

I.8.1 Applications des IoT dans les maisons intelligentes

Dans les maisons intelligentes, l'IoT offre une multitude d'applications qui améliorent la vie quotidienne des habitants. Tout d'abord, les appareils domestiques traditionnels tels que les thermostats, les ampoules, les serrures de porte et les caméras de sécurité peuvent être équipés de capteurs et de connectivité Internet, leur permettant ainsi de communiquer entre eux et avec les utilisateurs via des applications mobiles ou des interfaces web. Les applications de l'Internet des Objets dans les maisons intelligentes sont diverses et visent principalement à améliorer l'efficacité énergétique, la sécurité et le confort des habitations.

a) Gestion de l'Énergie

La gestion de l'énergie vise à réduire la consommation et les couts, par l'emploi des capteurs pour détecter la présence des personnes et ajuster automatiquement les appareils comme l'éclairage, le chauffage, la ventilation et la climatisation. Ainsi elle permet la surveillance de la consommation en permettant aux habitants de visualiser en temps réel leur utilisation de l'énergie, cela les aides à contrôler et à adapter leur comportement afin de réduire au maximum leur consommation.

b) Sécurité Domestique

Les systèmes de sécurité IoT offrent une surveillance en temps réel grâce à des caméras de surveillance connectées, des serrures intelligentes, des détecteurs de mouvement et des systèmes d'alarme qui peuvent être contrôlés à distance via des smartphones ou d'autres appareils [59]. Ainsi en cas de détection de fumée ou de monoxyde de carbone le système de sécurité domotique permet de déclencher immédiatement les alarmes et les alertes.

c) Confort et Commodité

L'IoT, dans les maisons intelligentes, inclut des appareils tels que des assistants vocaux, des éclairages intelligents, des stores automatiques et des systèmes de divertissement connectés qui peuvent être contrôlés par la voix ou via des applications, créant une expérience utilisateur personnalisée et confortable.

d) Santé et Bien-être

Les dispositifs IoT peuvent surveiller les conditions environnementales telles que la qualité de l'air et l'humidité pour plus de confort et bien-être. Aussi une maison intelligente offre aux personnes âgées, aux personnes à mobilité réduite et ceux qui souffrent de maladie une certaine autonomie en suivant et en diagnostiquant leur état de santé pour la prévention des problèmes et la prise en charge rapide.

e) Maintenance et Gestion des Appareils

La maintenance prédictive grâce à l'IoT permet de surveiller l'état des appareils ménagers et des systèmes pour prévenir les pannes avant qu'elles ne surviennent, grâce à l'analyse des données collectées en temps réel. Elle a pour avantages la réduction des coûts de maintenance et la prolongation de la durée de vie des appareils.

I.8.2 Composition d'une maison intelligente

Une maison intelligente, c'est un peu comme une maison qui a son propre cerveau pour gérer tout ce qui s'y passe, de la température à la sécurité (figure (I.10)). Elle est généralement composée de :



Figure I.10. Composition d'une maison intelligente.

a) Unité de Gestion Centrale (Le Cerveau)

L'unité de gestion centrale est le cerveau de la maison intelligente. Elle coordonne tous les dispositifs connectés de la maison. Souvent, cette unité est un hub ou une passerelle qui

permet la communication entre la personne (via un smartphone, une tablette ou un ordinateur) et les différents appareils de la maison. L'unité de gestion est compatible avec différents protocoles de communication tels que Zigbee, Z-wave, Wifi, Bluetooth, etc., ce qui permet de communiquer avec une large gamme d'appareils connectés.

b) Capteurs

Les capteurs sont les éléments clés d'une maison intelligente. Ils collectent des informations sur l'environnement. Il y a plusieurs types de capteurs :

- Capteurs de mouvement : Détectent la présence des personnes dans une pièce.

- Capteurs de température : Mesurent la température de l'air ambiant.

- Capteurs d'humidité : Pour suivre le taux d'humidité dans l'air.

- Capteurs de lumière : Détectent la quantité de lumière naturelle pour ajuster l'éclairage ou les stores.

- Capteurs de qualité de l'air : Mesurent les niveaux de CO2 ou d'autres polluants dans l'air.

c) Actionneurs

Les actionneurs sont des composants essentiels dans une maison intelligente, ils permettent de transformer les signaux de control reçus de l'unité de gestion centrale en action physique. Par exemple :

- Thermostats intelligents : Ajustent le chauffage ou la climatisation automatiquement en fonction des préférences des habitants, des conditions météorologiques et d'autres facteurs.

- Serrures intelligentes : Verrouillent ou déverrouillent les portes à distance à travers une application mobile ou par contrôle vocal.

- Actionneurs d'éclairage : contrôlent l'allumage, l'extinction et le changement de l'intensité ou la couleur de l'éclairage. Ils peuvent être activés manuellement ou automatiquement en réponse à un déclencheur come le mouvement ou autres.

- Prises intelligentes : Permettent de contrôler l'alimentation des appareils branchés.

d) Connectivité

La connectivité assure la communication entre l'unité de gestion centrale, les capteurs, et les actionneurs. Elle peut être assurée par différents protocoles et technologies, comme le Wi-Fi, le Zigbee, le Z-Wave, ou le Bluetooth. Chacun a ses avantages en termes de portée, de consommation d'énergie, et de capacité de connexion.

e) Interface Utilisateur

L'interface utilisateur est la façon dont vous interagissez avec votre maison intelligente. Cela peut être fait à travers des applications sur votre smartphone ou tablette, par commande vocale via des assistants comme Alexa ou Siri, ou même par des interfaces web.

Tous ces éléments travaillent ensemble afin de créer un système de maison intelligente qui facilite la vie en automatisant des tâches quotidiennes, en économisant de l'énergie et en augmentant la sécurité du domicile. Le plus important est de s'assurer que tous ces dispositifs sont compatibles et peuvent communiquer efficacement entre eux pour créer un écosystème harmonieux.

I.8.3 Les réseaux de capteurs sans fil dans les maisons intelligentes

Les réseaux de capteur sans fils (RCSF) sont un ensemble de capteurs autonomes pour surveiller et collecter des données environnementales. Dans les maisons intelligentes ils jouent un rôle important dans l'automatisation et la gestion intelligente de l'habitat. Ces réseaux collectent des données à partir de l'environnement domestique et communiquent avec les systèmes de contrôle pour prendre des décisions automatisées, améliorant ainsi l'efficacité, le confort, la sécurité, et la gestion de l'énergie.

a) Fonctionnement des Réseaux de Capteurs

Un réseau de capteurs est composé de nœuds de capteur. Chaque nœud comprend un ou plusieurs capteurs, une unité de traitement (microcontrôleur), un module de communication sans fil et une source d'alimentation. Les capteurs mesurent divers paramètres environnementaux tels que la température, l'humidité, la qualité de l'air, la présence de personnes, les niveaux de lumière et d'autres paramètres. Ces données sont ensuite transmises vers l'unité de traitement. La communication au sein des réseaux de capteurs s'effectue via des protocoles de communication sans fil adaptés aux besoins spécifiques de la maison intelligente, tels que Zigbee, Z-Wave, Wi-Fi, Bluetooth Low Energy (BLE), et LoRaWAN. Ces protocoles sont choisis en fonction de leur portée, consommation d'énergie, bande passante, et sécurité.

b) Défis des réseaux de capteurs sans fils

Les réseaux de capteurs sans fils présentent plusieurs défis qui peuvent affecter leur efficacité. Le problème majeur de ces réseaux dans une maison intelligente est qu'ils sont vulnérables aux attaques comme l'écoute clandestine et les attaques par déni de service. Les données collectées peuvent être sensibles et nécessitant une protection adéquate.

Il y a d'autres défis tels que la consommation d'énergie, l'interopérabilité et la standardisation.

I.9. Menaces et Attaques dans les maisons intelligentes

Les maisons intelligentes avec toutes leurs commodités, sont aussi sujettes à diverses menaces et attaques. Cela est dû à leur connectivité accrue et à la quantité de données personnelles qu'elles gèrent.

I.9.1. Écoute clandestine (Eavesdropping)

L'écouté clandestine, ou eavesdropping en anglais, est une menace commune dans les communications électroniques où les attaquants peuvent écouter clandestinement pour collecter des données personnelles [51][56]. Dans ce qui suit on cite quelques menaces et attaques associées à l'écouté clandestine :

- Interception de données non chiffrées : Lorsque les données ne sont pas chiffrées pendant leur transmission sur un réseau, un attaquant peut simplement écouter le trafic réseau pour capturer les informations sensibles telles que des mots de passe, des informations de carte de crédit, des messages personnels, etc.
- Sniffing réseau : Les attaquants peuvent utiliser des outils de « sniffing » réseau pour capturer et analyser le trafic réseau. Ces outils peuvent être utilisés sur des réseaux locaux câblés ou sans fil pour intercepter des paquets de données échangés entre les périphériques.
- Attaques Man-In-The-Middle (MITM) : L'attaquant peut se positionner entre les parties légitimes de la communication et intercepter toutes les données qui passent à travers lui. L'attaquant peut relayer les données après les avoir capturées, ce qui donne l'impression aux parties légitimes qu'elles communiquent directement entre elles.
- Écoute de communication sans fil : Dans les environnements sans fil, les attaquants peuvent capturer les signaux radio émis par les appareils et écouter les communications. Cela inclut les réseaux Wi-Fi, Bluetooth, Near Field Communication (NFC), etc.
- Ecoute des conversations téléphoniques : Les conversations téléphoniques peuvent être interceptées par divers moyens, y compris l'interception des signaux radio des téléphones mobiles ou la compromission des réseaux téléphoniques.
- Écoute électronique : Les attaquants peuvent installer des dispositifs d'écoute dans des locaux physiques pour intercepter les conversations audio ou surveiller les échanges de données électroniques.

I.9.2. Denial of Service (DoS)

Une attaque par Denial of Service (DoS) vise à rendre un réseau ou un appareil indisponible pour ses utilisateurs prévus, souvent en le surchargeant de requêtes. Dans une maison intelligente, cela pourrait saturer la passerelle ou le hub central avec des demandes inutiles, rendant impossible le contrôle des appareils intelligents. Cela peut aller de l'incapacité à allumer les lumières à ne pas pouvoir verrouiller les portes, créant non seulement des désagréments mais aussi des risques de sécurité [46][55]. On cite quelques formes courantes d'attaques par déni de service :

- Attaques de surcharge réseau : Les attaquants envoient un grand volume de trafic vers la cible, ce qui sature la bande passante du réseau et rend difficile voire impossible l'accès aux services pour les utilisateurs légitimes.
- Attaques par amplification : Les attaquants exploitent des services mal configurés pour envoyer des réponses disproportionnées à des requêtes légitimes, amplifiant ainsi le volume de trafic dirigé vers la cible. Les attaques par amplification peuvent utiliser des protocoles comme DNS (Domain Name System), NTP (Network Time Protocol), SNMP (Simple Network Management Protocol), etc.
- Attaques par épuisement des ressources : Les attaquants exploitent les vulnérabilités dans les systèmes cibles pour épuiser leurs ressources, telles que le CPU, la mémoire, la bande passante ou les connexions réseau. Par exemple, une attaque par synflood inonde un serveur de requêtes de synchronisation pour épuiser ses ressources de connexion.
- Attaques d'application : Les attaquants ciblent des applications spécifiques en envoyant des requêtes malveillantes ou en exploitant des vulnérabilités logicielles pour épuiser les ressources du serveur, entraînant une indisponibilité du service.
- Attaques distribuées par déni de service (DDoS) : Dans une attaque DDoS, les attaquants coordonnent un grand nombre de machines zombies (ordinateurs compromis) pour cibler simultanément la victime, augmentant ainsi la puissance de l'attaque. Les attaques DDoS sont souvent plus difficiles à contrer en raison de leur ampleur.

I.9.3. Hijacking (Détournement)

Le Hijacking fait référence à la prise de contrôle non autorisée d'un appareil dans la maison intelligente. Un attaquant peut, par exemple, prendre le contrôle d'un thermostat et modifier les réglages de température, ou pire, déverrouiller des serrures intelligentes pour permettre l'accès physique à la maison. Le Hijacking d'appareils peut avoir des conséquences

allant de la simple nuisance à des menaces sérieuses pour la sécurité personnelle [54]. Le Hijacking prend plusieurs formes telles que :

- Détournement de session TCP/IP : Dans cette forme d'attaque, un attaquant intercepte les paquets réseau échangés entre deux parties légitimes et usurpe l'identité d'une des parties pour prendre le contrôle de la session. Par exemple, dans une attaque de type TCP/IP hijacking, un attaquant peut capturer une séquence de paquets TCP, usurper l'adresse IP de l'une des parties légitimes, et continuer la communication sans que les parties légitimes ne se rendent compte du détournement.
- Détournement de session HTTP : Lors d'une attaque de détournement de session HTTP, un attaquant peut voler ou manipuler les cookies d'authentification d'un utilisateur légitime afin de s'authentifier en tant qu'utilisateur et d'accéder à des informations privilégiées ou sensibles. Cela peut être réalisé via des techniques telles que le sniffing réseau, le phishing, l'injection de code malveillant, etc.
- Détournement DNS : Dans une attaque de détournement DNS, un attaquant manipule les réponses DNS pour rediriger le trafic d'un site légitime vers un site malveillant contrôlé par l'attaquant. Cela peut être utilisé pour des attaques de phishing ou pour rediriger le trafic vers des serveurs compromis afin de voler des informations sensibles.
- Détournement de session SSL/TLS : Les attaquants peuvent compromettre la confidentialité et l'intégrité des communications sécurisées en détournant les sessions SSL/TLS. Cela peut être réalisé en utilisant des certificats frauduleux, en exploitant des vulnérabilités dans les implémentations SSL/TLS, ou en effectuant des attaques de type Man-in-the-Middle (MITM) pour intercepter et manipuler le trafic chiffré.

I.9.4. Attaques Sinkhole et Wormhole

Sinkhole et Wormhole sont deux types d'attaques réseau dans lesquelles un attaquant cherche à manipuler le trafic réseau pour son propre bénéfice. Voici une explication de chaque type d'attaque :

- **Attaque de Sinkhole (puits absorbant) :** L'attaque de Sinkhole implique la création d'un faux point d'ancrage (sinkhole) sur un réseau. Cela peut se faire en compromettant un nœud de réseau existant ou en introduisant un nœud malveillant. Une fois que le sinkhole est établi, il attire le trafic réseau légitime vers lui. Cela peut être réalisé en modifiant les tables de routage ou en diffusant de fausses informations de routage. L'objectif de cette attaque peut être de capturer et d'analyser le trafic réseau pour voler des données

sensibles, d'intercepter des communications ou de mener d'autres activités malveillantes [53][57].

- **Attaque de Wormhole (tunnel de ver) :** Dans une attaque de Wormhole, un attaquant crée un canal de communication direct entre deux parties du réseau distantes, contournant ainsi les protocoles de sécurité et les contrôles de sécurité existants. Ce tunnel de ver est généralement créé en établissant une connexion directe entre deux points du réseau via un réseau alternatif, tel qu'Internet, ou en exploitant des vulnérabilités dans les protocoles de communication pour créer un chemin direct. L'objectif de cette attaque peut être de contourner les pare-feux, les filtres de contenu, ou d'autres contrôles de sécurité pour permettre à l'attaquant d'exploiter des vulnérabilités dans le système cible, de voler des données ou de compromettre l'intégrité du système [52][58].

Ces deux types d'attaques peuvent être utilisés dans le cadre d'attaques plus complexes visant à compromettre la sécurité d'un réseau ou d'un système. Pour se protéger contre ces attaques, il est important de mettre en œuvre des mesures de sécurité telles que l'authentification, le chiffrement des communications, la surveillance du trafic réseau pour détecter les activités suspectes, et la mise à jour régulière des logiciels pour corriger les vulnérabilités. De plus, il est important de mettre en place des politiques de sécurité robustes et de sensibiliser les utilisateurs aux risques de sécurité associés à l'utilisation des réseaux informatiques.

La riposte contre ces attaques exige une vigilance constante, l'emploi de pratiques de sécurité solides, et une mise à jour régulière des appareils et logiciels afin de remédier aux éventuelles vulnérabilités, ainsi que la mise en œuvre d'une authentification robuste, peut également aider à renforcer la sécurité des maisons intelligentes.

I.10. Stratégies de sécurisation des maisons intelligentes

I.10.1 L'authentification

L'authentification est le processus de vérification de l'identité d'un utilisateur ou d'un système pour s'assurer qu'il est légitime et autorisé à accéder à des ressources ou à des informations spécifiques. C'est un élément crucial de la sécurité de l'information et est largement utilisé dans divers contextes, y compris les systèmes informatiques, les réseaux, les applications en ligne, etc [60]. L'authentification peut se faire par différents moyens :

1. Identifiant : Un identifiant est une information unique attribuée à un utilisateur, tel qu'un nom d'utilisateur, une adresse e-mail, un numéro d'identification, etc. Il est utilisé pour distinguer les utilisateurs les uns des autres dans un système.

2. Mot de passe : Un mot de passe est une chaîne de caractères secrète associée à un identifiant et utilisé pour vérifier l'identité d'un utilisateur. Les mots de passe sont généralement choisis par l'utilisateur lors de la création de son compte et doivent être gardés confidentiels pour des raisons de sécurité.

3. Facteurs d'authentification : Les facteurs d'authentification sont les éléments utilisés pour vérifier l'identité d'un utilisateur. Les trois principaux facteurs d'authentification sont :

- Facteur de connaissance : Comme un mot de passe, une phrase secrète ou une réponse à une question de sécurité.

- Facteur de possession : Comme un jeton d'authentification, une carte à puce, ou un appareil mobile utilisé pour générer des codes à usage unique.

- Facteur d'inhérence : Comme une empreinte digitale, un scan rétinien, ou d'autres caractéristiques biométriques uniques à l'individu.

4. Authentification à plusieurs facteurs : L'authentification à plusieurs facteurs combine deux ou plusieurs facteurs d'authentification pour renforcer la sécurité. Par exemple, un système peut exiger à la fois un mot de passe et un code de vérification envoyé via SMS pour authentifier un utilisateur.

5. Protocoles d'authentification : Les protocoles d'authentification définissent les méthodes et les étapes utilisées pour vérifier l'identité d'un utilisateur. Des protocoles couramment utilisés incluent OAuth, OpenID, SAML, LDAP, Kerberos, etc.

6. Gestion des identités et des accès : La gestion des identités et des accès est le processus de gestion des identités utilisateurs, des autorisations d'accès et des politiques de sécurité dans un système informatique. Cela comprend la création, la suppression et la mise à jour des comptes d'utilisateur, ainsi que la définition des droits d'accès aux ressources.

L'authentification efficace est essentielle pour prévenir l'accès non autorisé aux systèmes et aux données sensibles. Il est important de choisir des méthodes d'authentification appropriées en fonction des besoins de sécurité spécifiques d'un système ou d'une application, tout en tenant compte de la convivialité pour les utilisateurs finaux.

I.10.2 La fraîcheur des données

Cela se réfère à la garantie que les données utilisées ou manipulées sont à jour et n'ont pas été altérées où manipulées depuis leur dernière utilisation. Les données peuvent être horodatées pour enregistrer le moment où elles ont été créées ou modifiées pour garantir leur

fraîcheur. Les systèmes de contrôle de version sont souvent utilisés dans le développement logiciel pour suivre les modifications apportées aux fichiers et garantir que seules les versions les plus récentes sont utilisées. En plus, dans les environnements distribués, des mécanismes de synchronisation sont utilisés pour s'assurer que les différentes instances des données sont cohérentes et à jour.

1. Actualisation régulière : Pour maintenir la fraîcheur des données, il est important de les actualiser régulièrement en fonction de la fréquence à laquelle les informations changent dans le système ou dans l'environnement externe. Cela peut nécessiter des processus automatisés de collecte, de mise à jour et de synchronisation des données.

2. Latence des données : La latence des données fait référence au temps écoulé entre la création ou la modification des données dans le système source et leur disponibilité dans le système de destination. Réduire la latence des données est souvent essentiel pour maintenir la fraîcheur des données, en particulier dans les applications nécessitant des informations en temps réel.

3. Cohérence des données : En plus d'être à jour, il est également important que les données soient cohérentes et exactes. Cela signifie que les données doivent être correctement synchronisées entre les différentes sources et systèmes pour éviter les incohérences et les erreurs.

4. Communication en temps réel : Dans certains cas, la fraîcheur des données nécessite une communication en temps réel entre les différents composants du système. Cela peut être réalisé à l'aide de technologies telles que les flux de données en continu, les systèmes de messagerie, les websockets, etc.

5. Indicateurs de fraîcheur : Il peut être utile d'avoir des indicateurs de fraîcheur des données pour suivre et évaluer la pertinence des informations fournies. Ces indicateurs peuvent inclure des horodatages de dernière mise à jour, des métriques de latence, des alertes en cas de retard significatif, etc.

I.10.3 La confidentialité

Dans le domaine de la sécurité des données, la confidentialité est un pilier fondamental qui vise à garantir que les données sensibles sont protégées contre tout accès non autorisé ou toute divulgation involontaire. Cela nécessite une combinaison de techniques de chiffrement, de contrôle d'accès, de gestion des clés, de politiques de confidentialité et de conformité réglementaire pour garantir une protection efficace des données sensibles [61].

1. Cryptage : Le chiffrement est une technique essentielle pour garantir la confidentialité des données. Il consiste à convertir les données en un format illisible sans la clé de chiffrement correspondante. Ainsi, même si les données sont compromises, elles restent inutilisables pour les personnes non autorisées.

2. Contrôle d'accès : Le contrôle d'accès est utilisé pour limiter l'accès aux données sensibles uniquement aux utilisateurs autorisés. Cela implique la mise en place de politiques de sécurité et de mécanismes techniques tels que les listes de contrôle d'accès (ACL), les rôles et les permissions, l'authentification, etc.

3. Masquage et pseudonymisation : Pour réduire les risques liés à la divulgation accidentelle de données sensibles, des techniques telles que le masquage des données (remplacement des données sensibles par des données factices) et la pseudonymisation (remplacement des identifiants par des identifiants fictifs) peuvent être utilisées.

4. Gestion des clés : Les clés de chiffrement jouent un rôle critique dans la protection de la confidentialité des données. Il est essentiel de mettre en œuvre des pratiques de gestion des clés robustes pour garantir la sécurité et la disponibilité des clés de chiffrement, ainsi que pour limiter l'accès aux personnes autorisées.

5. Politiques de confidentialité : Les organisations doivent élaborer et mettre en œuvre des politiques de confidentialité claires et détaillées pour gerer la collecte, l'utilisation, le partage et la protection des données sensibles. Ces politiques doivent être communiquées aux utilisateurs et aux parties concernées.

6. Audit et surveillance : La mise en place de mécanismes d'audit et de surveillance est importante pour détecter les activités suspectes ou non autorisées et pour garantir le respect des politiques de confidentialité. Cela inclut la surveillance des accès aux données, les journaux d'audit, les alertes de sécurité, etc.

7. Conformité réglementaire : Dans de nombreux domaines, il existe des réglementations et des normes strictes concernant la protection et la confidentialité des données, telles que le RGPD en Europe ou la HIPAA aux États-Unis. Les organisations doivent se conformer à ces réglementations et mettre en œuvre les mesures de sécurité appropriées pour protéger la confidentialité des données.

I.10.4 L'anonymat

Cela se réfère à la capacité de protéger l'identité des utilisateurs ou des systèmes, empêchant ainsi leur identification ou leur traçabilité. Les services de proxy et les réseaux privés virtuels sont utilisés pour masquer l'adresse IP réelle de l'utilisateur, préservant ainsi

son anonymat. Des techniques telles que l'anonymisation des données permet de protéger la vie privée des individus lors de l'analyse des données.

I.10.5 L'intégrité

L'intégrité assure que les données n'ont pas été altérées, endommagées ou modifiées de manière non autorisée. Les mécanismes d'intégrité permettent de détecter toute altération des données et de les restaurer dans leur état d'origine si nécessaire. Les fonctions de hachage comme SHA-256 sont utilisées pour générer des empreintes numériques des données, permettant de vérifier si elles ont été altérées. En plus les signatures numériques sont utilisées pour garantir l'intégrité et l'authenticité des données en associant un message à une entité spécifique, de sorte que toute modification du message puisse être détectée.

I.10.6 La disponibilité

Elle garantit que les ressources informatiques sont accessibles et utilisables lorsque cela est nécessaire. Cela implique souvent la mise en place de mesures pour prévenir les interruptions de service et garantir une disponibilité continue des systèmes. Les systèmes exploitent la redondance matérielle et la réplication des données pour assurer une disponibilité continue, même en cas de défaillance matérielle ou de catastrophe et les serveurs utilisent des techniques d'équilibrage de charge pour repartir la charge de travail de manière égale entre les différents serveurs, évitant ainsi les goulets d'étranglement et les temps d'arrêt.

I.10.7 La non-répudiation

Cela signifie qu'une personne ne peut pas nier avoir effectué une action spécifique. Les mécanismes de non-répudiation fournissent des preuves suffisantes pour prouver qu'une action a été effectuée, ce qui empêche les utilisateurs de rejeter leur responsabilité. Les journaux d'audit enregistrent les actions des utilisateurs et des systèmes, fournissant ainsi une piste de vérification pour prouver qui a fait quoi et quand. De plus les signatures électroniques sont utilisées pour garantir que les transactions électroniques ne peuvent pas être niées par les parties impliquées, fournissant ainsi une preuve juridique de l'accord.

Ainsi toutes les propriétés qu'on a citées sont importantes et essentielles pour garantir la sécurité et la fiabilité des systèmes informatiques et des données. Elles sont souvent mises en œuvre à l'aide de différentes techniques et technologies, telles que le chiffrement, les protocoles d'authentification, les pare-feux, etc.

I.11. Conclusion

Dans ce premier chapitre, nous avons introduit des généralités sur l'internet des objets et ses applications dans les maisons intelligentes en commençant par donner son évolution depuis son apparition jusqu'à nos jours, ses paradigmes ainsi que les différentes architectures adoptées et protocoles de communication associés. Ensuite nous avons présenté les multiples domaines d'application de l'IoT notamment dans la domotique et son apport dans le confort des habitations. Néanmoins cette technologie n'est pas sans défis, ce qui nous a amenés à citer et décrire les différents risques et menaces associés à la sécurisation des données ainsi que les stratégies de sécurisation pour atténuer ces risques et assurer ainsi une protection des utilisateurs et de leurs données sensibles.

Chapitre II

CHAOS EN CRYPTOGRAPHIE

II.1. Introduction

Dans ce deuxième chapitre, nous allons présenter quelques généralités sur les différents concepts et outils utilisés, que nous jugeons nécessaires, pour la réalisation du travail présenté dans cette thèse. En premier lieu, nous présentons des généralités sur la théorie du chaos, en donnant un bref historique, des notions sur les systèmes chaotiques ainsi que leurs propriétés puis nous terminons avec la présentation des différents outils permettant l'étude et l'analyse de ces derniers. En deuxième lieu, nous présentons brièvement des notions sur le cryptage et la cryptanalyse. En dernier lieu, nous donnons les principaux outils et métriques les plus utilisés dans la littérature pour l'évaluation et l'analyse de la robustesse des cryptosystems.

II.2. Théorie du chaos

La Théorie du Chaos est une science avec son vocabulaire, ses lois et ses concepts et qui a pour objet de décrire les phénomènes qui nous entourent de la plus microscopique transmission neuronale jusqu'au mouvement macroscopique des astres de notre univers. Cette théorie captivante par bien de ses aspects offre des perspectives d'avenir encore insoupçonnées et inexplorées [69].

Cette nouvelle théorie traite des systèmes dynamiques rigoureusement déterministes, mais qui présentent un phénomène fondamental d'instabilité appelé « sensibilité aux conditions initiales » qui signifie qu'une modulation d'une propriété supplémentaire de récurrence, rend le système non prédictible en pratique à « long » terme [70]. Elle s'attache principalement à la description de ces systèmes à petit nombre de degrés de liberté, souvent très simples à définir, mais dont la dynamique nous apparaît comme très désordonnée [70].

II.2.1. Définition du Chaos

- **Étymologie du mot chaos** : mot d'origine Grecque « ***χάος***» ou « ***Khaos***». Le Chaos est la première entité de la mythologie grecque, un mélange (désordre) d'où est sortie la terre. Ce désordre est composé de tous les éléments de l'univers (eau, terre, feu, vent...) [70]. Le Chaos est souvent considéré comme étant la base de la création du monde dans différentes mythologies.

Dans la littérature, on retrouve quelques définitions relatives à la théorie du chaos, que nous avons résumées dans les points suivants :

- **Mythologie Grecque** : Entité primordiale d'où naît l'univers.
- **Dictionnaire Larousse** :
  - Confusion générale des éléments de la matière, avant la formation du monde
  - Ensemble de choses sens dessus dessous et donnant l'image de la destruction, de la ruine, du désordre
  - État de confusion générale : Mettre le chaos dans une économie fragile par des mesures intempestives.
- **Scientifique** : Théorie scientifique qui étudie les systèmes non linéaires qui présentent des comportements à l'apparence complexes ou aléatoires, bien qu'ils soient parfaitement définis et déterministes.

II.2.2. Historique

Dans cette section, nous allons présenter les événements clés qui ont marqué le développement de la théorie du chaos depuis sa naissance à ce jour.

- ***Poincaré* et la sensibilité aux conditions initiales**

Poincaré, dans un mémoire datant de 1888, a démontré que le mouvement des trois corps (Soleil, Terre et Mars) est impossible à prévoir avec précision [71]. Le calcul des probabilités lié à ce problème a mené *Poincaré* à l'affirmation suivante : « *Une cause très petite, qui nous échappe, détermine un effet considérable que nous ne pouvons pas ne pas voir, et alors nous disons que cet effet est dû au hasard. Si nous connaissions exactement les lois de la nature et la situation de l'univers à l'instant initial, nous pourrions prédire exactement la situation de ce même univers à un instant ultérieur. Mais, lors même que les lois naturelles n'auraient plus de secret pour nous, nous ne pourrions connaître la situation qu'approximativement. Si cela nous permet de prévoir la situation ultérieure avec la même approximation, c'est tout ce qu'il nous faut, nous disons que le phénomène a été prévu, qu'il est régi par des lois ; mais il n'en est pas toujours ainsi, il peut arriver que de petites différences dans les conditions initiales en engendrent de très grandes dans les phénomènes finaux ; une petite erreur sur les premières produirait une erreur énorme sur les derniers. La prédiction devient impossible et nous avons le phénomène fortuit*. »

Cette citation définit parfaitement les deux principales propriétés du chaos, la sensibilité aux conditions initiales et le déterminisme qui réside dans le fait que si une condition initiale est parfaitement déterminée, alors l'évolution du système l'est aussi.

Le déterminisme traduit l'unicité de la solution pour l'équation différentielle d'un système donné.

- **Stabilité de Lyapunov**

Le mathématicien russe *Alexandre Lyapunov* (1857-1918), fut le premier à suivre les traces de *Poincaré*. Il introduit l'idée de mesurer l'écart entre deux trajectoires ayant des conditions initiales voisines lorsque cet écart évolue exponentiellement, alors on parle de sensibilité aux conditions initiales. Les travaux de *Lyapunov*, d'abord tombés dans l'oubli, seront plus tard très précieux pour étudier certains aspects de la théorie du chaos.

- **Naissance du chaos**

En 1961, le météorologue *Edward Lorenz*, professeur de mathématique au MIT (Massachusetts Institut of Technologies), expérimentait une méthode lui permettant de prévoir les phénomènes météorologiques. C'est par pur hasard qu'il observa qu'une modification minime des données initiales pouvait changer de manière considérable ses résultats. *Lorenz* venait de découvrir le phénomène du chaos en trouvant le premier exemple d'attracteur chaotique qui porta son nom « *Attracteur de Lorenz* » [72].

En mars 1963, la communauté scientifique fait connaissance de ce nouveau phénomène à travers la publication de *Lorenz* « *Deterministic non periodic flow* » [70].

- **Attracteurs étranges**

En 1971, le concept « *d'attracteurs étranges* » a été défini par *David Ruelle* et *Floris Takens* [70]. Les systèmes se caractérisant par ces figures étranges seront dénommés « systèmes chaotiques » à partir de 1975.

- ***Mandelbrot* et Fractal**

En 1973, *Benoît Mandelbrot*, un mathématicien franco-américain (1924-2010), publia dans une revue d'économie l'article « *Formes nouvelles du hasard dans les sciences* » [70], où il développa une nouvelle classe d'objets mathématiques : les objets fractals, ou fractales. Il a également montré qu'un grand nombre d'objets dans la nature étaient bien décrits par des fractales, conduisant ainsi à de nouveaux terrains de recherches. Des fractales se retrouvent également dans des phénomènes étudiés en théorie du chaos et dans les phénomènes économiques, comme les fluctuations boursières.

- **Le terme Chaos**

Le terme « *Chaos* » fut introduit pour la première fois en 1975 par *Tien Yien Li* et *James Yorke* [73], deux mathématiciens qui travaillaient à l'Université du Maryland, dans un article intitulé : « *Period three implies Chaos* ». C'est donc au cours des années soixante-dix que la théorie du chaos a pris son essor.

- ***Feigenbaum* et la transition vers le chaos**

En 1975, le physicien théoricien américain *Mitchell Jay Feigenbaum* a découvert le scénario de transition vers le chaos par doublement de période, lors d'un séjour à Los Alamos. Cette transition est caractérisée par deux constantes universelles, baptisées depuis « nombres de *Feigenbaum* » en son honneur. Ces deux constantes expriment des rapports apparaissant dans les diagrammes de bifurcation de la théorie du chaos [70].

- **Chaos en électronique**

En 1983, *Chua* et *Matsumoto* ont conçu et réalisé le premier circuit électronique chaotique autonome, « *The double scroll oscillator* », connu sous le nom de circuit chaotique électronique de *Chua* [74].

- **Contrôle du Chaos**

En 1990, *Ott*, *Grebogi* et *Yorke* ont publié le premier travail sur la possibilité de contrôler le chaos [75]. Depuis, le contrôle de chaos est devenu un axe de recherche largement exploré et étudié dans les systèmes dynamiques non-linéaires.

- **Chaos et Communication**

En 1993, *Cuomo et al.*, ont pu démontré la robustesse de la synchronisation, suivant la technique de *Pecora* et *Carroll*, pour certains systèmes chaotiques (Lorenz) malgré la présence de faibles perturbations dans le signal de couplage. Cette propriété a ainsi permis à *Cuomo et al.*, de proposer le premier schéma d'un système de communication chaotique basé sur le principe du Masquage Chaotique par Addition (MCA) [76].

En 1998, *Van Wiggeren* et *Roy* ont démontré expérimentalement la faisabilité d'une communication avec des Lasers chaotiques synchronisés [77].

En 2000, des chercheurs russes, ont développé et démontré expérimentalement la faisabilité de générer le chaos directement dans la bande des micro-ondes [78]. Cette grande découverte a bouleversé le domaine de la génération expérimentale du chaos analogique. En effet, celle-ci fut le point de départ pour l'utilisation d'un signal chaotique directement comme porteuse de l'information.

En 2002, Dmitriev *et al.*, ont initiés et développés un nouveau domaine d'application du chaos dans les communications qu'ils nomment Direct Chaotique Communication (DCC) [79]. Cette nouvelle technologie de communication chaotique a ouvert un nouveau domaine d'application du chaos dans les communications qui est les communications Ultra Large Bande (ULB).

- **Chaos et Cryptographie**

En 1997, *Yang et al.*, ont publié un premier travail présentant le concept de crypto-système chaotique (cryptographie basée-chaos) [80]. Ils ont proposé d'intégrer le principe de la cryptographie dans les systèmes de communications chaotiques pour augmenter et améliorer le degré de sécurité dans ces derniers.

A travers le travail présenté dans cette thèse, nous nous intéressons à ce dernier domaine d'application de la théorie du chaos qui connait des progrès très remarquables ces dernières années. Ceci est dû essentiellement aux nombreux et intéressants avantages en termes de simplicité de réalisation, de coût et de performances que procure le chaos.

II.2.3. Systèmes dynamiques

Un système dynamique consiste en un ensemble d'états possibles, avec une loi qui détermine de façon unique l'état présent du système en fonction de ses états passés. Aucun élément aléatoire n'est admis dans cette définition [81]. C'est un système classique qui évolue au cours du temps de façon à la fois :

- Causale : C'est-à-dire que son avenir ne dépend que des phénomènes du passé ou du présent.
- Déterministe : Le système est régi par des lois déterministes non probabilistes. De ce fait, on exclut ici les systèmes stochastiques, qui relèvent pour leur part de la théorie des probabilités.

II.3. Systèmes chaotiques

Un système chaotique est tout système dynamique non-linéaire qui génère un comportement chaotique pour certaines valeurs de ses paramètres et conditions initiales. Ce type de systèmes est déterministe et se caractérise par une extrême sensibilité aux moindres changements des valeurs de ses paramètres ou conditions initiales [82].

Pour être chaotique, le système doit être au minimum de dimension 3 (possède 3 variables d'états) et posséder au moins un exposant de *Lyapunov* positif. L'atmosphère

terrestre, le robinet qui coule et le système solaire, le cerveau et le cœur sont des exemples quotidiens [70].

L'évolution déterministe du système chaotique peut se modéliser de deux façons distinctes :

- Une évolution continue dans le temps, représentée par une équation différentielle ordinaire. C'est a priori la plus naturelle physiquement, puisque le paramètre temps nous semble continu. Nous parlons alors de systèmes chaotiques continus.

- Une évolution discontinue dans le temps. Ce second cas est souvent le plus simple à décrire mathématiquement, même s'il peut sembler a priori moins réaliste physiquement [70]. Cependant, l'étude théorique de ces modèles discrets est fondamentale, car elle permet de mettre en évidence des résultats importants, qui se généralisent souvent aux évolutions dynamiques continues. Nous parlons alors de systèmes chaotiques discrets.

II.3.1. Systèmes chaotiques à temps continu

Ces systèmes sont gérés par une évolution continue dans le temps, représentée par un système d'équations différentielles ordinaires du premier ordre sous la forme :

$$\dot{x}(t) = F(x(t), t) \tag{II.1}$$

Où *t* est le paramètre temps, $x(t)$ est le vecteur d'état de dimension $n \geq 3$, et *F* est la dynamique non linéaire du système.

Parmi les systèmes chaotiques continus, on peut citer :

- Système de *Lorenz*

$$\begin{cases} \dot{x} = \sigma(y - x) \\ \dot{y} = x(r - z) - y \\ \dot{z} = xy - bz \end{cases} \tag{II.2}$$

Avec $\sigma = 10, \mathrm{r} = 28$ et $\mathrm{b} = 8/3$ sont des constantes positives [83].

- Système de *Rössler*[84]

$$\begin{cases} \dot{x} = -(y + z) \\ \dot{y} = x + ay \\ \dot{z} = b + z(x - c) \end{cases} \tag{II.3}$$

- Système de *chen*

$$\begin{cases} \dot{x} = a(y - x) \\ \dot{y} = bx - xz + cy \\ \dot{z} = xy + dz - yz + xz \end{cases} \tag{II.4}$$

- Oscillateur de *Chua*

$$\begin{cases} \dot{x} = \alpha(y - x - f(x)) \\ \dot{y} = x - y + z \\ \dot{z} = -\beta y - \gamma z \end{cases} \quad \text{(II.5)}$$

Avec $\alpha, \beta \; et \; \gamma$ sont des constantes positives.

$f(x) = bx + \frac{1}{2}(a - b)(|x + 1| + |x - 1|)$, et a$< b < 0$.

La figure (II.1) représente les signaux chaotiques continus des trois variables d'état issues du système de *Rössler* [84] de paramètres *a*=0.398, *b*=2, *c*=4 avec les conditions initiales suivantes$(x_0, y_0, z_0) = (1,1,1)$.

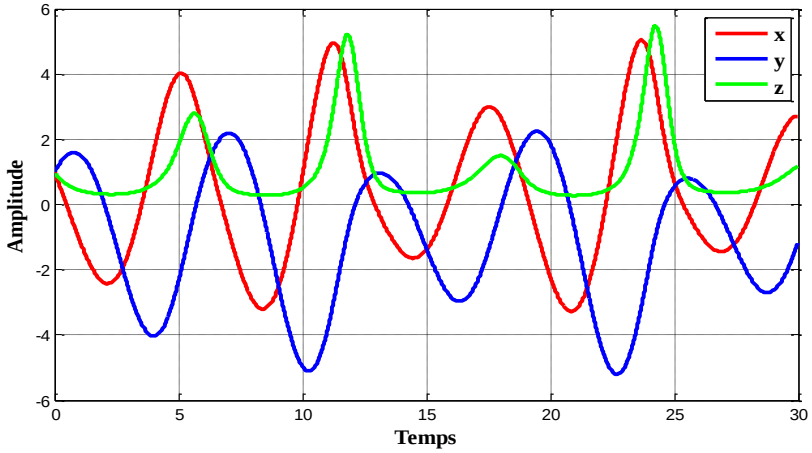


Figure II.1. Système chaotique de Rössler.

Si la dynamique *F* dépend du temps *t*, on parle d'un système dynamique non autonome, sinon, le système est dit autonome. Dans, les systèmes non autonomes le chaos est généré par une excitation (source) externe. Un exemple des systèmes chaotiques non autonomes, est le système de *Duffing* donné par :

$$\begin{cases} \dot{x} = y \\ \dot{y} = -\alpha y + x - x^3 + \beta \cos(\omega t) \end{cases} \quad \text{(II.6)}$$

Si on pose $z = \omega t$, le système devient :

$$\begin{cases} \dot{x} = y \\ \dot{y} = -\alpha y + x - x^3 + \beta \cos(z) \\ \dot{z} = \omega \end{cases} \quad \text{(II.7)}$$

II.3.2. Systèmes chaotiques à temps discret

Ils sont générés par une évolution discontinue dans le temps représentée par une équation aux différences non linéaire (équation de récurrence). Dans la littérature anglo-saxonne, on parle de *Map chaotique* (ou *carte chaotique*) [85] :

$$x(n+1) = F(x(n), n) \tag{II.8}$$

Parmi les systèmes chaotiques discrets, on peut citer :

- Hénon Map $\begin{cases} x_{k+1} = 1 + y_k - a.x_k^2 \\ y_{k+1} = b.x_k \end{cases}$ (II.9)

- Logistic Map $x_{k+1} = r x_k (1 - x_k)$ (II.10)

- Tent Map $\mathcal{T}(X_k) = \begin{cases} u.x_k \ si \quad x_k > 0.5 \\ u(1-x_k) \ si \ x_k \leq 0.5 \end{cases}$ (II.11)

La figure (II.2) représente le signal chaotique discret issu de la récurrence *Logistique* (Equation II.10) avec le paramètre r=3.7 et la condition initiale x_0=0.3 [86].

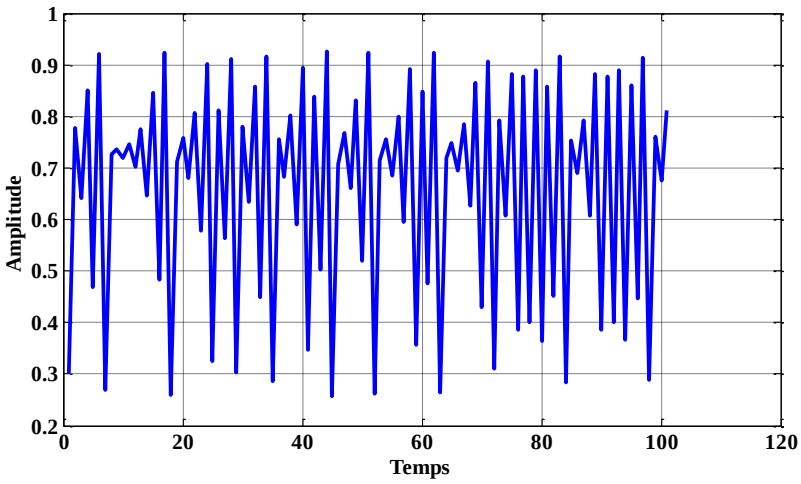


Figure II.2. Evolution de la fonction Logistique.

II.3.3. Systèmes hyperchaotiques

Les systèmes hyperchaotiques sont définis toujours comme étant des systèmes chaotiques à temps continus mais possédants plus d'un exposant de *Lyapunov* positif [87]. Récemment, en physique et en ingénierie, une grande attention est portée à ces systèmes par les scientifiques, car ils génèrent un comportement dynamique plus complexe que celui

généré par de simples systèmes chaotiques 3*D*. Un système dynamique non-linéaire peut générer l'hyperchaos s'il possède au minimum quatre variables d'état (*N*> 3). Ainsi, ces systèmes sont modélisés au minimum par quatre équations différentielles non-linaires. Parmi les plus connus et les plus étudiés dans la littérature, on cite le système hyperchaotique de *Chen* (*N*=4) qui est modélisé par le système d'équations suivant :

$$\begin{cases} \dot{x} = a(y - x) \\ \dot{y} = x(b - z) + cy \\ \dot{z} = xy + (d - y + x)z + w \\ \dot{w} = ew + (y - x)z \end{cases} \quad \text{(II.12)}$$

Où l'hyperchaos est généré pour les valeurs des paramètres : *a*=37, *b*=-11, *c*=26, *d*=-3,

e=-38 et pour les conditions initiales (x_0, y_0, z_0, w_0)= (0.2, 0.3, 0.5, 0,1).

Ce type de générateur chaotique à 4D est le sujet d'intérêt dans ce travail. En effet, nous avons comme objectif d'utiliser ce dernier pour générer des clés de chiffrement robustes, sachant que ce type de système chaotique génère un comportement très complexe et très riche en dynamique chaotique.

II.4. Propriétés des systèmes chaotiques

Les systèmes chaotiques jouent un grand rôle dans plusieurs domaines de la physique et de l'ingénierie grâce à leurs intéressantes et remarquables propriétés. Dans cette partie nous allons citer les différentes propriétés les plus connues et les plus importantes [82].

II.4.1. Déterminisme

De nature, les systèmes chaotiques génèrent un comportement déterministe, car ils obéissent à des lois déterminées et préétablies. Chaque condition initiale détermine entièrement l'évolution future car il n'y a pas de hasard : le système est déterministe.

II.4.2. Sensibilité aux conditions initiales

La sensibilité aux conditions initiales est la propriété la plus importante des systèmes chaotiques. Elle peut être illustrée par ce qu'on appelle « *Effet papillon* ». Popularisé par le météorologue *Lorenz* [72] par sa fameuse métaphore « *On peut considérer que le simple battement d'aile d'un papillon en Australie peut entraîner une tempête sur la côte américaine* ». Ceci signifie qu'une perturbation en apparence mineure à l'échelle de l'atmosphère peut avoir de grandes répercussions. Les courbes de la figure (II-3) illustrent parfaitement ce phénomène pour des conditions initiales très voisines. Dans un premier temps,

les deux systèmes (en bleu et en rouge) évoluent de la même manière ; mais, très vite, leur comportement diverge et devient différent.

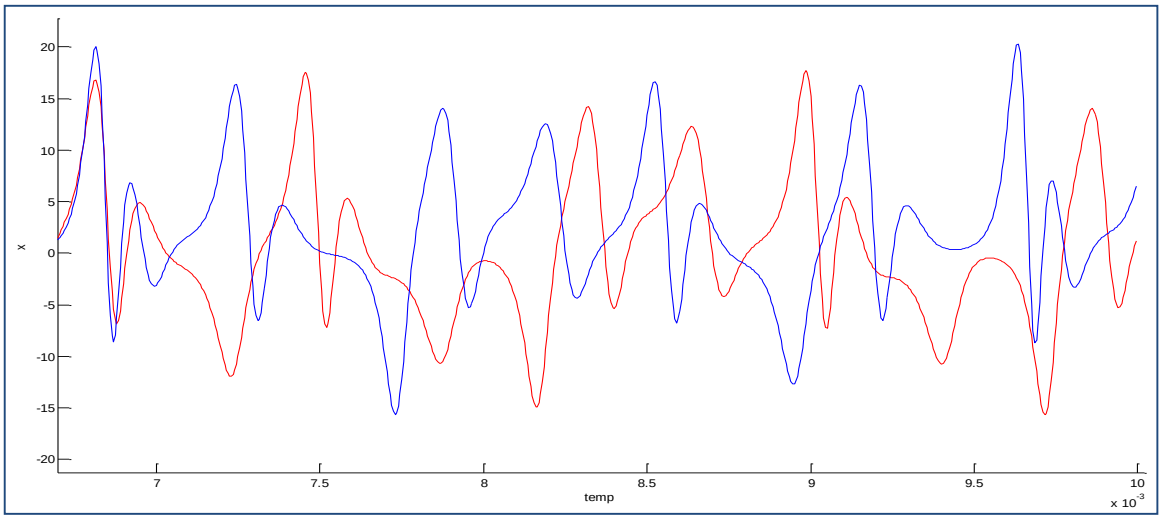


Figure II.3. Sensibilité aux conditions initiales dans le système de Chen pour des conditions initiales très voisines.

II.4.3. Aspect aléatoire

Une autre caractéristique importante des systèmes chaotiques, représentée sur la courbe de la figure (II.4), est l'aspect aléatoire. En effet, un système chaotique évolue d'une manière qui semble aléatoire. La figure (II.4) présente l'évolution du système chaotique de *Lorenz* (pour sa composante y).

II.4.4. Imprédictibilité

Un système chaotique est imprédictible, bien qu'il soit parfaitement décrit par des équations simples et déterministes. Cette imprédictibilité est due à deux raisons principales qui sont les suivantes : La première, c'est la sensibilité aux conditions initiales : c'est-à-dire qu'un léger biais au départ va forcément s'accroître dans le temps. Une différence minime dans les valeurs de départ suffit pour que les courbes finissent par s'écarter. La seconde raison est le changement d'attracteur (également appelé bifurcation) pendant son évolution par exemple le passage du point fixe à l'état d'oscillation.

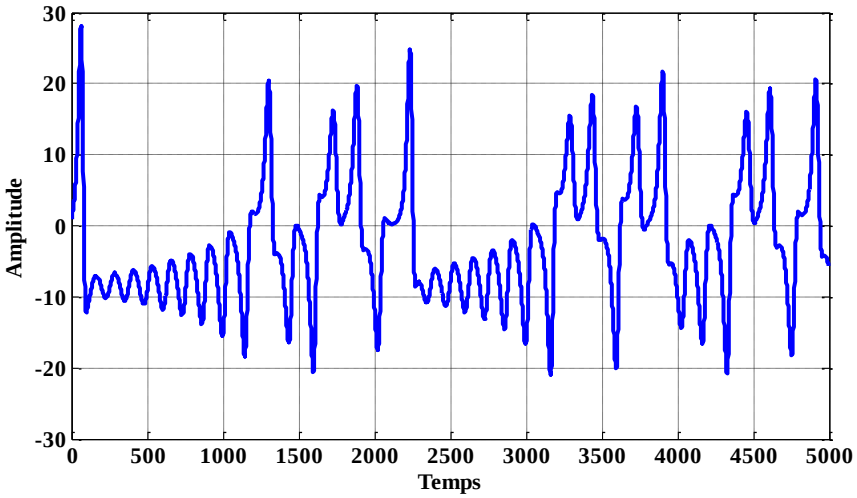


Figure II.4. Aspect aléatoire du système chaotique de *Lorenz* (signal $x(t)$).

II.4.5. Apériodicité

Les systèmes chaotiques génèrent des signaux qui sont ni périodiques ni aléatoires, comme le montre la figure (II.5)[69]. Ces signaux présentent en fait, une répétition non-régulière de certains schémas morphologiquement typés, c'est à dire des symétries appelées *fenêtres périodiques*, dans l'ensemble des valeurs générées (signal).

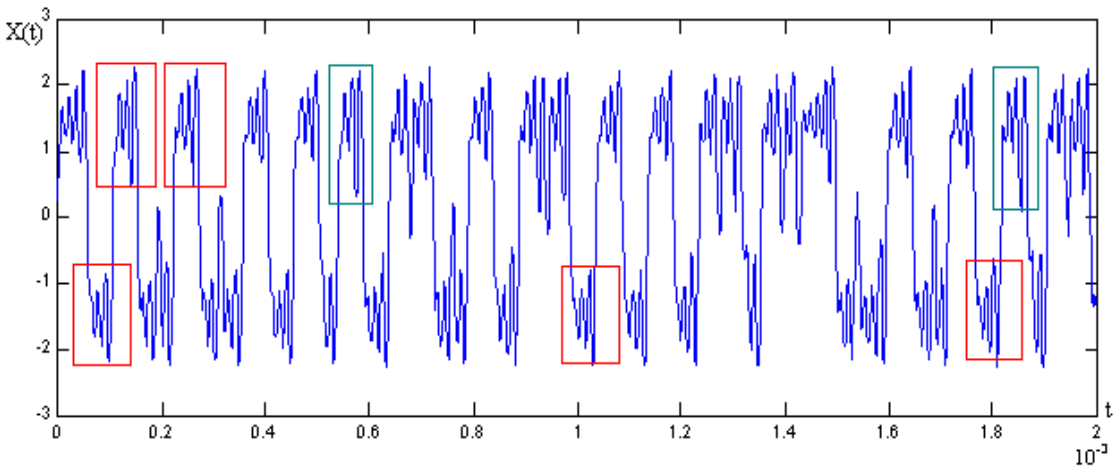


Figure II.5. Le signal chaotique de *Chua*.

II.5. Outils d'étude des systèmes chaotiques

Grâce à la théorie du chaos, des méthodes mathématiques particulières ont été développées afin de comprendre les irrégularités et l'imprédictibilité du comportement des systèmes non-linéaires à faibles dimensions. Dans ce qui suit, nous allons présenter les outils d'études les plus utilisés.

Par définition, les générateurs chaotiques sont appelés à fonctionner en régime chaotique. Et certains outils sont plus appropriés à la détection du chaos. Dans ce contexte, la première méthode consiste à observer la représentation temporelle. Celle-ci n'apporte pas pour autant de preuves fiables. La seconde est le diagramme de bifurcation ou la visualisation de l'exposant de *Lyapunov*. Si ce dernier est positif alors le système associé se trouve en régime chaotique. La troisième est l'attracteur dans l'espace des phases du système. Si celui-ci est à une dimension fractale, alors il s'agit bien d'un système chaotique.

II.5.1 Diagramme de bifurcation

Les comportements d'un système dynamique non linéaire peuvent être caractérisés dans l'espace d'état en temps ou en fréquence. Un diagramme de bifurcation est une représentation du champ attracteur du système par rapport à un paramètre de contrôle. Il est utilisé pour étudier les différentes transitions vers le chaos d'un système dynamique non-linéaire. A partir de ce diagramme, il est possible de déduire les paramètres qui conduisent vers le régime chaotique du système étudié. Ces derniers sont appelés les paramètres de bifurcation [88]. La figure (II.6) présente le diagramme de bifurcation de la fonction *Logistique* définie par l'équation (II.13) en variant le paramètre de contrôle *r* :

$$x(k+1) = r.x(k).(1 - x(k)) \qquad \text{(II.13)}$$

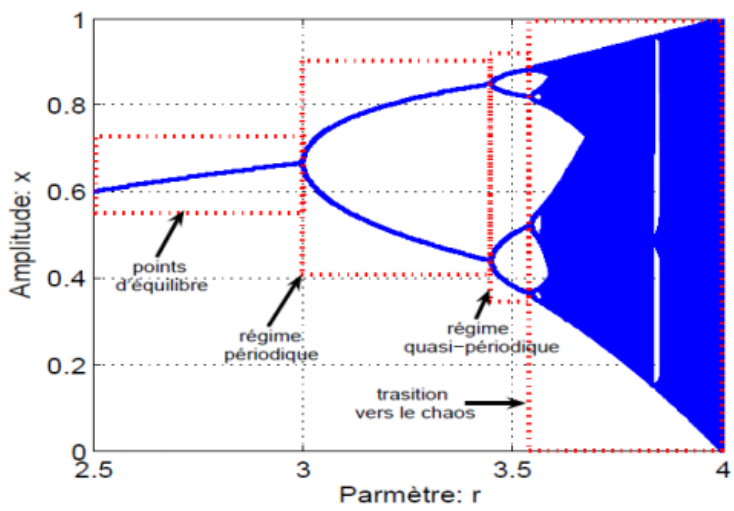


Figure II.6. Diagramme de bifurcation de la fonction Logistique [77].

A partir de ce diagramme, on peut déduire que l'intervalle du paramètre *r* assurant le comportement chaotique est [3.57, 4], et que la valeur de bifurcation est de *r*=3.57.

II.5.2. Exposant de Lyapunov

Les exposants de *Lyapunov* mesurent le taux de convergence ou de divergence de deux trajectoires initialement choisies très voisines. Le nombre d'exposants de *Lyapunov* est égal au nombre de degrés de liberté du système. Lorsque la divergence entre deux trajectoires croit exponentiellement avec le temps pour presque toutes les conditions initiales voisines d'un point donné, on parle alors de phénomène de sensibilité aux conditions initiales. L'exposant de *Lyapunov* sert à quantifier cette sensibilité aux conditions initiales et à mesurer le degré de stabilité du système.

Soit une application discrète *f* de *R* dans *R* qui applique $x(t)$ sur $x(t+1)$. Choisissons deux conditions initiales très proches $x_0 \; et \; x_0 + \varepsilon$, et observons comment se comportent les trajectoires qui en sont issues. Supposons qu'elles s'écartent en moyenne à un rythme exponentiel, on pourra alors trouver un réel λ tel qu'après *t* itérations [89] :

$$|f^t(x_0 + \varepsilon) - f^t(x_0)| \cong \varepsilon e^{\lambda t} \tag{II.14}$$

En passant au logarithme, on trouve :

$$\lambda = \frac{1}{t} ln \left| \frac{f^t(x_0+\varepsilon) - f^t(x_0)}{\varepsilon} \right| \tag{II.15}$$

Si l'on fait tendre ε vers zéro, il vient que :

$$\lambda = \frac{1}{t} ln \left| \frac{df^t(x_0)}{dx_0} \right| \tag{II.16}$$

Ainsi, deux trajectoires dans le plan de phase initialement séparées par un taux Z_1 divergent après un temps *t* (ou *n* itérations) vers un taux Z_2 tel que :

$$|Z_2| = |Z_1|. e^{\lambda t} \tag{II.17}$$

λ est dit exposant de *Lyapunov*.

Le nombre d'exposants de *Lyapunov* est égal à la dimension de l'espace des phases et ils sont généralement indexés du plus grand au plus petit. L'apparition du chaos exige que les exposants de *Lyapunov* doivent remplir trois conditions :

- Au moins un d'eux est positif, ce qui exprime la divergence des trajectoires.
- Au moins un d'eux est négatif pour justifier le repliement des trajectoires.

- La somme de tous les exposants est négative pour expliquer qu'un système chaotique est dissipatif, c'est-à-dire qu'il perd de l'énergie.

La figure (II.7) montre la divergence des deux trajectoires initialement proches d'un système chaotique après un certain temps.

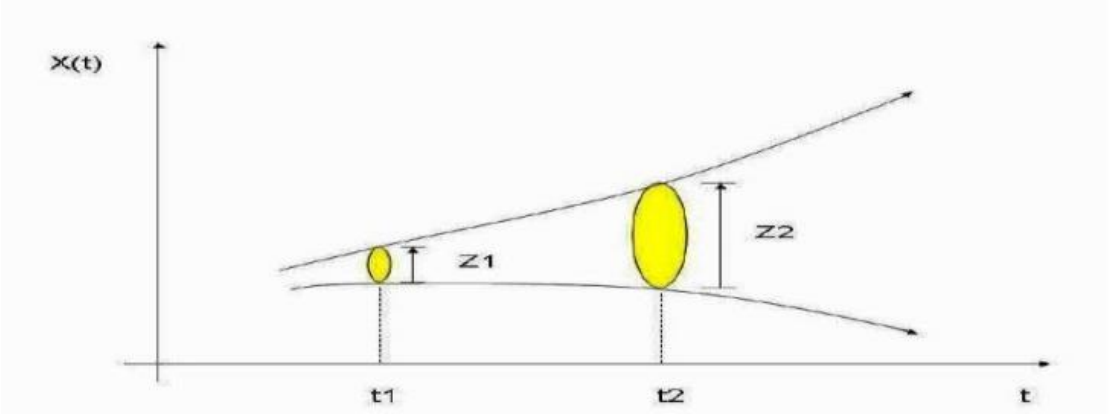


Figure II.7. Divergence de deux trajectoires dans le plan de phase.

II.5.3. Attracteur étrange

Un attracteur est la limite asymptotique des solutions partant de toute condition initiale située dans un bassin d'attraction qui est un domaine de volume non nul. Dans la représentation des systèmes chaotiques, les attracteurs sont des courbes complexes qui présentent une symétrie interne : si l'on applique un « zoom » avant ou arrière sur l'attracteur, on y retrouve toujours la même forme, la même structure. À chaque échelle, il est semblable à lui-même [82].

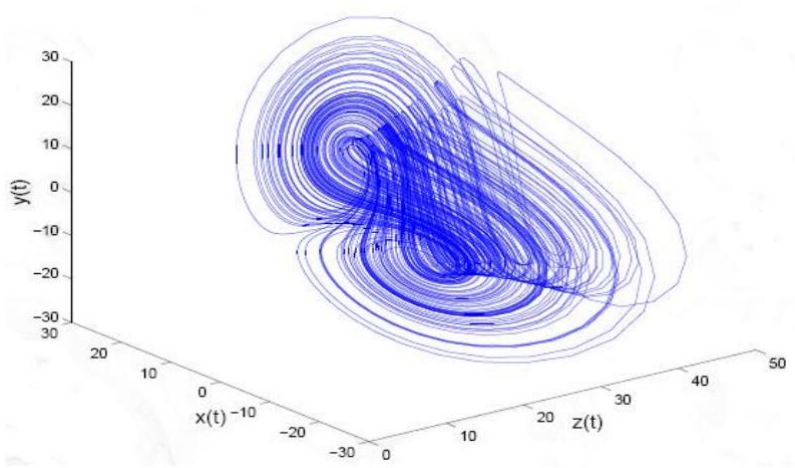


Figure II.8. Exemples d'attracteurs étranges : attracteur étrange de *Chen*.

Les attracteurs chaotiques, dits aussi attracteurs étranges, sont en fait des courbes fractales. Le désordre est dans le système, mais l'ordre fractal, en revanche, apparaît dans ses représentations géométriques (figure II-8). L'évolution du système consiste en :

- Un comportement chaotique au sens étymologique (origine) du terme, où l'évolution de ses coordonnées se fera dans une totale anarchie (comportement aléatoire)[69] ;
- Un comportement chaotique déterministe, où les trajectoires sont régies par des règles fondamentales déterministe et non probabilistes.

II.5.4. Section de Poincaré

L'une des méthodes utilisée pour étudier les systèmes chaotiques est la section de Poincaré, elle se résume à choisir une surface dans l'espace de phase et de tracer les points d'intersections successives entre la trajectoire et la surface. Une fois la section est établie, elle peut nous renseigner sur la nature déterministe ou chaotique du système. La figure (II.9) illustre le principe et montre les différentes intersections avec un plan de section [90].

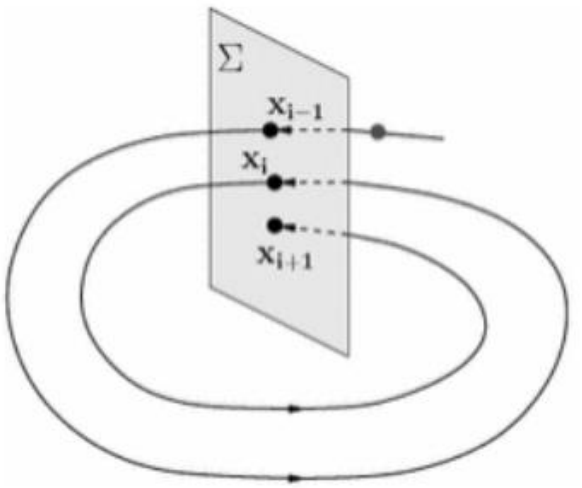


Figure II.9. Exemple de sections de *Poincaré* d'un attracteur chaotique.

II.6. Notion de base et primitives cryptographique

La cryptologie est une science mathématique qui regroupe la cryptographie et la cryptanalyse. Si le rôle des cryptographes est de construire et de prouver des systèmes de chiffrement ou de signature, l'objectif des cryptanalystes est de " casser " ces systèmes. Ainsi de façon complémentaire à la cryptographie, la cryptanalyse concerne l'étude des moyens pour faire échouer ou pour faire briser un procédé cryptographique [91].

II.6.1. La cryptographie

La cryptographie est une technique à base mathématique qui permet de transmettre des données confidentielles sur un médium non sécurisé sans qu'un intrus ne puisse découvrir le contenu. Ces données seront déchiffrées seulement par le destinataire ou celui connaissant la clé de chiffrement. La cryptographie garantit entre autre l'intégrité, la non répudiation et l'authenticité des données en plus de la confidentialité [92].

- La confidentialité permet de garantir que seul le destinataire ou le détenant de la clé puisse découvrir le message en clair.
- L'intégrité permet de s'assurer de la non modification ou la non altération des données pendant le stockage ou la transmission.
- La non répudiation empêche de nier la participation des individus à un échange ou à un traitement de données.
- L'authenticité permet de garantir l'origine et l'identité de l'émetteur.

Une théorie fondamentale a été annoncée en 1883 par A. Kerckhoffs (cryptologue néerlandais). Elle suppose que l'intrus connaît tous les détails du cryptosystème sauf la clé. Le secret doit entourer seulement la clé de cryptage. La clé doit être alors le sésame aboutissant à la solution [92]. La sécurité d'un système de chiffrement est beaucoup plus sûre si sa cryptanalyse nécessite un temps de calcul presque égal à celui de la recherche exhaustive de la clé. Cette évaluation est appelée la sécurité calculatoire.

Claude Shannon, a identifié dans son document publié en 1949 " *Communication Theory of Secrecy Systems*" deux propriétés fondamentales d'un crypto-système qui sont : La confusion et la diffusion. D'après lui, la confusion correspond à une volonté de rendre la relation entre la clé du chiffrement et le texte chiffré la plus complexe possible [93]. La substitution (remplacement d'un symbole du texte clair par un autre) fut une première approche pour introduire cette notion. La diffusion est une propriété où la redondance statistique dans un texte en clair est dissipée dans les statistiques du texte chiffré. Ainsi, les statistiques de la sortie doivent donner le moins possible d'informations sur l'entrée. La permutation des bits d'un bloc entre eux par exemple augmente la diffusion.

Il existe deux grandes familles d'algorithmes cryptographiques à base de clés : les algorithmes à clé secrète (ou algorithmes symétriques), et les algorithmes à clé publique (ou les algorithmes asymétriques).

II.6.2. La cryptographie symétrique

Le principe du chiffrement conventionnel dit à clé secrète est illustré sur la figure (II.10**)**. L'émetteur et le récepteur utilisent la même clé qui doit être privée ou alors une clé peut être déduite à partir de l'autre.

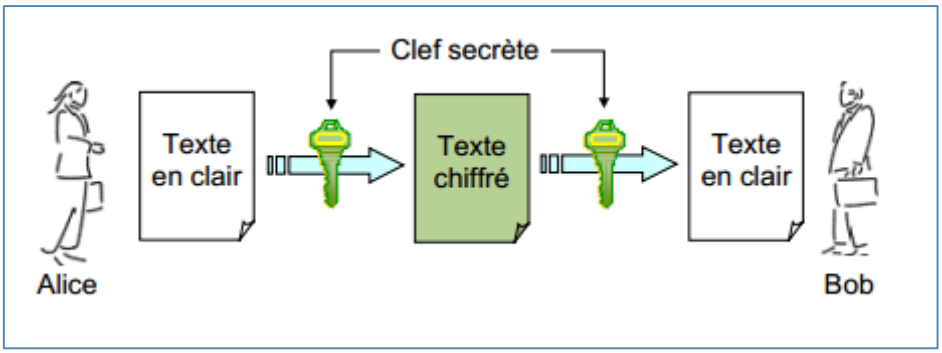


Figure II.10. Principe du chiffrement symétrique.

Le chiffrement et le déchiffrement symétrique d'un message peuvent se faire de deux manières :

a) Chiffrement par flux ou flot (Continu) : il se présente souvent sous la forme d'un générateur de nombres pseudo-aléatoires avec lequel on opère un XOR entre un bit à la sortie du générateur et un bit provenant des données à chiffrer. Toutefois, le XOR n'est pas la seule opération possible. L'opération d'addition est également envisageable (par exemple, addition entre deux octets modulo 256) [91]. Le RC4 (Rivest Cipher 4) est un exemple de chiffrement par flux.

b) Chiffrement par bloc : le message à chiffrer est divisé en blocs de bits de taille fixe. Les blocs sont ensuite chiffrés les uns après les autres. On peut citer comme exemple l'AES (Advanced Encryption Standard) et le DES (Data Encryption Standard).

II.6.3. La cryptographie asymétrique

Le principe du chiffrement asymétrique est illustré sur la figure (II.11). Les clés de chiffrement et de déchiffrement sont différentes et ne peuvent se déduire l'une de l'autre.

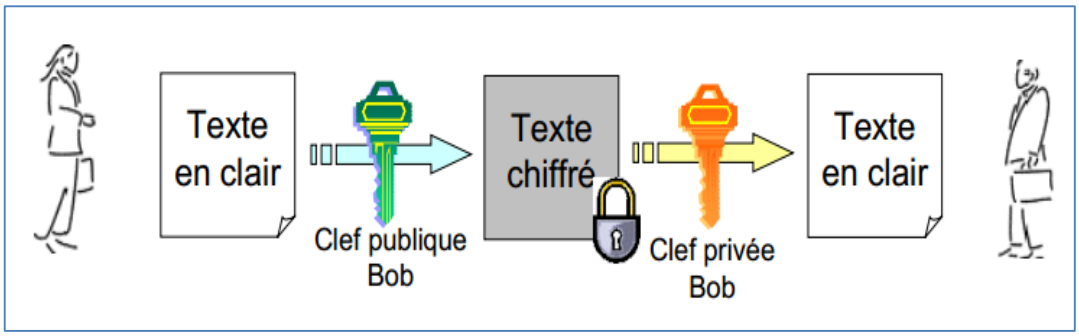


Figure II.11. Principe du chiffrement asymétrique.

Le problème de distribution des clés dans le chiffrement symétrique est résolu par la cryptographie à clé publique, dont le concept fut inventé par Whitfield Diffie et Martin Hellman en 1976[94]. La cryptographie à clé publique repose en fait, sur un schéma asymétrique qui utilise une paire de clés pour le chiffrement : une clé publique, qui chiffre les données, et une clé privée correspondante, aussi appelée clé secrète, qui sera utilisée pour le déchiffrement. Les algorithmes RSA (Rivest-Shamir-Adleman, 1978) et ElGamal (1984) en sont des exemples.

II.6.4. Le masque jetable (Chiffrement de Vernam)

Le masque jetable, également appelé chiffre de Vernam, bien que simple, est le seul algorithme de cryptage connu comme étant indéchiffrable. Il fut inventé par Gilbert Vernam en1917 [95], puis perfectionné par le major Joseph O. Mauborgne en 1918.Néanmoins, son utilisation impose les contraintes suivantes :

- La clé doit être une suite de caractères aussi longue que le message à chiffrer.
- Chaque clé, ou "masque", ne doit être utilisée qu'une seule fois (d'où le nom de masque jetable).
- La clé doit être gardée secrète.

II.6.5. La cryptanalyse

La cryptanalyse regroupe tous les moyens pour décrypter des données (textes, images, etc.) chiffrées sans avoir connaissance de la clé. La plupart des attaques sont adaptées aux techniques de chiffrement. La cryptanalyse des systèmes de chiffrement peut être effectuée sous un certain nombre d'hypothèses. Une hypothèse fondamentale, connue sous le nom de principe de Kerckhoff, est que l'adversaire connait les détails de l'algorithme de chiffrement (la clé est inconnue). La sécurité du crypto-système repose donc entièrement sur la clé secrète. Le but du cryptanalyste est alors de retrouver le texte (image), qui dans la plupart des cas,

nécessite de trouver la clé secrète. L'objectif commun des attaques est de retrouver le texte clair à partir du texte chiffré ou de déduire la clé secrète. Suivant la nature et le type du crypto-système considéré (symétrique, asymétrique) plusieurs attaques sont envisagées, nous citons [96] :

- *L'attaque à texte chiffré (Cipher text attack)* : Le cryptanalyse a connaissance du texte chiffré de plusieurs messages. Il tente alors de déduire la clé secrète ou le texte clair en observant seulement le texte chiffré.
- *L'attaque à texte clair connu (known plaintext attack)* : Le cryptanalyse a accès à plusieurs textes chiffrés ainsi qu'aux textes clairs correspondants. Il essaye alors d'extraire des informations concernant la clé ou le texte en clair à partir du texte chiffré.
- *L'attaque à texte clair choisi (chosen plaintext attack)* : L'attaquant a accès à l'algorithme de chiffrement. Il l'utilise pour générer des paires de texte (Xi : texte en clair, Yi : texte chiffré) de son choix. La différence principale par rapport à l'attaque à texte clair connu, est que le cryptanalyste peut choisir le texte clair à chiffrer.
- *L'attaque adaptative à texte clair choisi (adaptive chosen plaintext attack)* : Cette attaque est une attaque a texte clair choisi mais le choix du texte clair peut dépendre du texte chiffré récupère précédemment.
- *L'attaque à texte chiffré choisi (chosen cipher text attack)* : L'attaquant a accès à l'algorithme de déchiffrement. Il peut choisir les textes à déchiffrer et il connait également les textes en clairs correspondants.
- *L'attaque exhaustive ou attaque par force brute (brute force attack)* : L'attaquant cherche et essaye toutes les combinaisons possibles des clés (recherche exhaustive) jusqu'à l'obtention d'un texte clair. Cette attaque est la plus couteuse en termes de ressources mémoires et du temps de calcul.
- *La cryptanalyse différentielle* : C'est une attaque à message en clair choisi, applicable aux algorithmes de chiffrement par blocs itératifs. Dans ce cas, l'adversaire analyse l'effet de la différence entre les textes clairs sur les textes chiffrés résultants. Il observe l'évolution des différences entre deux textes lorsqu'ils sont chiffrés avec la même clé. En analysant ces différences entre paires de texte, il est possible d'attribuer une probabilité à chaque clé possible. A force d'analyser des paires de texte, on finit soit par trouver la clé recherchée, soit par réduire suffisamment le nombre de clés possibles pour pouvoir mener une attaque exhaustive rapide.
- *La cryptanalyse linéaire* : C'est une attaque à texte en clair connu. Le cryptanalyste propose de construire une expression linéaire approximative du texte chiffré.

- *L'attaque par canaux auxiliaires* : Dans ce type d'attaque, toutes les approches pour analyser les propriétés non contrôlées (consommations électrique, rayonnement électromagnétique, temps de calcul etc.) d'un algorithme sont prises en compte pour réussir à casser le cryptosystème. Par exemple, dans le cas des algorithmes de chiffrement implémentés, la consommation électrique mesurée ou observée pour chaque type de calcul du chiffrement peut être exploitée pour déduire certaines informations sur la clé.

II.7. Cryptographie basée-chaos

Le concept de la cryptographie basée-chaos est introduit dans la littérature à partir de 1997. Il représente la troisième génération des communications basées-chaos présentée dans [97]. Son principe est présenté par le schéma de la figure (II.12). Cette génération englobe les crypto-systèmes chaotiques qui reposent sur l'association des techniques de cryptographie classiques avec la synchronisation chaotique pour accroitre le degré de sécurité. En effet, comme le montre la figure (II.12), dans un crypto-système chaotique le message $p(t)$ est crypté par une certaine règle de cryptage utilisant une clé $k(t)$ générée par un système chaotique au niveau de l'émetteur. Le message ainsi crypté est transmis à travers le canal public qui est à la portée des intrus. Puisque l'intrus ne connait pas la clé de chiffrement chaotique, il est très difficile pour lui d'extraire $p(t)$ de $s(t)$.

Au niveau du récepteur, le signal reçu $r(t)= s(t)+n(t)$, où $n(t)$ représente le bruit du canal, est utilisé pour synchroniser les deux systèmes chaotiques de l'émetteur et du récepteur. Après que la synchronisation est obtenue, les signaux $k(t)$ et $y(t)$ peuvent être récupérés au niveau du récepteur avec un certain bruit et sont notés ainsi $k'(t)$ et $y'(t)$ respectivement. En injectant $k'(t)$ et $y'(t)$ dans le bloc règle de décryptage du récepteur, le message $p(t)$ peut être récupéré avec un certain bruit. Cette génération reste la plus sécurisée dans tous les systèmes de communication chaotiques sécurisés proposés auparavant.

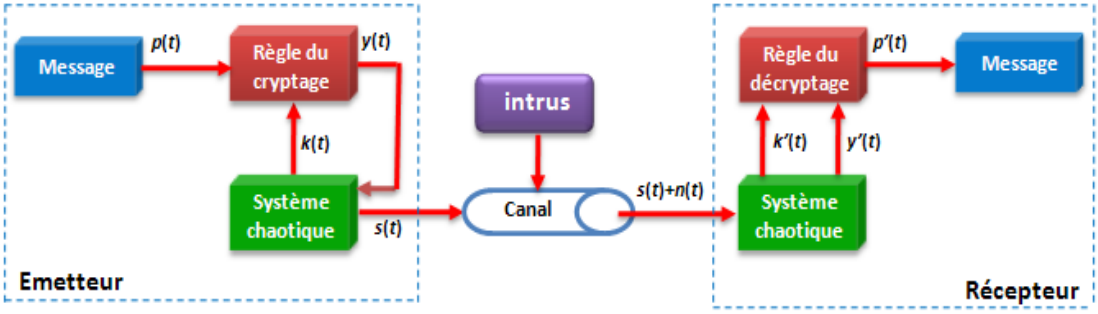


Figure II.12. Principe de la cryptographie basée-chaos[69].

Il est à noter que dans la majorité des cas, la règle du cryptage utilisée dans la cryptographie basée-chaos est tout simplement l'opérateur « XOR » réalisant ainsi un crypto-système à masque jetable (également appelé chiffre de Vernam) [98]. Le chiffrement par la méthode du masque jetable consiste à combiner le message en clair avec une clé présentant les caractéristiques suivantes :

- La clé doit être au moins aussi longue que le message à chiffrer.
- Les caractères composant la clé doivent être choisis de façon totalement aléatoire.
- Chaque clé, ou « masque », ne doit être utilisée qu'une seule fois (d'où le nom de *masque jetable*).

Plus précisément, les auteurs dans [98] ont donné le schéma de principe du crypto-système basé-chaos le plus adéquat pour le chiffrement des données multimédia. Ce schéma est représenté sur la figure (II.13).

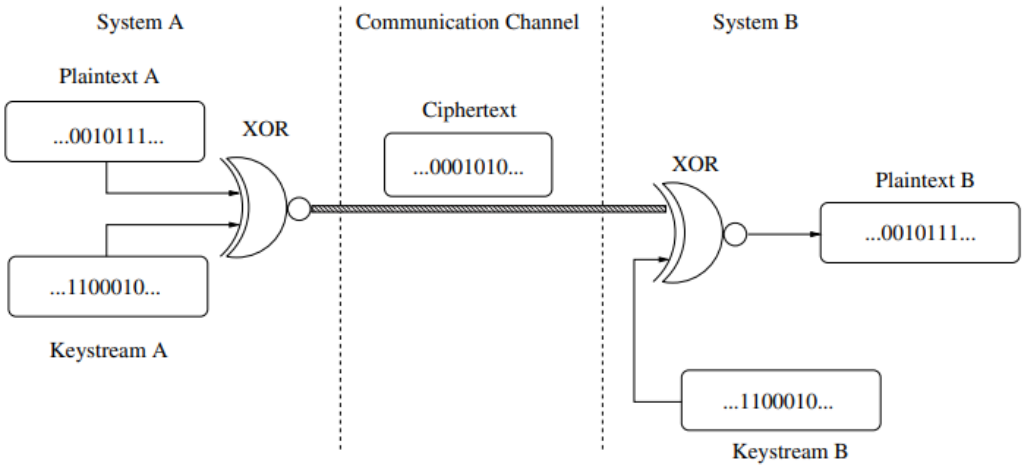


Figure II.13. Cryptographie basée-chaos de Vidal [98].

II.7.1. Chaos vs Cryptographie

Depuis les années 90, un grand nombre de chercheurs ont remarqué qu'il existe une relation très intéressante entre le chaos et la cryptographie. En effet, de nombreuses propriétés des systèmes chaotiques contribuent à cette correspondance, plus particulièrement :

- L'extrême sensibilité des systèmes chaotiques à la moindre variation des valeurs de leurs conditions initiales et paramètres de contrôle.
- L'imprédictibilité des trajectoires du comportement chaotique due à cette sensibilité.

En 2006, *Alvares* et *Li* ont identifié et résumé en détails la ressemblance des propriétés des systèmes chaotiques et cryptographiques dans le tableau (II.1) suivant [99] :

Propriété chaotique	Propriété cryptographique	Description
Ergodicité	Confusion	La sortie a la même distribution quelque que soit l'entrée
Sensibilité aux conditions initiales et aux paramètres	Diffusion avec un petit changement de la clé	Une légère variation en entrée engendre un grand changement en sortie
Propriété de mélange	Diffusion avec un petit changement dans le texte	Une petite déviation dans une zone de localisation engendre un grand changement dans tout l'espace
Dynamique déterministe	Pseudo-aléatoire déterministe	Un processus déterministe peut engendrer un comportement pseudo-aléatoire
Complexité de la structure	Complexité de l'algorithme	Un processus simple a une complexité élevée

Tableau II.1. Comparaison des propriétés chaotiques et cryptographiques.

Ainsi, les similarités entre les méthodes cryptographiques classiques et les méthodes cryptographiques chaotiques sont nombreuses, toutefois quelques différences subsistent. En effet, la cryptographie classique utilise un ensemble de nombres entiers fini alors que la cryptographie chaotique utilise un ensemble de nombres réels. De plus, cette dernière s'appuie sur des méthodes analytiques alors que la méthode classique utilise des méthodes algébriques.

II.7.2. Type de cryptosystèmes chaotiques

Selon l'approche de conception menée, les cryptosystèmes basés chaos sont classés en deux catégories, en l'occurrence [82] :

- *Crypto-systèmes chaotiques analogiques* : Dans ce type de crypto-systèmes, la synchronisation chaotique entre l'émetteur et le récepteur est nécessaire. Les générateurs chaotiques utilisés sont réalisés à l'aide de circuits électroniques analogiques. Le principe de base est la génération d'un signal chaotique continu analogique, au niveau de l'émetteur, pour masquer un signal information analogique. L'avantage de ce type de chiffrement chaotique réside dans l'utilisation des nombres réels (le signal chaotique est analogique). Mais, leur plus grand inconvénient réside dans l'impossibilité de réaliser la synchronisation chaotique en analogique à cause de la sensibilité des générateurs chaotiques aux moindres variations de leurs paramètres et conditions initiales. Par conséquent, ce type de cyptosystèmes chaotique est totalement abandonné.

- *Crypto-systèmes chaotiques numériques* : Ces crypto-systèmes utilisent principalement les systèmes chaotiques à temps discrets, modélisés par des systèmes d'équations récursives et appelés couramment cartes (Map en anglais) chaotiques, ou à temps continu implanté numériquement en software ou hardware. Ces derniers sont utilisés pour générer des séquences binaires qui seront exploitées comme clés de chiffrement. Dans cette thèse, nous allons nous intéresser à ce type de crypto-systèmes.

II.8. Outils d'évaluation de la qualité des clés de chiffrement

D'une manière générale, l'aléatoire ne peut pas être prouvé mathématiquement. Toutefois, pour analyser l'aspect aléatoire d'une séquence de bits produite par un générateur de nombres aléatoires RNG (Random Number Generator) ou pseudo-aléatoire PRNG, différentes batteries de tests statistiques standardisées sont proposées. Afin de détecter les faiblesses du RNG testé, chaque test statistique détermine si la séquence de bits générée, de taille voulue, présente au maximum les mêmes caractéristiques qu'une suite vraiment aléatoire vis-à-vis de certaines propriétés statistiques. Plus précisément, chaque test calcule une donnée statistique liée à la séquence générée et compare cette valeur à une valeur théorique. Si ces valeurs sont trop éloignées, alors la séquence est rejetée, i.e., elle ne se comporte pas comme une séquence aléatoire ; sinon, elle est acceptée comme une séquence aléatoire. Ainsi, les RNG doivent subir des processus d'évaluations de qualité et de sécurité avant qu'ils puissent être utilisés dans des crypto-systèmes. Pour ce faire, plusieurs méthodologies sont approuvées pour assurer l'objectif d'évaluation de la qualité de ce type de générateurs. Nous citons ci-dessous les quatre les plus connus :

- **FIPS 140** : Deux versions sont disponibles **:** FIPS140-1 et FIPS 140-2 dont la différence réside uniquement dans les seuils de décision [100]. Cette batterie de test se compose de quatre tests et applicable sur une séquence binaire de 20000 bits.
- **DIEHARD** : Proposée par *Marsaglia* de l'Université de Floride en 1995 [101], cette batterie de test présente un certain nombre de tests très rigoureux dans le sens qu'il est très difficile de les passer contrairement aux autres batteries de tests existantes. Elle est considérée comme une collection de tests complète pour détecter l'aspect non aléatoire sur de très longues séquences (quelques Mbits).
- **AIS** : Proposée par le BSI (*Bundes amtfûr Sicherheitinder Informations technik*), qui est l'office fédéral pour la sécurité des technologies de l'information Allemande, cette méthode inclut de nouveaux tests statistiques examinant l'entropie, mais également

une méthodologie pour l'évaluation de la sécurité des générateurs TRNG (True Random Number Generator) [102].

- **NIST** : Cet institut a proposé la série STS (*Statistical Test Suite*) comprenant 17 tests pour conclure sur le caractère aléatoire ou non des séquences binaires [103]. Dans cette série, une hypothèse est faite sur une longueur *n* de la séquence à étudier, où *n* est de l'ordre de 10^3 à 10^7, car les lois de distributions choisies sont représentatives pour des séquences longues. Ainsi pour chaque test, un ensemble *n* de *P_value* devrait indiquer un échec. En effet, un niveau de confiance de α=0.01 indique que 1% des séquences devraient échouer. Pour une *P_value* $\geq \alpha$=0.01 cela signifie que la séquence est aléatoire avec un taux de confiance égal à (1-α)=99%, tandis que pour une *P_value* < α=0.01 cela indique que la séquence n'est pas aléatoire avec un taux de confiance égal à (1-α)=99%.

II.9. Métriques et analyse de sécurité des images chiffrées

La mesure fondamentale pour évaluer la qualité d'un cryptosystème est sa capacité à résister à toute tentative d'accès non autorisée pour soustraire l'information. Cette mesure est appelée sécurité. Le succès d'une méthode de cryptage est sa résistance face aux diverses attaques telles que l'attaque par force brute, l'attaque statistique et l'attaque différentielle. Dans ce contexte, nous allons citer quelques tests de sécurité les plus connus et utilisés tels que l'espace des clés, le calcul d'entropie, l'autocorrélation, les histogrammes et la sensibilité à la clé pour les attaques statistiques. Pour les attaques différentielles on citera les tests UACI/NPCR ainsi que la distance de Hamming. Ces tests seront utilisés par la suite pour validé l'efficacité et la robustesse des solutions proposées.

II.9.1. Analyse différentielle

Une propriété recherchée des cryptosystèmes est la haute sensibilité aux petits changements dans l'image originale. 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. Ainsi si un changement mineur dans l'image en clair peut provoquer un changement significatif dans l'image cryptée (effet d'avalanche), alors l'attaque différentielle devient inutile. Généralement deux métriques sont utilisées pour mesurer la résistance d'un cryptosystème contre les attaques différentielles introduites par Eli Biham et Adi Shamir [104]. Il s'agit du NPCR (Number of Pixel Change Rate) et du UACI (Unified Average Changing Intensity) définies par les formules suivantes [105] :

$$NPCR = \frac{\sum_{i=0}^{M-1}\sum_{j=0}^{N-1} D(i,j)}{MxN} \text{x } 100 \quad \text{(II.18)}$$

$$UACI = \frac{1}{MxN}\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}\frac{|I_1(i,j)-I_2(i,j)|}{255} \text{x}100 \quad \text{(II.19)}$$

Où M et N représentent respectivement la largeur et la hauteur de l'image. $I_1(i,j)$ et $I_2(i,j)$ 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. $D(i,j)$ est une matrice de la même taille que I_1 et I_2 tel que :

$$D(i,j) = \begin{cases} 1 & \text{si } I_1(i,j) \neq I_2(i,j) \\ 0 & \text{sinon} \end{cases} \quad \text{(II. 20)}$$

Ainsi le NPCR mesure le pourcentage du nombre de pixels différents par rapport au nombre total de pixels entre deux images. Tandis que l'UACI mesure la moyenne de différence d'intensité entre les deux images. La valeur optimale du NPCR est de 99,61%, et celle de l'UACI est de 33.46% [105].

Ajouter à ces deux métriques, qui opèrent sur les octets, nous utilisons aussi une autre métrique à savoir la Distance de Hamming (DH), qui opère sur les bits, pour mesurer l'effet d'avalanche. La DH est donnée par la relation suivante [106] :

$$DH(I_1, I_2) = \frac{1}{I_b}\sum_{K=1}^{I_b}(I_1(K) \oplus I_2(K)) \quad \text{(II.21)}$$

Avec I_b=MxNx8, est la taille de l'image en bits alors que I_1 et I_2 représentent les images cryptées des deux images qui ne diffèrent que d'un seul pixel.

La valeur optimale DH est de 50%. En effet, un bon chiffrement devrait produire une DH voisine de 50%, ce qui signifie qu'un changement d'un bit dans l'image originale va produire une probabilité de ½ de changement dans les bits de l'image chiffrée [91,107].

II.9.2. Entropie

L'entropie $H(m)$, définie par Shannon, d'une source M comportant m symboles (m=256 niveaux de gris) est donnée par [108] :

$$H(M) = \sum_{i=1}^{m} p(m_i) log_2 \frac{1}{P(m_i)} \quad bits \quad \text{(II.22)}$$

Où $p(m_i)$ représente la probabilité du symbole m_i et l'entropie se mesure en bits.

Dans le cas d'une distribution uniforme, l'entropie $H(M)$ est maximale et donnée par :

$$Hmax = log_2(m) = log_2(256) = 8 \text{ bits} \quad \text{(II.23)}$$

Ainsi, si la valeur de l'entropie est très proche de 8, cela signifie que la distribution des niveaux de gris dans l'image est quasiment uniforme ce qui implique que le cryptosystème résiste bien à l'attaque statistique.

II.9.3. Espace des clés

L'espace des clés est le nombre total des différentes clés qui peuvent être utilisées dans le processus de chiffrement. Pour que le chiffrement d'image soit sécurisé, l'espace des clés doit être suffisamment large pour rendre l'attaque par force brute inefficace. Toutefois, dans la cryptographie basée-chaos, une attention particulière doit être donnée à la notion de clés de chiffrement et notion de clés secrètes. En effet, comme nous l'avons expliqué auparavant les cryptosystèmes basés-chaos utilisent généralement le chiffre de Vernam à base du XOR [98] et comme les générateurs chaotiques sont utilisés pour générer des clés de chiffrement selon les valeurs de leurs conditions initiales et paramètres de contrôle, alors il faut définir :

1. **Clés secrètes** : les valeurs des paramètres et conditions initiales.
2. **Clés de chiffrement** : les solutions du générateur chaotique.

Dans la littérature, pour qu'un cryptosystème basé-chaos résiste à l'attaque par force brute, il faut que son espace des clés doit être supérieur ou égal à 2^{128} [109].

II.9.4. Sensibilité des clés

Un chiffrement d'image robuste doit être sensible à tout changement dans la clé secrète. Le moindre changement dans la clé doit produire une image chiffrée complètement différente. De ce fait, la sensibilité à la clé est essentielle pour un cryptosystème. En chiffrement basé-chaos, le moindre changement dans la clé de déchiffrement entraîne un grand changement dans le résultat de déchiffrement [110].

II.9.5. 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. En fait l'histogramme illustre comment les niveaux de gris des pixels dans une image sont distribués [111]. L'analyse par histogramme nous renseigne sur la distribution des pixels dans une image en traçant le nombre de pixels à

chaque niveau d'intensité. Une image sécurisée est caractérisée par un histogramme plat qui correspond à une distribution uniforme.

II.9.6. Test de chi carré

Le test visuel d'histogramme n'est pas suffisant pour affirmer l'uniformité de la distribution. Pour mesurer cette uniformité statistiquement, nous appliquons le test de chi-carré donné par la relation suivante :

$$\chi^2_{exp} = \sum_{i=0}^{Q-1} \frac{(o_i - e_i)^2}{e_i} \qquad (II.24)$$

où Q=256 est le nombre de niveaux de gris, o_i est la fréquence d'occurrence observée pour chaque niveau de gris $i \in [0,255]$ dans l'histogramme de l'image chiffrée, e_i *est* la fréquence d'occurrence théorique pour une distribution uniforme, ici $e_i = T_0/Q$, avec *To* la taille de l'image en octet. La distribution de l'histogramme d'une image est uniforme si la valeur du chi carré calculée est inférieure au chi carré théorique χ^2_{th} qui est de 293 pour un seuil α fixé à α = 0,05 (α ici est le niveau de signification) [112].

II.9.7. Corrélation des pixels adjacents

Le calcul du coefficient de corrélation entre les pixels adjacents permet l'évaluation de la qualité de cryptage des crypto-systèmes. Chaque pixel est fortement corrélé avec le pixel adjacent dans une image claire, ceci permettrait d'estimer la valeur d'un pixel à partir de celles des pixels se trouvant tout autour. Si deux variables sont étroitement liées, le coefficient de corrélation est proche de 1 ou de -1, sinon il tend vers zéro. Donc, l'idéal pour une image chiffrée, est d'avoir des coefficients de corrélations nuls dans toutes les directions horizontale, verticale et diagonale

Le coefficient de corrélation r de deux images pourra être calculé en utilisant les formules suivantes :

$$E(x) = \frac{1}{N}\sum_{i=1}^{N} x_i \qquad (II.25)$$

$$D(x) = \frac{1}{N}\sum_{i=1}^{N} ((x_i - E(x_i))^2 \qquad (II.26)$$

$$cov(x,y) = \frac{1}{N}\sum_{i=1}^{N} (x_i - E(x_i))\big(y_i - E(y_i)\big) \qquad (II.27)$$

$$r_{xy} = \frac{cov(x,y)}{\sqrt{D(x)D(y)}} \quad \text{(II. 28)}$$

Où *x* et *y* sont deux pixels adjacents dans l'image, *cov*(*x*, *y*) est la covariance entre deux pixels adjacents et *N* est le nombre total des pixels.

II.10. Conclusion

Dans ce deuxième chapitre, nous avons présenté et énoncé brièvement des généralités sur la théorie du chaos que nous avons jugé nécessaire à la compréhension du travail présenté dans cette thèse. En effet, nous avons montré les propriétés, les plus attirantes, du comportement chaotique qui ont permis à la théorie du chaos d'évoluer à grand pas en un peu de temps et d'être présente dans plusieurs domaines de la physique et de l'ingénierie. Ensuite, nous avons présenté des notions de base sur tout ce qui est cryptage et cryptanalyse. Puis nous nous somme intéressé aux différents outils et métriques les plus répondus dans la littérature permettant le test et l'évaluation de la robustesse des cryptosystemes contre les différentes attaques.

Chapitre III

CRYPTOSYSTEME HYPERCHAOTIQUE POUR RESEAU DE CAPTEURS SANS FIL

III.1. Introduction

Dans ce troisième chapitre, nous allons présenter et détailler le cryptosysteme basé hyperchaos pour la sécurisation des réseaux de capteur sans fil qu'on a développés ainsi que son implémentation hardware. On commence par la présentation et la simulation, sous Matlab, du générateur de clés de chiffrement hyperchaotiques basé sur le système hyperchaotique de Chen. Nous montrons les signaux temporels et attracteurs hyperchaotiques caractérisant ce dernier. Après la génération des clés de chiffrement, une analyse statistique est effectuée sur ces dernières en utilisant la batterie de tests de NIST. Ensuite, une analyse de sécurité est élaborée incluant une analyse différentielle et statistique sur des données chiffrées (images). Puis nous présentons l'architecture du la plateforme matérielle développée qui est basée sur un RCSF (Réseaux de Capteur San Fil) utilisant une carte Arduino Due, un shield WI-Fi et un capteur DHT22 pour la collecte de la température et de l'humidité du l'environnement. Par la suite une description et une analyse des performances de l'algorithme (basé sur le mécanisme de génération de clés de chiffrement) servant pour le cryptage des données, avant leurs transmissions, est effectuée. Enfin, une conclusion clôture ce chapitre.

III.2. Implémentation du générateur hyperchaotique de Chen

Il existe plusieurs systèmes chaotiques divisés en famille selon leurs dimensions à 3D ou plus, à temps continu ou discret et autonomes ou non. Dans ce travail, nous nous intéressons aux systèmes hyperchaotiques qui sont très attractifs pour application en cryptographie. En effet, ils se distinguent des autres types de systèmes chaotiques par la richesse en dynamique chaotique du comportement, qu'ils génèrent et l'immense espace des clés qu'ils offrent. Plus particulièrement, nous nous sommes intéressés à l'utilisation du système de Chen [113]. Ce choix est justifié dans la suite de cette section.

Le système hyperchaotique de Chen est décrit par le système d'équations non-linéaires suivant [113] :

$$\begin{cases} \dot{x} = a(y - x) \\ \dot{y} = x(b - z) - cy \\ \dot{z} = xy + (d - y + x)z - w \\ \dot{w} = ew + (y - x)z \end{cases} \quad \text{(III.1)}$$

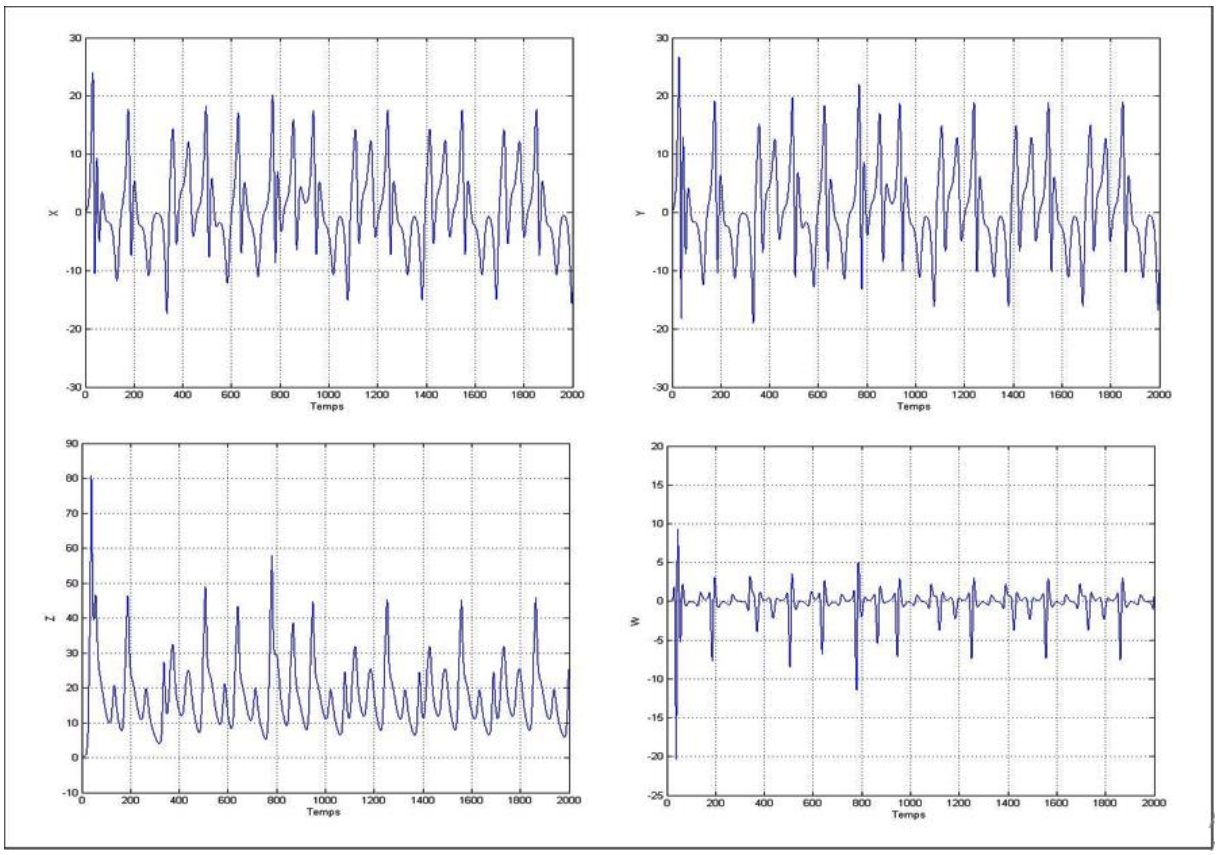
Où *x*, *y*, *z* et *w* sont les quatre variables d'état et *a*, *b*, *c*, *d* et *e* sont des constantes réelles. Le système est hyperchaotique pour les valeurs des paramètres *a* = 37, *b* = -11, c= -26, *d*= -3 et e=-38, et avec les valeurs des conditions initiales $x_0 = 0.2$, $y_0 = 0.3$, $z_0 = 0.5$ et $w_0 = 0.1$.

Pour la résolution du système hyperchaotique de Chen (Equation. III.1), nous implantons la méthode d'intégration numérique de Runge-Kutta d'ordre 4 (RK-4).

III.2.1. Signaux temporels et attracteurs hyperchaotiques

Les résultats de simulation temporelle, sous Matlab, du système hyperchaotique de Chen, résolu par la méthode RK-4 avec un pas *h*=0.01 et pour les paramètres *a*=37, *b*=-11, *c*=, *d*=-3 et *e*=-38, sont présentés dans les figures (III.1) et (III.2). La figure (III.1) présente les signaux hyperchaotiques *x*(*t*), *y*(*t*), *z*(*t*) et *w*(*t*) et la figure (III.2) présente les attracteurs hyperchaotiques 2D et 3D dans les espaces (*x*-*y*), (*x*-*z*), (*x*-*w*), (*x*-*w*-*z*), (*x*-*y*-*z*) et (*w*-*y*-*z*) respectivement.

De ces résultats, nous remarquons que les signaux hyperchaotiques présentent des comportements non-périodiques de forme semblable à celle du bruit. De plus, les attracteurs sont formés par des orbites qui occupent des parties importantes des espaces de phases. Nous remarquons aussi que les trajectoires de phases prises pour des conditions initiales très voisines, divergent dans le temps. Cela se traduit par le phénomène de sensibilité aux conditions initiales.



FigureIII.1. Signaux temporels du système hyperchaotique de Chen.

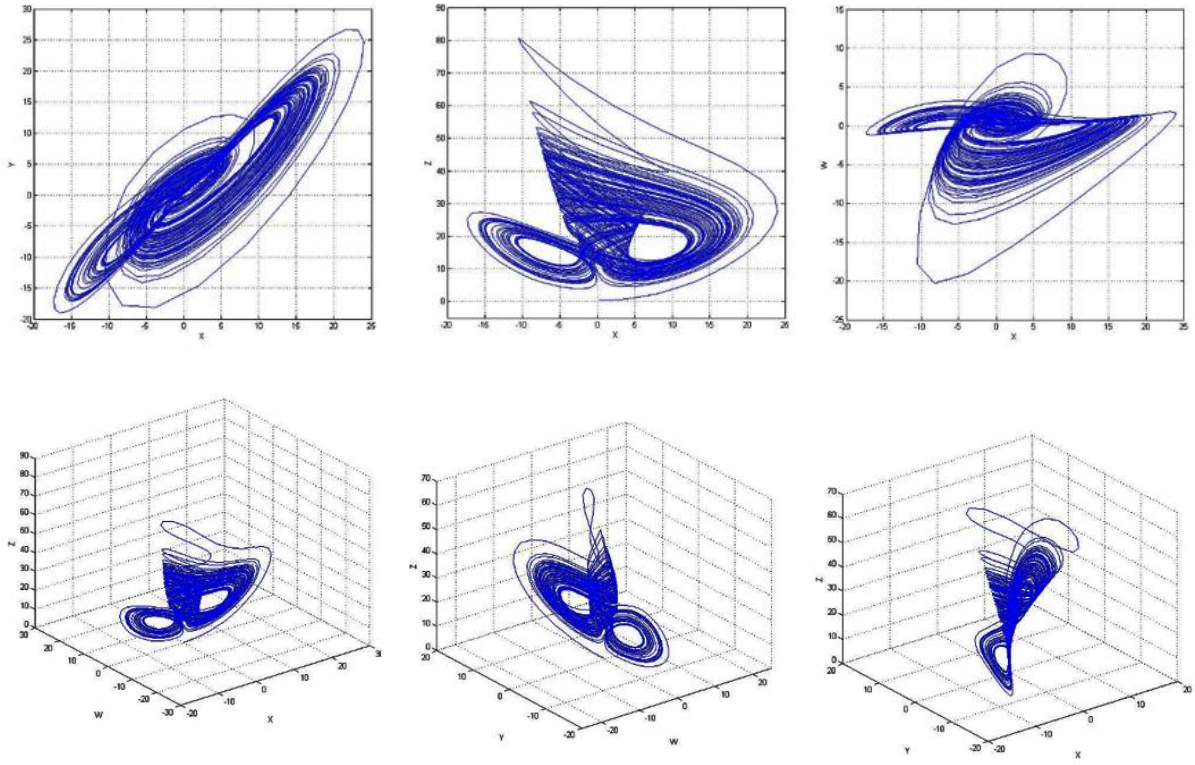
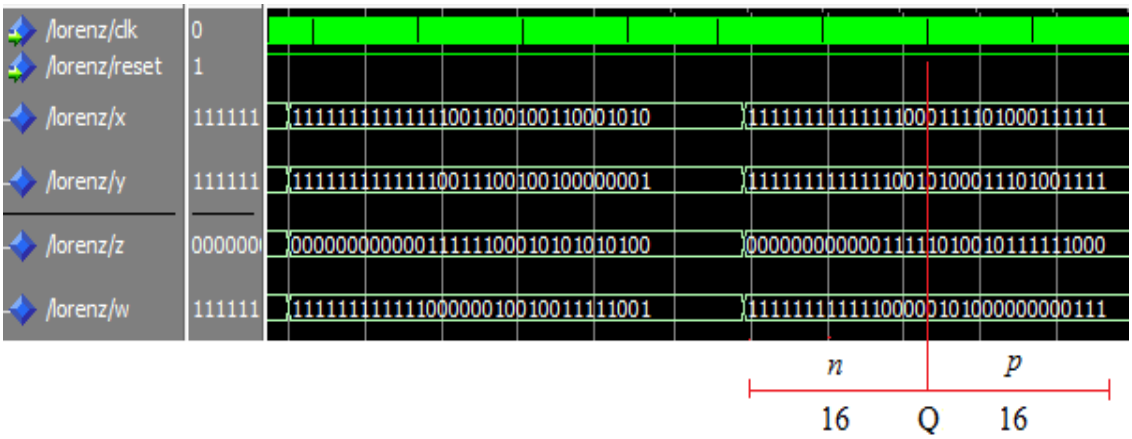


Figure III.2. Attracteurs hyperchaotiques 2D et 3D du système de Chen.

III.3. Génération des clés de chiffrement

Des travaux antérieurs ont démontré qu'il n'est pas judicieux de prendre les séquences binaires des systèmes hyperchaotiques continus directement comme clés de chiffrement. Ce constat est justifié par l'auteur dans [69] par la figure (III.3) qui montre que les parties entières des séquences générées par le système hyperchaotique de Lorenz, implanté sur Field Programmable Gate Array (FPGA), contiennent de longues suites de ''0'' et de ''1'' ce qui dégrade ainsi considérablement les performances statistiques des clés générées.



FigureIII.3. Exemple de séquences binaires (16Q16) générées par le system hyperchaotique de *Lorenz* [69].

Pour contourner ce problème, nous proposons trois modifications à apporter dans les modèles des systèmes chaotiques continus implantés numériquement et qui se résument à :

1. Augmenter la taille des parties fractionnaires pour augmenter la précision : ceci permet de résoudre le problème lié à la dégradation de la dynamique chaotique à cause du problème de la précision finie.
2. Prendre uniquement les parties fractionnaires comme clés de chiffrement : ceci augmente la complexité des clés et élimine aussi le problème de longues suites de ''0'' et de ''1''.
3. Employer les quatre fonctions booléennes non linéaires, utilisées également dans les quatre boucles de la fonction de hachage MD5 [114] et SHA-1[115], comme fonction de couplage. Ces fonctions non-linéaires sont :

$$F(X_1, X_2, X_3) = (X_1 \wedge X_2) \vee (\neg X_1 \wedge X_3)$$

$$G((X_1, X_2, X_4) = X_1 \oplus X_2 \oplus X_4 \quad \text{(III.2)}$$

$$P(X_1, X_3, X_4) = (X_1 \wedge X_3) \vee (\neg X_4 \wedge X_3)$$

$$I(X_2, X_3, X_4) = X_3 \oplus (X_2 \wedge \neg X_4)$$

Où :

$\wedge, \vee, \neg$ et $\oplus$ sont respectivement les opérateurs logiques ET, OU, PAS et OU- exclusif.

En tenant compte de ces différentes recommandations, nous proposons d'appliquer cette technique au système hyperchaotique de Chen. Ainsi le générateur de clés de chiffrement peut être modélisé par le système d'équations suivant :

$$\begin{cases} X1 = Fract(x) \\ Y1 = Fract(y) \\ Z1 = Fract(z) \\ W1 = Fract(w) \end{cases} \quad \text{(III.3)}$$

Avec :

$$Fract(u) = |u| - E(|u|) \quad \text{(III.4)}$$

Où $E(u)$ représente la partie entière de *u* et les *x*, *y*, *z* et *w* représentent les signaux hyperchaotiques de Chen (Eq.III.1).

Notons K_1, K_2, K_3 et K_4 les clés de chiffrement hyperchaotiques données par :

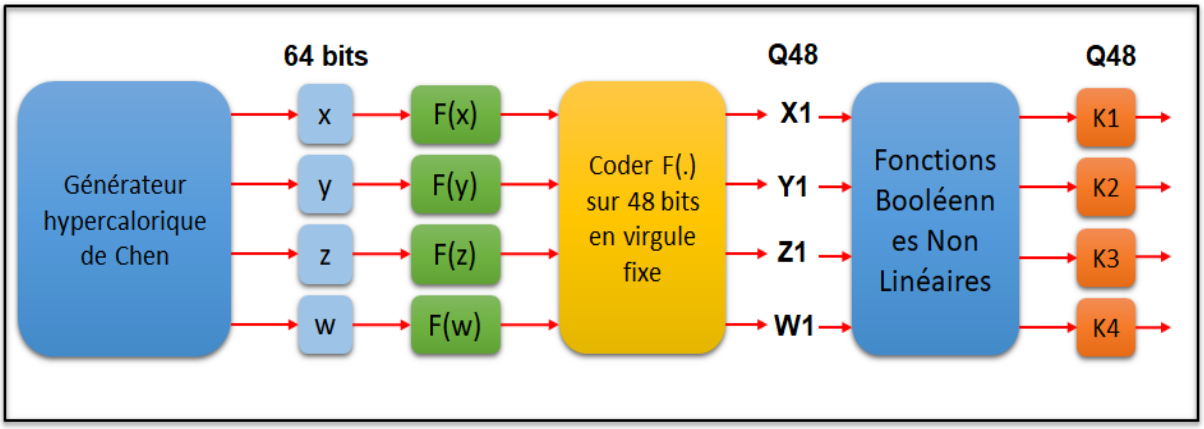
$$K_1 = (X_1 \wedge Y_1) \vee (\neg X_1 \wedge Z_1)$$

$$K_2 = X_1 \oplus Y_1 \oplus W_1 \quad \text{(III.5)}$$

$$K_3 = (X_1 \wedge Z_1) \vee (\neg W_1 \wedge Z_1)$$

$$K_4 = Z_1 \oplus (Y_1 \wedge \neg W_1)$$

Nous pouvons ainsi illustré la génération des clés de chiffrement K_1,K_2, K_3 et K_4 par le schéma bloc de la figure (III.4). Celle-ci présente ce que nous allons appeler Générateur basé-Hyperchaos de Clés de Chiffrement (GHCC).



FigureIII.4. Schéma Bloc du GHCC.

III.3.1. Signaux temporels et attracteurs du GHCC

Les résultats de simulation temporelle, sous Matlab, des signaux hyperchaotiques X_1,Y_1, Z_1 et W_1 pour les mêmes valeurs des paramètres a=37, b=-11, c=-26, d=-3 et e=-38, sont présentés dans les figures (III.5) et (III.6). Ainsi, la figure (III.5) illustre l'évolution temporelle des signaux hyperchaotiques X_1, Y_1, Z_1 et W_1 respectivement alors que la figure (III.6) présente les différents attracteurs hyperchaotiques 2D dans les espaces (X_1-Y_1), (X_1-Z_1) et (X_1-W_1) respectivement.

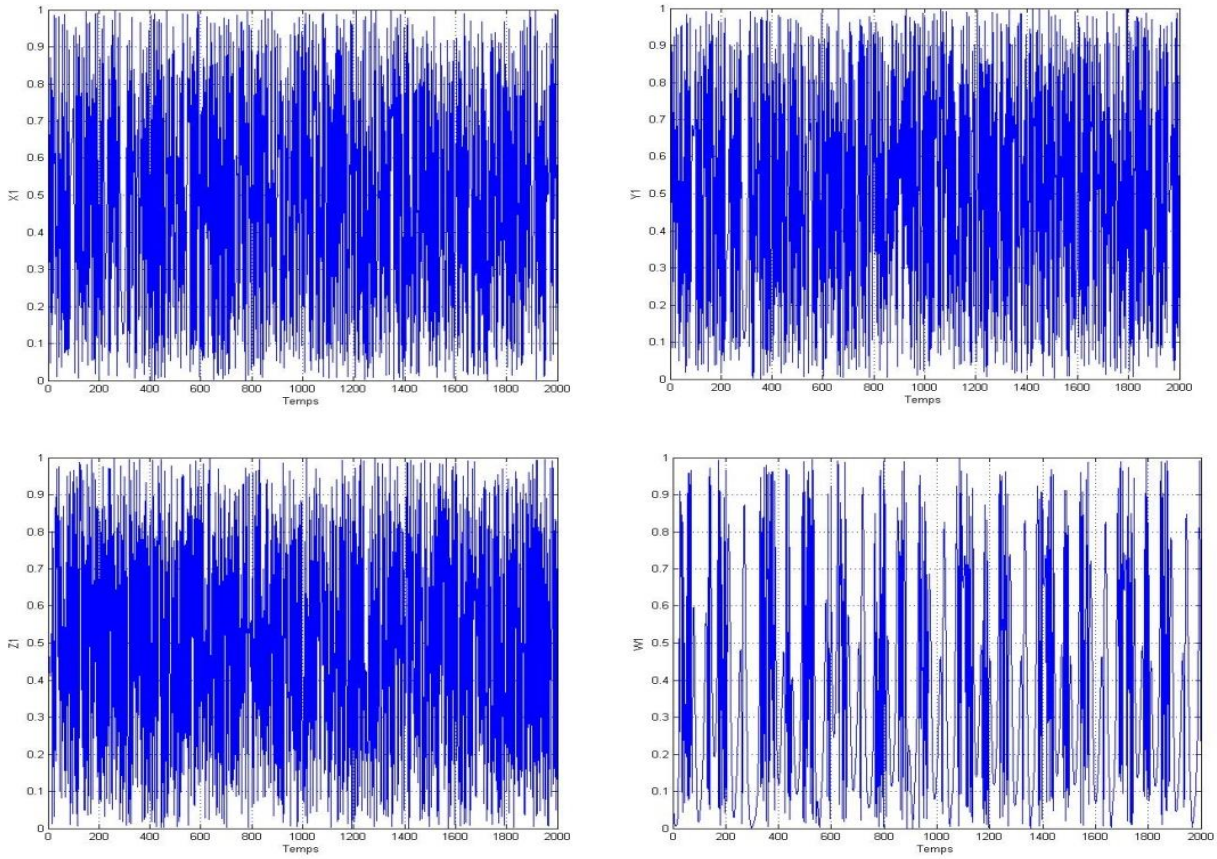


Figure III.5. Signaux temporels des signaux hyperchaotiques $X_1(t)$, $Y_1(t)$, $Z_1(t)$ et $W_1(t)$.

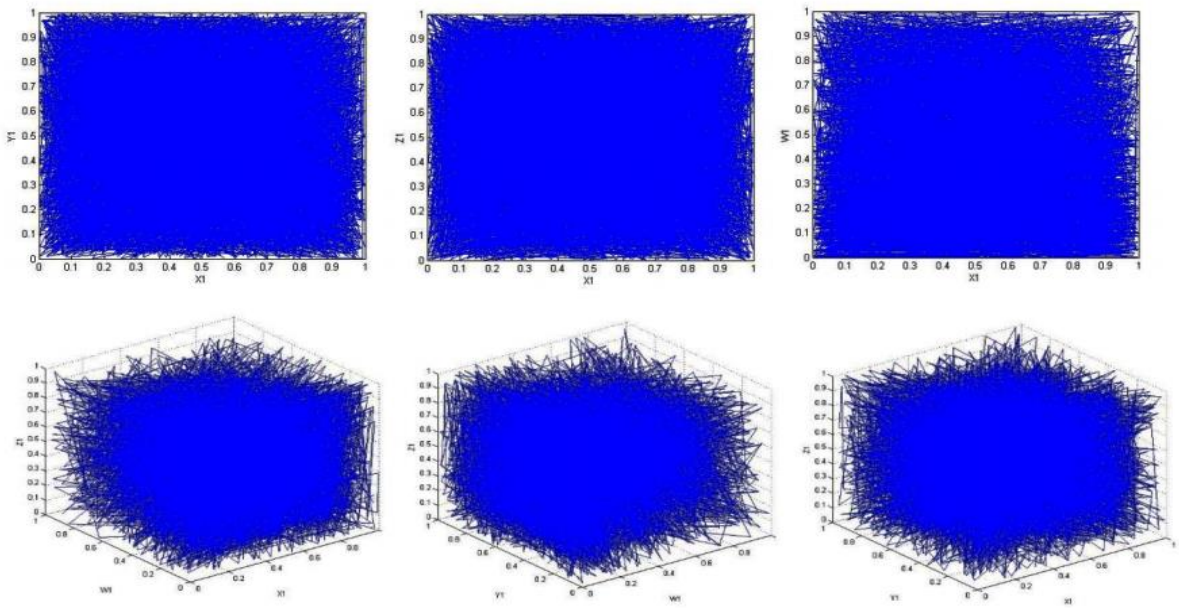


Figure III.6. Attracteurs 2D et 3D de la partie fractionnaire de Chen.

Les résultats de la figure (III.5) montrent bien que les signaux hyperchaotiques X_1, Y_1, Z_1 et W_1 se caractérisent par une grande variabilité temporelle, semblable au bruit, comparativement à ceux du système de Chen (Figure.III.1). De même, les trajectoires de leurs

attracteurs 2D occupent complètement les espaces de phases, contrairement à ceux du système de Chen de la figure (III.2).

III.3.2. Section Poincaré

Pour bien illustrer l'aspect aléatoire des clés générées, nous utilisons l'outil de la théorie du chaos connu sous le nom de section de Poincaré. En considérant la section de Poincaré formée par l'intersection de la trajectoire de phases dans l'espace (*x*, *y*, *z*) (Fig.III.7) avec le plan *x=0*, nous remarquons que le régime hyperchaotique du système de Chen présente un nuage de points occupant une partie importante de la section, indiquant ainsi que les orbites de l'attracteur hyperchaotique étrange ne se coupent jamais (figure III.8). Ceci illustre la propriété de non-périodicité que possèdent les systèmes hyperchaotiques. En revanche, nous remarquons que le nuage de point pour les parties fractionnaires est dispersé et occupe tout l'espace des phases. Ainsi, la figure (III.8) montre bien l'avantage de prendre la partie fractionnaire des signaux hyperchaotiques comme base pour former les clés déchiffrement K_1,K_2, K_3 et K_4.

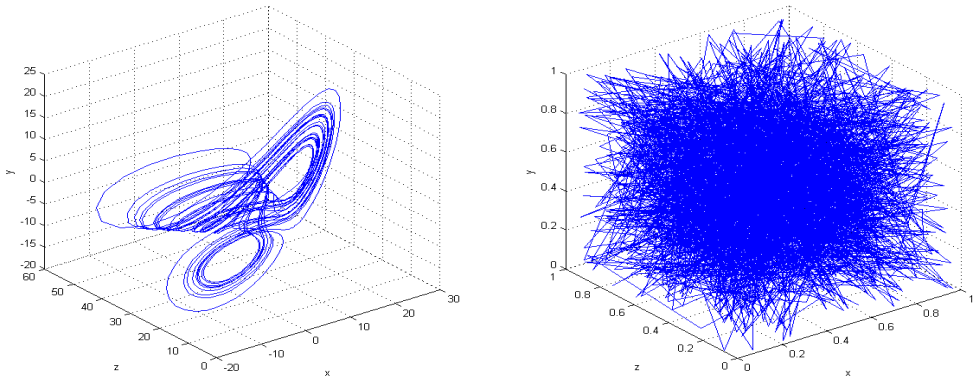


Figure III.7. Attracteur 3D du générateur de Chen (à gauche) et du GHCC (à droite).

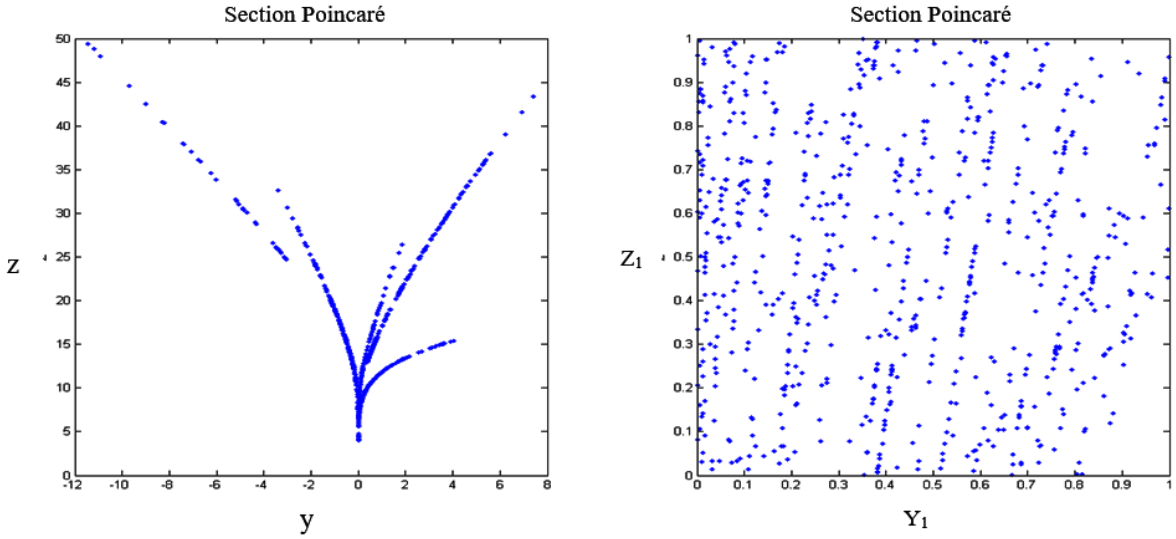


Figure III.8. Section Poincaré du plan de phase (*y*, *z*) et (Y_1, Z_1).

III.3.3. Choix du générateur de Chen

La figure (III.9) représente les fonctions d'autocorrélation des quatre solutions du nouveau générateur hyperchaotique proposé. On remarque bien que les quatre fonctions sont similaires à celle d'un bruit blanc du fait qu'elles présentent des pics en zéro et sont nulles ailleurs. Par conséquent, la prédiction de l'une des solutions à partir de son passé n'est pas possible. Dans la figure (III.10) sont illustrées les différentes fonctions d'inter-corrélation entre les quatre solutions du nouveau générateur hyperchaotique proposé. On remarque que les quatre fonctions sont nulles i.e. les quatre signaux sont complètement décorrélés. Cela signifie qu'aucune dépendance linéaire n'existe entre les signaux et donc, il est impossible d'estimer une solution à partir d'une autre.

En plus des bonnes qualités d'auto et d'inter-corrélation le système hyperchaotique de Chen possède cinq paramètres de contrôle, alors que celui de Lorenz par exemple n'en possède que quatre, de ce fait il est judicieux de choisir le générateur de Chen. Ce point va être justifié davantage à la section suivante.

Il convient également de signaler, que toutes les propriétés des signaux hyperchaotiques X_1,Y_1, Z_1 et W_1 (évolution temporelle, attracteurs, section Poincaré, fonctions d'auto et d'intercorrélations) seront les mêmes pour les clés de chiffrement K_1,K_2, K_3 et K_4. En effet, ce qu'il y a d'important avec les quatre fonctions booléennes non linéaires (citées plus haut) est que si les bits de leurs arguments X_1,Y_1, Z_1 et W_1 sont indépendants, alors les bits du résultat le sont aussi (ce qui est confirmé aussi par les résultats de l'auteur dans [91]).

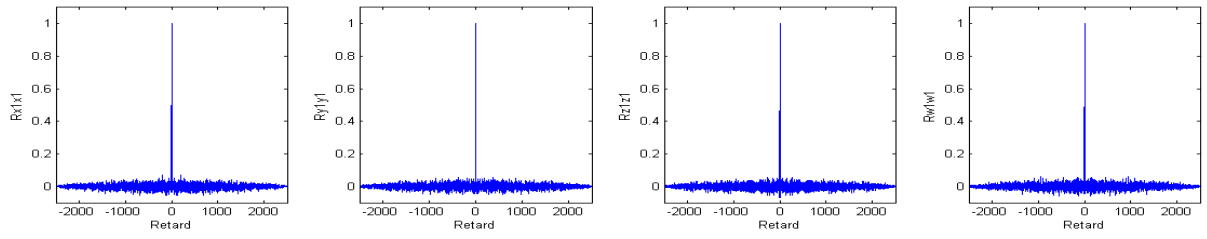
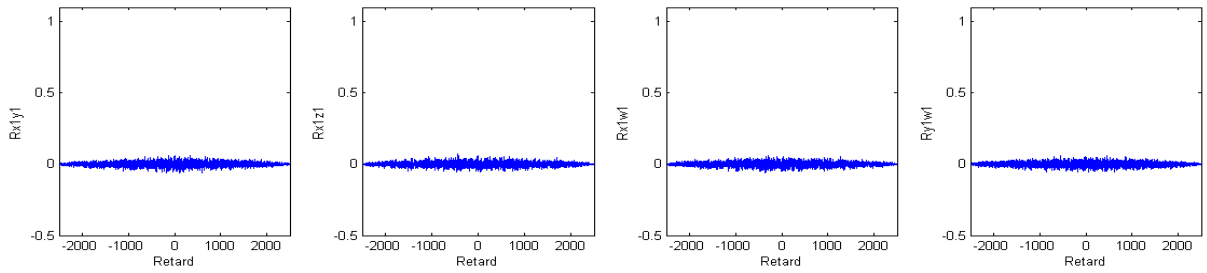


Figure III.9. Les fonctions d'autocorrélations des signaux X_1(t), Y_1(t), Z_1(t) et W_1(t).



FigureIII.10. Les fonctions d'intercorrélations des signaux X_1(t), Y_1(t), Z_1(t) et W_1(t).

III.3.4. Tests statistiques

Pour évaluer le critère aléatoire des clés générées, nous avons effectué une analyse statistique en utilisant la batterie de tests statistiques du NIST. Notre analyse a été faite sur 300 séquences de longueur d'un million de bits (1000000 bits). Théoriquement, il suffit d'effectuer les tests sur 55 séquences d'un million de bits pour avoir des résultats statistiquement corrects [103]. Les résultats des tests sont présentés dans ce qui suit sous forme de tableau, des histogrammes et des figures.

Les Tests		P_value_T			
N°	**Désignation**	K_1	K_2	K_3	K_4
1	Frequency Test	0.188260	0.350485	0.772760	0.266680
2	Block Frequency Test	0.602458	0.407091	0.746572	0.931952
3	Cumulative Sums Test Up	0.766282	0.487885	0.507512	0.262249
4	Cumulative Sums Test Up	0.284959	0.443451	0.148094	0.726503
5	Runs Test	0.401199	0.198690	0.983453	0.902420
6	Long Runs of Ones Test	0.911413	0.165646	0.030130	0.834308
7	Rank Test	0.437274	0.804337	0.05536	0.299251
8	Discrete Fourier Transform Test	0.334538	0.108791	0.137282	0.746572
9	Non-overlapping Template Matching Test	0.699313	0.021505	0.350485	0.699313
10	Overlapping Template Matching Test	0.037566	0.016182	0.061517	0.037566
11	Maurer's "Universal Statistical" Test	0.739918	0.699313	0.124717	0.902420
12	Approximate Entropy Test	0.902420	0.507512	0.019631	0.779188
13	Random Excursions Test	0.900716	0.734017	0.381126	0.566256
14	Random Excursions Variant Test	0.885536	0.415844	0.177424	0.791218
15	Serial Test 1	0.078889	0.554420	0.753185	0.595549
16	Serial Test 2	0.217094	0.474986	0.990369	0.074177
17	Linear complexity Test	0.739918	0.178278	0.029467	0.294431

Tableau III.1. Résultats des tests statistiques du NIST, la p_value_T des différents tests pour chaque séquence.

Le tableau (III.1) présente la valeur de la P_value_T pour évaluer la distribution des *P_value* de chaque test. Pour que la séquence satisfasse le critère d'uniformité, la P_value_T doit être supérieure à 10^{-4}. Nous remarquons que tous l'ensemble des clés K_1,K_2, K_3 et K_4 générées par le GHCC ont passés les 17 tests de NIST avec succès.

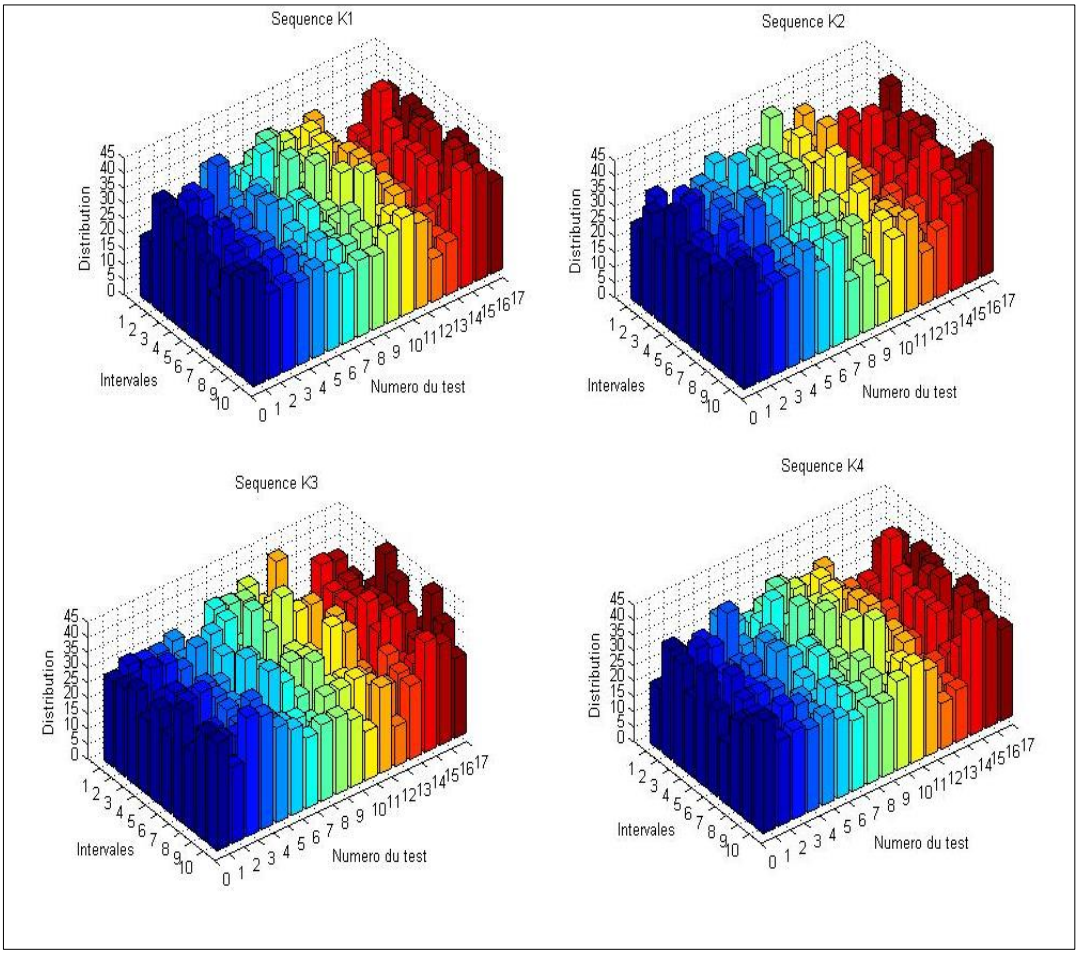


Figure III.11. La distribution des *P_value.*

La distribution des *P_value* est illustrée par des histogrammes (figure III.11). Chaque histogramme contient 10 bâtons, chacun de largeur 0.1 et de hauteur définie par le nombre d'occurrence des *P_value* dans chaque intervalle. L'analyse statistique des clés binaires formées par K_1,K_2, K_3 et K_4, montre clairement que les signaux de sorties du générateur hyperchaotique GHCC ont un bon caractère aléatoire.

Une autre manière de valider l'aspect aléatoire des clés de chiffrement générées en utilisant la proportion des *P_value* qui dépassent le seuil de 10^{-2}[103]. Dans ce contexte, les résultats obtenus pour les clés de 48 bits testées (tableau III.2) valident l'ensemble des 17 tests proposés par la batterie du NIST avec des proportions supérieures à 0.972766 (seuil).

Dans la figure (III.12), sont représentées les proportions des séquences du test. La ligne de bas (en rouge) représente la marge inférieure de l'intervalle de confiance pour l'ensemble des tests de NIST.

Les Tests		***Proportion***			
N°	**Désignation**	K_1	K_2	K_3	K_4
1	Frequency Test	0.9933	0.9967	0.9933	0.9967
2	Block Frequency Test	0.9900	0.9900	0.9933	0.9967
3	Cumulative Sums Test Up	0.9867	0.9967	0.9833	1.0000
4	Cumulative Sums Test Down	0.9900	0.9933	0.9833	0.9967
5	Runs Test	0.9867	0.9867	0.9933	0.9867
6	Long Runs of One's Test	0.9933	0.9867	0.9967	0.9867
7	Rank Test	0.9867	0.9933	0.9933	0.9800
8	Discrete Fourier Transform Test	1.0000	1.0000	0.9933	0.9967
9	Non-overlapping Template Matching Test	0.9833	0.9967	0.9867	0.9833
10	Overlapping Template Matching Test	0.9833	0.9832	0.9833	0.9833
11	Maurer's "Universal Statistical" Test	0.9933	0.9933	0.9900	0.9767
12	Approximate Entropy Test	0.9833	0.9933	0.9833	0.9867
13	Random Excursions Test	0.9948	0.9944	0.9895	0.9896
14	Random Excursions Variant Test	0.9948	0.9888	0.9948	0.9844
15	Serial Test 1	0.9900	0.9833	0.9867	0.9833
16	Serial Test 2	0.9933	0.9967	0.9900	0.9867
17	Linear complexity Test	0.9967	0.9967	0.9933	0.9867

Tableau III.2. Résultats des tests du NIST, proportion des clés qui passent chaque test avec succès.

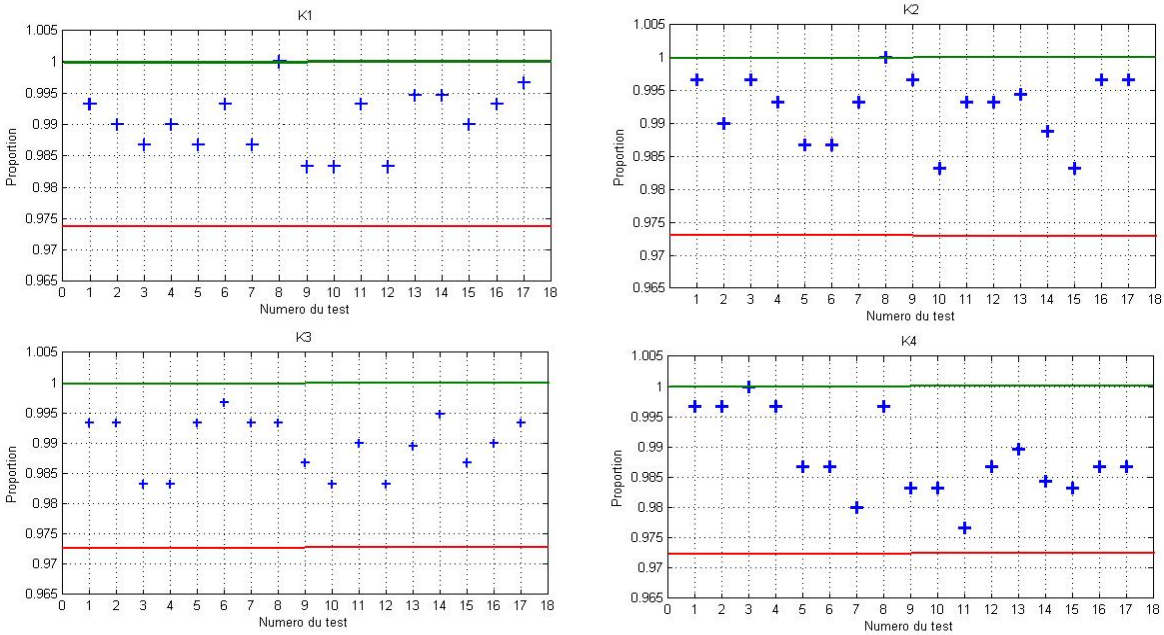


Figure III.12. Proportions des séquences.

III.4. Structure du cryptosystème proposé

Le schéma global du cryptosystéme proposé pour le chiffrement/déchiffrement est illustré par la figure (III.13). L'idée principale consiste à chiffrer le texte en claire avec les clés de notre générateur GHCC. Le message chiffré est rebouclé (feedback) vers l'entrée pour perturber le générateur et ainsi complexifier au maximum sa dynamique chaotique. En plus, ce procédé va nous assurer le phénomène d'avalanche et de diffusion. En effet, un petit changement dans le texte ou l'image en clair provoquera un grand changement sur le texte ou l'image chiffrée.

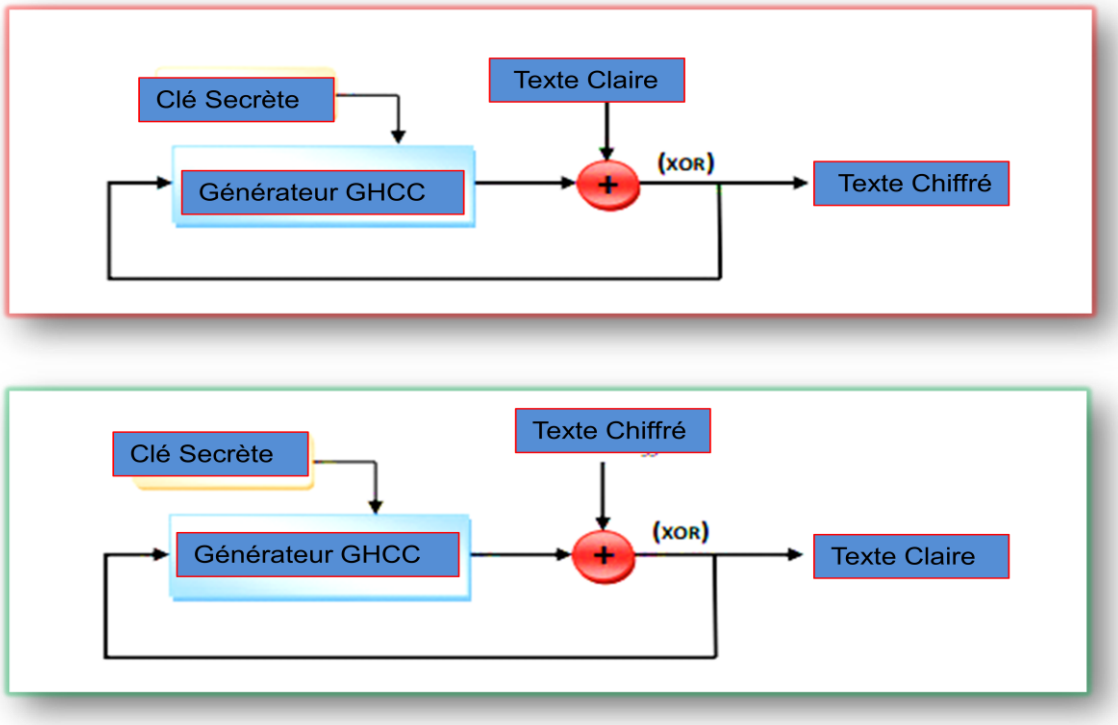


Figure III.13. Structure générale du cryptosystème.

Ainsi cette structure est composée essentiellement du GHCC comme générateur de clés de chiffrement et d'une rétroaction (feedback) de la sortie vers l'entrée selon la relation suivante :

$$\begin{cases} x_{n+1} = x_n + (tab[n] \,\%\, 0.02) \\ y_{n+1} = y_n + (tab[n] \,\%\, 0.01\,) \\ z_{n+1} = z_n + (tab[n] \,\%\, 0.04\,) \\ w_{n+1} = w_n + (tab[n] \,\%\, 0.03\,) \end{cases} \quad \text{(III. 6)}$$

Où *tab*[*n*] représente l'élément du tableau chiffré à l'itération *n.* L'usage de l'opérateur modulo permet de maintenir et de ne pas dégrader les propriétés statistiques de la dynamique chaotique et ce en limitant toujours l'amplitude de la perturbation à un niveau très faible. En plus, la rétroaction assure aussi un effet perturbateur des orbites chaotiques, ce qui engendre une dynamique chaotique plus complexe et plus riche.

III.5. Analyse de sécurité d'images chiffrées

La mesure fondamentale pour évaluer la qualité d'un cryptosystème est sa capacité à résister à toute tentative d'accès non autorisée pour soustraire l'information. Cette mesure est appelée sécurité. Le succès d'une méthode de cryptage est sa résistance face aux diverses attaques telles que l'attaque par force brute, l'attaque statistique et l'attaque différentielle. Dans ce contexte, nous avons effectué quelques tests de sécurité les plus connus et utilisés tels que l'espace des clés, le calcul d'entropie, l'autocorrélation, les histogrammes et la sensibilité à la clé. Ces tests ont été appliqués sur différentes images de référence telles que Lena, Cameraman, Barbara, Peppers, Mandrill et Airplane. Ces dernières sont chiffrées par l'opérateur XOR (masque jetable).

III.5.1. Analyse différentielle

Le tableau (III.3) donne les résultats de l'analyse différentielle du cryptosysteme proposé. L'analyse consiste en fait à crypter deux images claires différentes seulement par un pixel. La première est l'image en claire originale et la seconde est obtenue après le changement d'un bit dans le premier pixel de la première image. Notons que ces deux images sont chiffrées avec la même clé pour générer les deux images chiffrées correspondantes I_1 et I_2.

Image	Taille de l'image	NPRC (%)	UACI (%)	DH (%)
Lena	512x512x3	99.5971	33.405	0.499
Lena	512x512x1	99.6063	33.508	0.500
Barbara	512x512x3	99.6269	33.470	0.500
Pepper	512x512x3	99.605	33.478	0.499

Tableau III.3. Résultats d'analyse de l'attaque différentielle suivant les métriques NPCR, UACI et DH.

Les résultats représentés sur le tableau (III.3) montre bien la robustesse du cryptosystème proposé contre l'attaque différentielle. En effet, on constate bien que les valeurs du NPCR et de l'UACI sont très proches, respectivement, des valeurs optimales de 99,61% et de 33,46%

et que les valeurs des DH sont presque égales à 50%, ce qui indique que le cryptosysteme proposé est très sensible aux petits changements dans l'image originale, ce qui se répercute fortement sur l'image chiffrée. A titre comparatif, nous donnons dans le tableau (III.4) les valeurs de NPCR, UACI et de DH, de quelques cryptosystèmes basés chaos connus dans la littérature. On voit bien que les résultats obtenus sont assez comparables.

Algorithme	Image	UACI (%)	NPCR (%)	DH
Proposé	Lena	33.405	99.597	0.499483
Crypto-C [106]	Lena	33.462	99.609	0.499912
Ref.[117]	Lena	33.4618	99.6085	-
Proposé	Peppers	33.478	99.605	0.499
Ref.[116]	Peppers	33.4211	99.6185	-
Proposé	Barbara	33.470	99.6269	0.500401
Ref.[118]	Barbara	33.48	99.60	0.4995

Tableau III.4. Comparaison du NPRC, UACI et DH avec quelques références.

III.5.2. Entropie

Nous donnons dans le tableau (III.5) les résultats de calcul d'entropie obtenus pour quelques images. D'après les résultats nous remarquons que la valeur de l'entropie est très proche de 8, cela signifie que le système de chiffrement est robuste face à une attaque d'entropie.

Image	**Entropie (Image claire)**	**Entropie (Image chiffrée)**
Airplane	6.66390	7.99978
Mandrill	7.69575	7.99871
Lena	5.68222	7.99980
Peppers	7.66982	7.99974

Tableau III.5. Résultats de test d'entropie**.**

III.5.3. Espace des clés

Le GHCC proposé basé sur le générateur de Chen (Eq.III.1) possède cinq paramètres de contrôle *a*, *b*, *c*, *d* et *e* ainsi que quatre conditions initiales x_0, y_0, z_0, w_0. Nous pouvons ajouter le pas d'intégration *h* de la méthode RK-4 utilisée pour résoudre le système d'équation du générateur. Ainsi, nous obtenons un total de dix paramètres codés sur 64bits. Donc l'espace des clés *H* du cryptosystème proposé peut être calculé comme suit :

$$H(a, b, c, d, e, h, x_0, y_0, z_0, w_0) = (2^{64})^{10} = 2^{640} \quad \text{(III. 7)}$$

Dans la littérature, pour qu'un cryptosystème basé-chaos résiste à l'attaque par force brute, il faut que son espace des clés doit être supérieur ou égal à 2^{128} [109].Donc l'espace des clés de notre cryptosystème est assez large pour résister à l'attaque par force brute. Ce qui démontre l'avantage de choisir le système hyperchaotique de Chen.

III.5.4. Sensibilité des clés

On considère les deux images (Barbara, Airplane), nous avons évalué le critère de la sensibilité de la clé par deux approches :

- Sensibilité aux conditions initiales,
- Sensibilité aux paramètres.

Le résultat du test de sensibilité aux conditions initiales de la figure (III.14) montre que le déchiffrement avec la valeur y_0=0.2+10^{-16} de l'image chiffrée(Barbara) avec la valeur y_0=0.2 ne permet pas la reconstitution correcte de l'image originale. De même pour le test de sensibilité aux paramètres, pour un déchiffrement avec la valeur *a*=37+10^{-14} de l'image chiffrée (Airplane) avec la valeur *a*=37. Donc, notre cryptosystème est extrêmement sensible à la moindre variation soit dans les conditions initiales soit dans les paramètres de contrôle.



Figure III.14. Chiffrement d'image : Sensibilité aux conditions initiales (haut) et aux paramètres (bas).

III.5.5. Histogramme

Pour voir la diffusion de l'information dans les images cryptées, nous avons tracé trois histogrammes des images en couleur suivantes Airplane, Lena et Pepper qui sont représentés dans les figures (III.15), (III.16) et (III.17) respectivement.

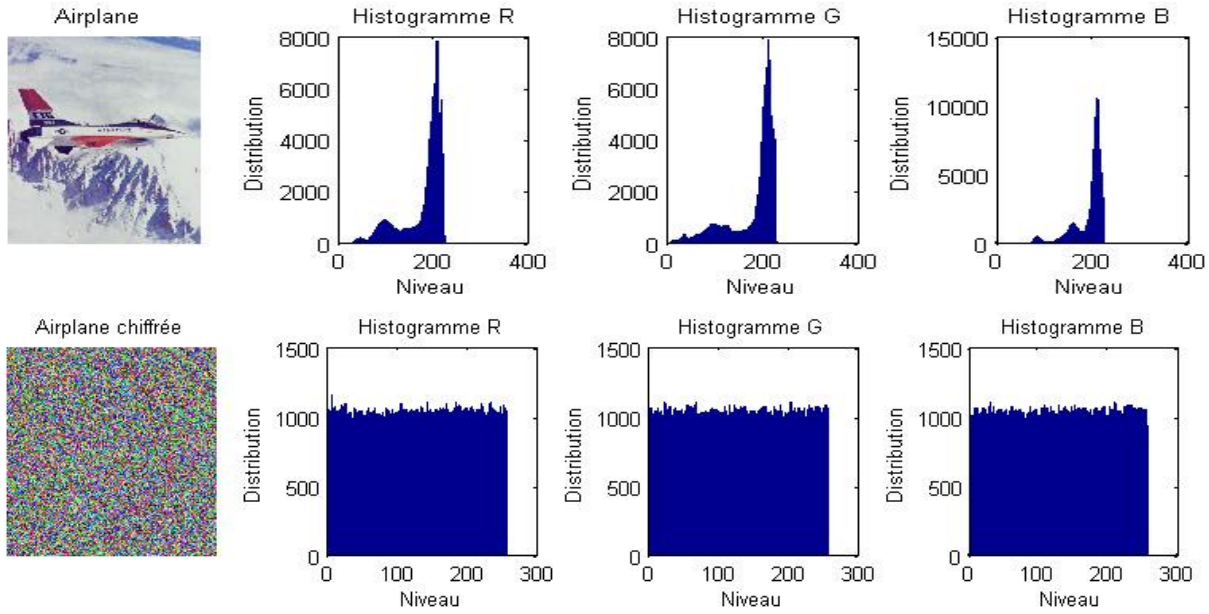
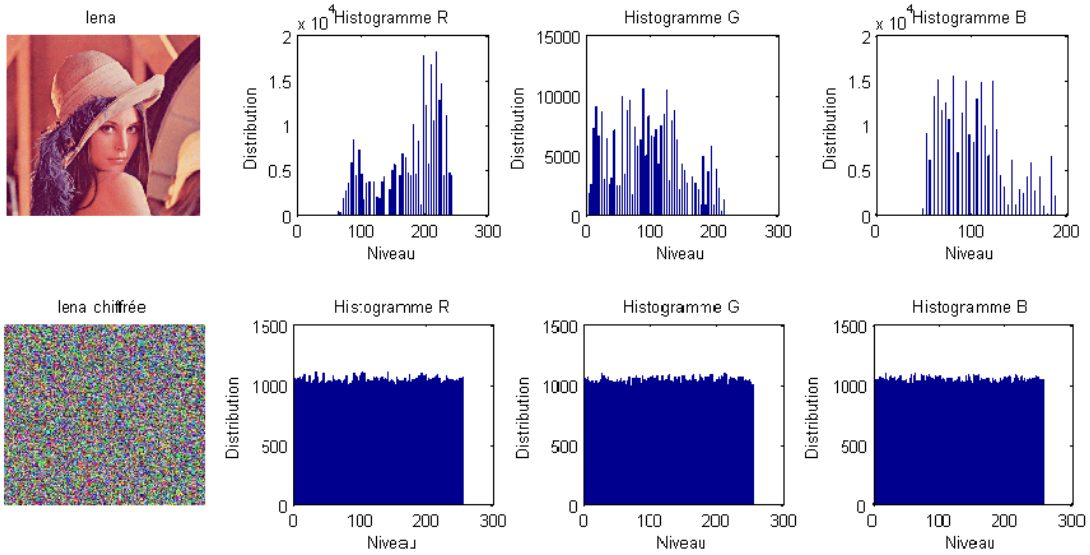


Figure III.15. Chiffrement d'image : Analyse d'histogrammes d'image en couleur (Airplane).



FigureIII.16. Chiffrement d'image : Analyse d'histogrammes d'image en couleur (Lena).

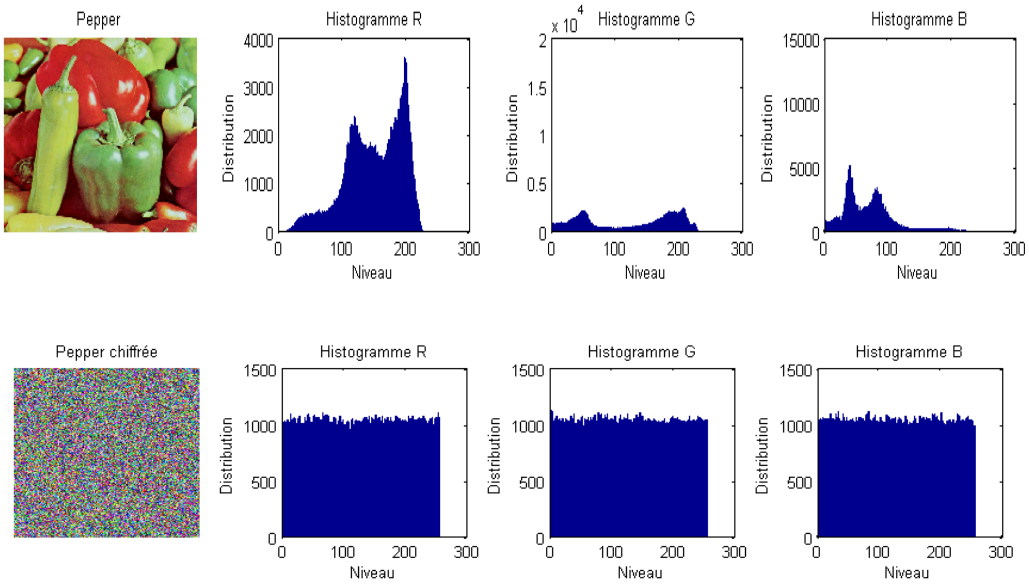


Figure III.17. Chiffrement d'image : Analyse d'histogrammes d'image en couleur (Pepper).

Nous remarquons que les histogrammes des images chiffrées présentent des niveaux d'amplitudes relativement plats par rapport à ceux des images originales, ce qui confirme leurs aspects aléatoires. Cependant, le test visuel seul n'est pas suffisant pour affirmer l'uniformité de la distribution. Pour mesurer cette uniformité statistiquement, nous appliquons le test de chi-carré défini précédemment dans le chapitre II.

Nous donnons dans le tableau (III.6) les résultats de test chi-carré pour les images Lena, Pepper, Airplane et Barbara.

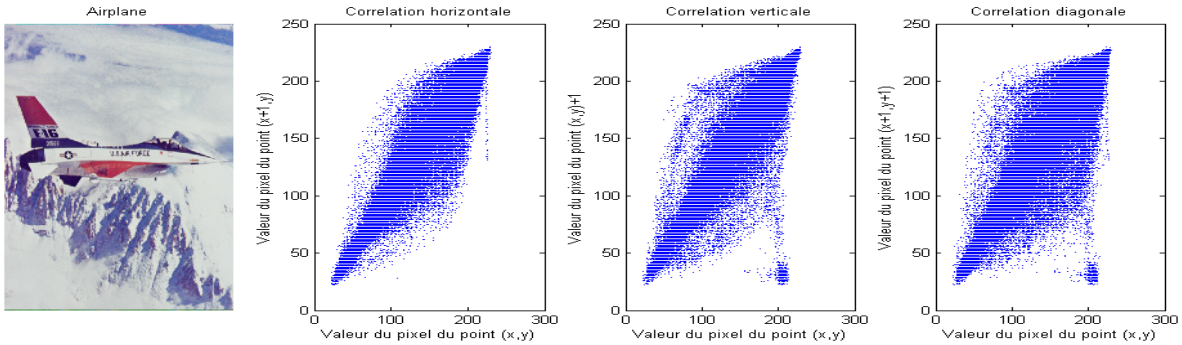
Image chiffrée	Test chi-carré
Lena 512*512*3	268.26367
Pepper 512*512*3	242.47461
Airplane 512*512*512	266.1797
Lena 512*512*1	241.04102
Barbara 512*512*1	267.39844

Tableau III.6. Test de chi-carré sur différentes images.

On voit bien sur le tableau que les valeurs des chi-carré calculées χ^2_{exp} sont inférieures à la valeur du chi-carré théorique $\chi^2_{th} = 293.2478$. Ainsi les résultats obtenus confirment bien statistiquement, l'uniformité des histogrammes des images chiffrées, ce qui montre que le cryptosystème proposé est robuste face à l'attaque d'histogramme.

III.5.6. Corrélation des pixels adjacents

Les figures (III.18), (III.19) et (III.20) montrent la corrélation des pixels adjacents dans les trois directions pour les images claires et chiffrées Airplane, Peppers et Mandrill respectivement. Nous remarquons que les images claires possèdent des nuages de point qui se présentent sous formes de droites. Ceci indique qu'il y a une dépendance linéaire entre les pixels adjacents de ces dernières, et ceci dans les trois directions. Par conséquent, la valeur d'un pixel peut être estimée par une interpolation linéaire entre ces pixels adjacents. Par contre, les nouveaux nuages de points obtenus pour les images chiffrées sont totalement dispersés. Ce qui signifie que les pixels adjacents sont complètement décorrélés. Par conséquent, aucune estimation de la valeur d'un pixel à partir des valeurs de ses pixels adjacents n'est possible. Ceci indique l'efficacité du cryptosystème proposé.



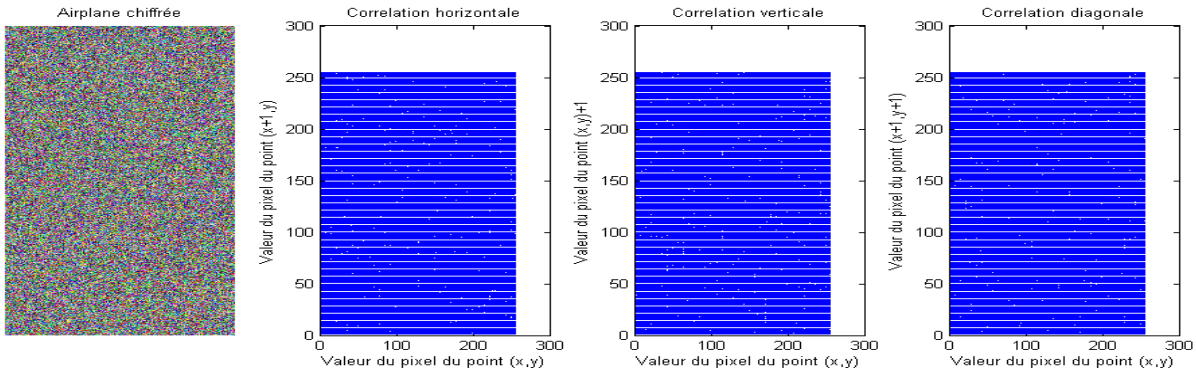


Figure III.18. Chiffrement de l'image Airplane: corrélation des pixels adjacents dans l'image originale et chiffrée.

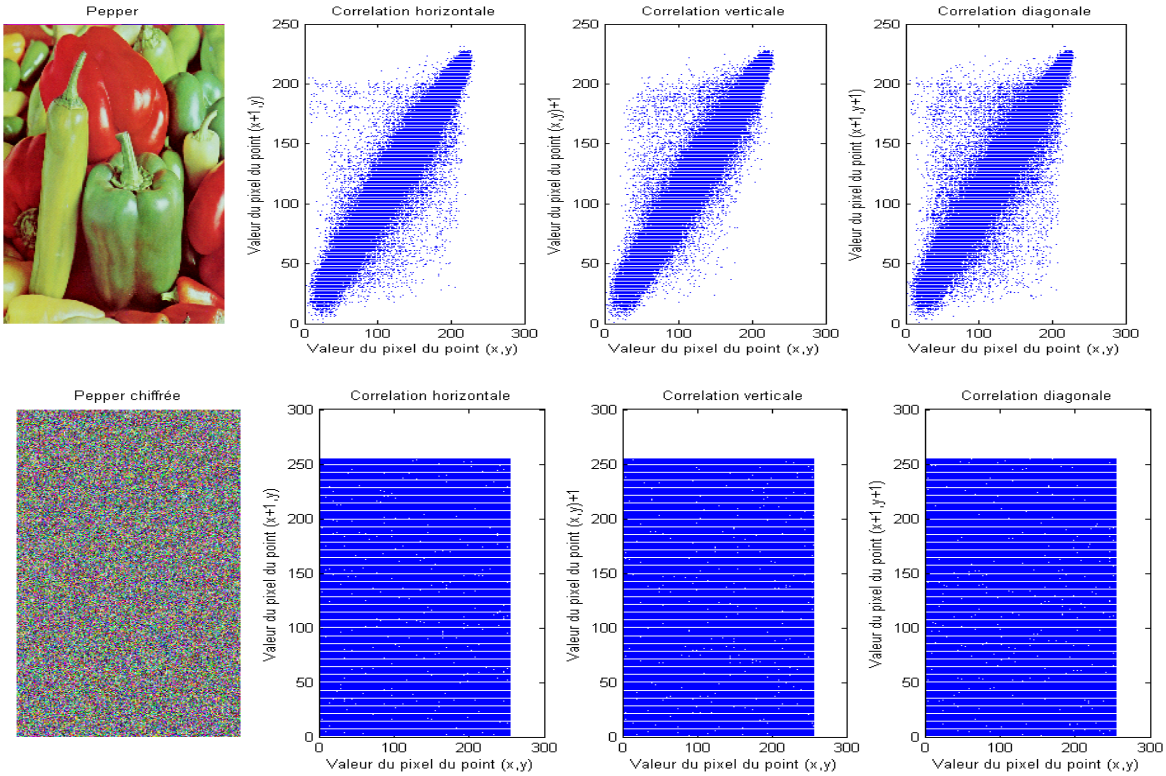
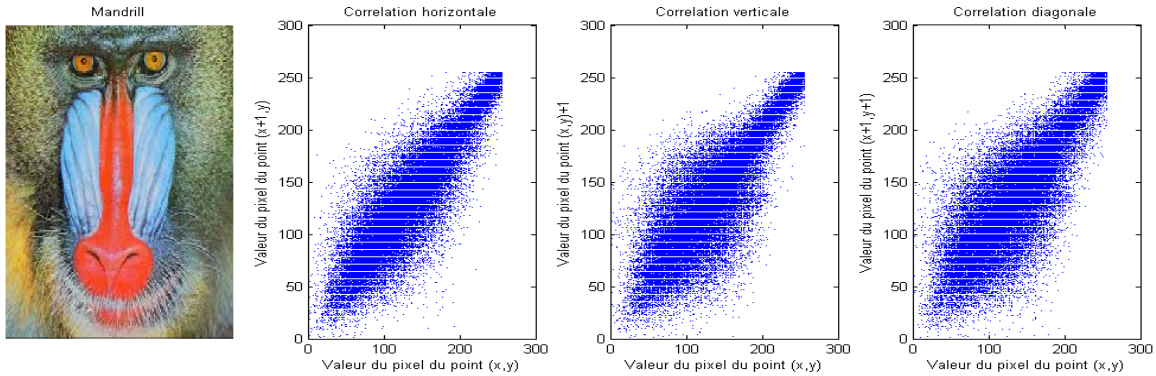


Fig.III.19. Chiffrement de l'image Peppers: corrélation des pixels adjacents dans l'image originale et chiffrée.



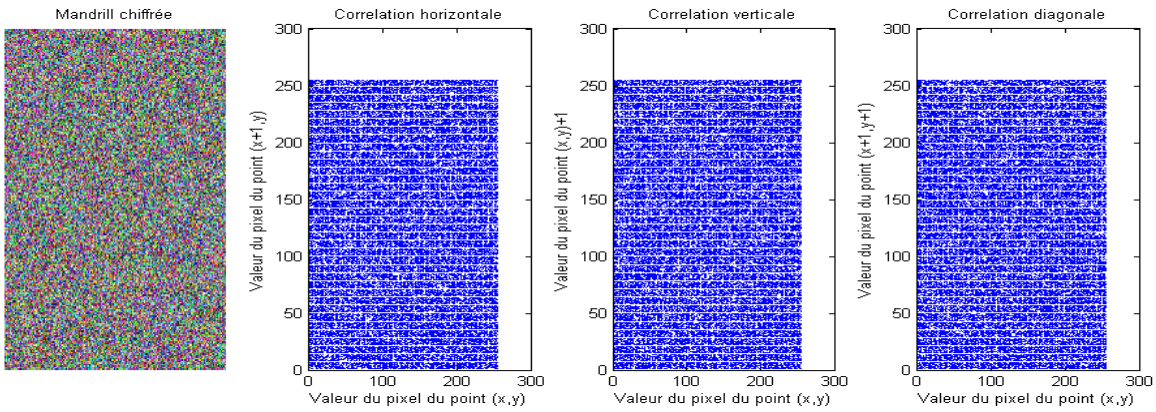


Figure III.20. Chiffrement de l'image *Mandrill* : corrélation des pixels adjacents dans l'image originale et chiffrée**.**

Le tableau (III.7) donne les coefficients de corrélation des pixels adjacents, dans les trois directions (horizontale, verticale et diagonale), calculés pour quelques images avec différents cryptosystèmes y compris le nôtre. Les résultats obtenus sont comparables.

Algorithme	**Image**	**Correlation original image**			**Correlation ciphered image**		
		H	**V**	**D**	**H**	**V**	**D**
Proposé	Lena	0.9752	0.98530	0.96483	-0.00022	-0.00043	-0.00162
Ref.[118]	Lena	0.9850	0.9719	0.9593	0.0093	0.0032	-0.0019
Ref.[119]	Lena	0.9685	0.96252	0.96462	0.0005	-0.0025	0.0028
Ref.[120]	Lena	0.9859	0.9741	0.9618	0.0052	00.0112	0.0034
Proposé	Barbara	0.8927	0.95421	0.88393	-0.00003	0.00129	-0.00016
Ref.[118]	Barbara	0.9542	0.8927	0.8839	0.0022	0.0053	-0.0005
Ref.[121]	Barbara	0.9588	0.8991	0.8733	-0.0002	0.0081	-0.0011

Tableau III.7. Les coefficients de corrélation des pixels adjacents.

III.6. La synchronisation chaotique dans un réseau informatique

Dans cette section, nous allons aborder le plus grand problème des cryptosystèmes basé-chaos qui est la synchronisation chaotique. Nous avons affirmé dans le chapitre précèdent que dans la majorité des cas, les cryptosystèmes basés-chaos sont des Stream Ciphers, c-à-d, ils réalisent un chiffrement par flot. Dans ce type de chiffrement, il faut générer les mêmes clés, utilisées pour le chiffrement, au niveau du récepteur, c'est le principe de chiffrement symétrique. Sachant que les systèmes chaotiques sont très sensibles à la moindre variation des valeurs de leurs paramètres et conditions initiales. Alors comment peut-on assurer ceci ? Dans ce qui suit, nous présentons notre solution à ce problème. Mais avant, nous commençons par donner une brève explication du principe de la synchronisation chaotique et plus précisément l'origine de son problème.

III.6.1. Principe de la synchronisation chaotique

En 1990, Pecora et Carroll ont démontré pour la première fois que certains systèmes chaotiques possèdent la propriété d'auto-synchronisation [122]. Ils affirment qu'un système chaotique autonome peut se décomposer en sous-systèmes qui peuvent se synchroniser s'ils sont stables et couplés avec un signal de couplage commun en configuration Maître/Esclave. Cette technique nommée Synchronisation Chaotique Identique (SCI) repose sur le couplage en configuration *Maître-Esclave* de deux systèmes chaotiques identiques, c-à-d deux systèmes possédants les mêmes conditions initiales et les mêmes valeurs de paramètres. Le système *Maître* force le système *Esclave* à le suivre en lui injectant continuellement les valeurs de l'une de ses variables d'état $x(t)$, $y(t)$ ou $z(t)$. Pour que le système *Esclave* suive le système *Maître*, il faut le décomposer en deux sous-systèmes comme le montre la figure (III.21). Le *sous-système*2 du système *Esclave* utilise le signal transmis $x(t)$ par le système *Maître* pour générer les signaux chaotiques $y(t)$ et $z(t)$. Ces deux derniers sont injectés dans le *sous-système*1 qui génère à son tour une copie identique du signal chaotique $x(t)$ généré et transmis par le système *Maître* [69].

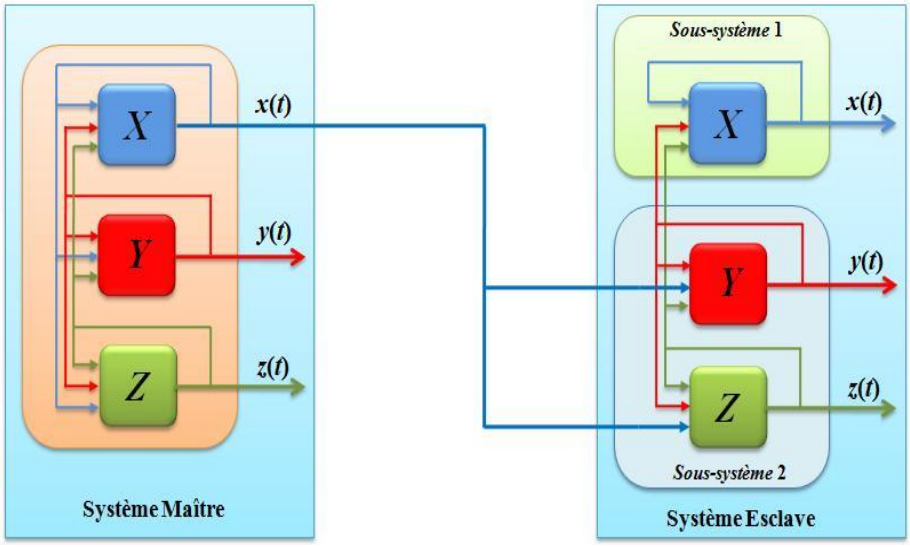


Figure III.21. Schéma de principe de la SCI[69].

III.6.2. Problème de la synchronisation chaotique

Le problème de la SCI de Pecora et Carroll se focalise sur le signal de couplage continu assurant la configuration *Maître/Esclave* (Fig.III.21). En effet, si un bruit vient s'ajouter à ce signal, il est évident que le système *Esclave* générerait un signal $x(t)$ totalement différent de celui de l'émetteur à cause de l'extrême sensibilité des systèmes chaotiques à la moindre variation des valeurs de ses conditions initiales. Pour cette raison, la synchronisation chaotique reste à ce jour le grand problème des communications basées-chaos analogiques qui

est du essentiellement au bruit du canal de transmission. Il est à noter que la majorité des techniques de synchronisation chaotique proposées dans la littérature sont basées sur le principe de couplage de la SCI.

Après avoir bien saisi ce problème et pour trouver une solution robuste dans notre domaine d'application, qui est la cryptographie basée-chaos, nous pouvons ainsi résumer le problème de la synchronisation aux deux points suivants :

- **P1** : Comment peut-on générer des clés de déchiffrement, au niveau du récepteur, identiques à celles du chiffrement utilisées au niveau de l'émetteur ?
- **P2** : Si la solution est trouvée pour **P1**, alors comment déchiffrer avec la bonne clé ? Et comment mettre en synchronisme celle-ci avec la donnée chiffrée correspondante ?

Dans la sous-section suivante, nous apportons nos propres solutions à ce problème.

III.6.3. Solutions proposées

- **Pour le problème P1**, nous travaillons sur la génération numérique du chaos en utilisant des calculateurs. Dans ces derniers, il est clair que la génération des clés hyperchaotiques identiques par deux générateurs hyperchaotiques distancés nécessite deux conditions :

1) Il faut que les générateurs soient identiques (Chen par exemple) et utilisent la même méthode de résolution numérique (par exemple RK4),

2) Il faut s'assurer que les deux générateurs aient les mêmes valeurs des paramètres et des conditions initiales.

- **Pour le problème P2**, la solution à ce problème réside sur l'exploitation du protocole TCP qui permet d'assurer le transfert des données de façon fiable puisqu'il possède un système d'accusé de réception permettant aux deux machines de s'assurer de la bonne réception mutuelle des données. Lors de l'émission d'un segment, un **numéro d'ordre** (appelé aussi numéro de séquence) est associé. A la réception d'un segment de donnée, la machine réceptrice va retourner un segment de donnée dont le drapeau ACK est à 1 (afin de signaler qu'il s'agit d'un accusé de réception) accompagné d'un numéro d'accusé de réception égal au numéro d'ordre précédent comme le montre la figure (III.22). De plus, grâce à une minuterie déclenchée, dès la réception d'un segment au niveau de la machine émettrice, le segment est réexpédié lorsque le temps imparti est écoulé, car dans ce cas la machine émettrice considère que le segment est perdu [123]. Toutefois, si le segment n'est pas perdu et qu'il arrive tout de

même à destination, la machine réceptrice saura grâce au numéro d'ordre qu'il s'agit d'un doublon et ne conservera que le dernier segment arrivé à la destination. Donc le protocole TCP assure d'une manière efficace l'arrivée de toutes les trames dans leur bon ordre. Donc, à la réception, le GHCC génère des clés de déchiffrement uniquement à l'arrivée d'une trame, cela résout le problème (**P2**) car l'ordre d'arrivée des trames et la condition de génération des clés permet à chaque trame d'être xorée par les bonnes clés.

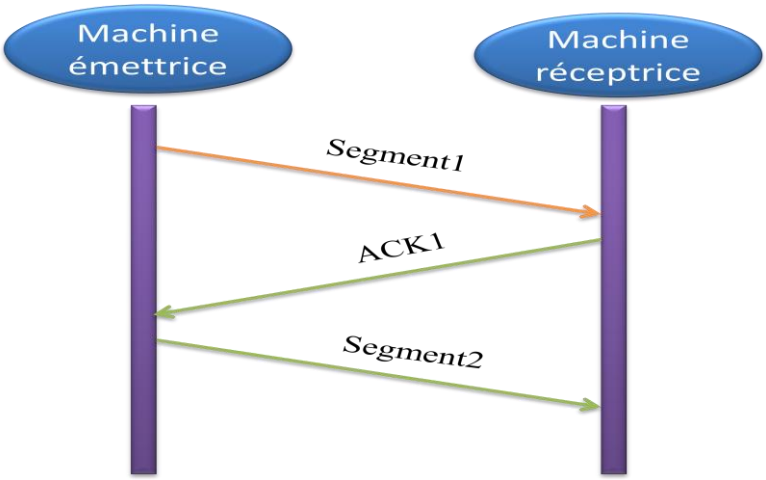


Figure III.22. Phase de transfert de données par TCP.

III.6.4 Fonction de permutation

En cryptographie la technique de diffusion est souvent réalisée par des permutations de blocs ou des permutations binaires. Pour notre travail, nous proposons d'intégrer une fonction de permutations simple et efficace donnée par les auteurs dans [124]. Ainsi pour un tableau ***A*** dé taille ***L*** l'algorithme de permutation ou de mélange de ses éléments est comme suit :

- Initialisation du tableau ***A***
- ***M*=*L***
- Pour *i*=1 ; *i*<***L*** ; *i*++ faire :
  - Générer nombre aléatoire ***R*** (GHCC)
  - ***T*= *R mod M***
  - Permuter **(*A*[*T*], *A*[*M*])**
  - ***M*=*M-1***

La figure (III.23) illustre clairement le procède de permutation pour un tableau de taille ***L***=5.

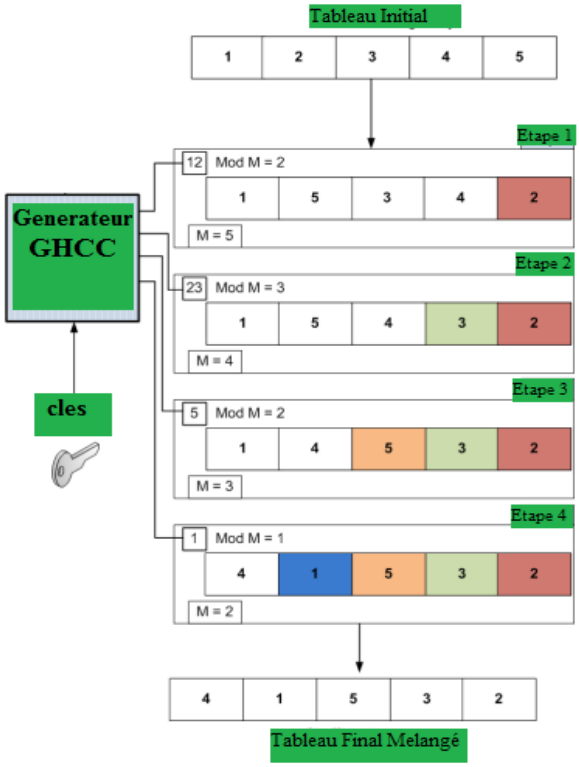


Figure III.23. Illustration du procédé de permutation pour un tableau de taille ***L*=5**.

Pour la suite de ce chapitre, nous définissons la fonction PERMUT(***L***), basée sur l'algorithme décrit plus haut, qui sera utilisée comme couche de permutation pour les tableaux de taille ***L.***

III.7. Approche et déploiement de l'architecture proposée

Le réseau de capteurs sans fil (RCSF) conçu vise la collecte et la transmission sécurisée et efficace de données entre les points d'origines et de destinations. Les données recueillies par chaque capteur sont stockées sur un serveur central, permettant un traitement ultérieur après leur vérification. Ainsi, la sécurisation de la transmission des données s'avère cruciale pour les applications de domotique. Il est essentiel de satisfaire à diverses exigences de sécurité, telles que la confidentialité, l'authentification, l'intégrité, la disponibilité, et la fraîcheur des données [125,126]. Notre contribution consiste à proposer une solution sécurisée basée sur le RCSF, utilisant le générateur de clés GHCC, pour répondre à ces exigences sécuritaires. Le capteur DHT22, mesurant la température et l'humidité, a été sélectionné pour la collecte de données environnementales, constituant ainsi notre base de données expérimentale. La figure (III.24) illustre les composants utilisés dans l'assemblage des nœuds capteurs : le capteur DHT22 est présenté en figure (III.24.b), et le module Wi-Fi (shield Wi-Fi pour Arduino) est visible en figure (III.24.c), tous deux intégrés à une carte Arduino Due, comme le montre la figure (III.24.a). L'Arduino Due, équipée du microcontrôleur ARM Cortex-M3 SAM3X8E d'Atmel,

est notre choix pour le développement de nœuds capteurs intelligents sans fil programmée via le logiciel de l'Arduino. Le shield Wi-Fi, connecté à la carte à proximité du capteur DHT22, constitue le nœud capteur. Dans le contexte de la maison intelligente, il est vital que les nœuds capteurs reçoivent les mises à jour du serveur à distance rapidement et de manière fiable. Le shield Wi-Fi permet à la carte Due de se connecter au réseau sans fil et de communiquer les données, supportant les protocoles TCP/IP et UDP grâce à sa pile réseau (IP). Le DHT22, un capteur numérique de température et d'humidité, est basé sur un capteur capacitif pour l'humidité et une thermistance pour la température et transmettant les données via un signal numérique sans nécessiter d'entrée analogique.

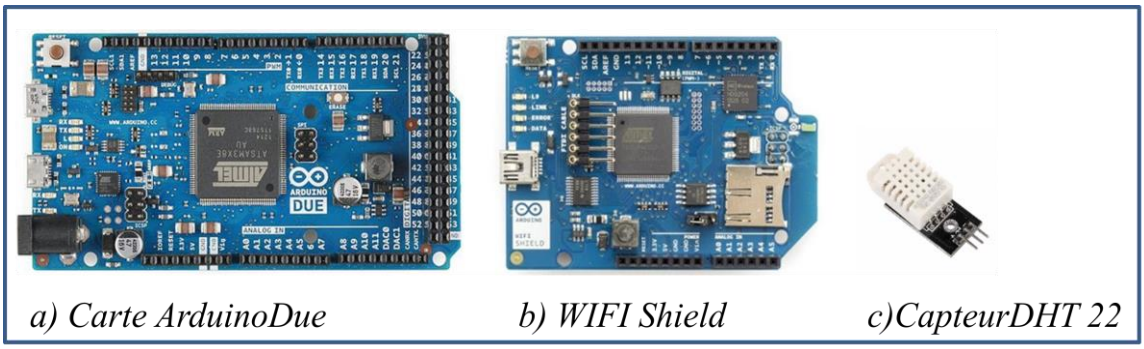


Figure III.24. Composants des nœuds capteurs.

La figure (III.25) montre une utilisation typique du capteur DHT22 connecté à un microcontrôleur.

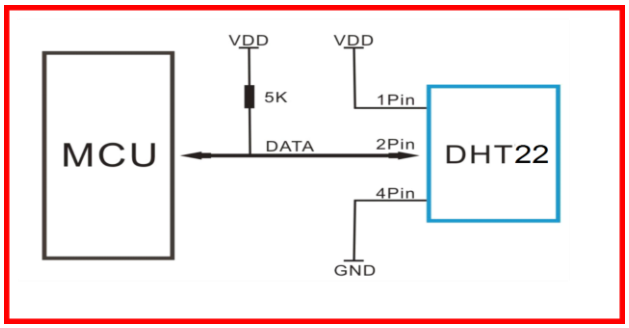


Figure III.25. Application typique du capteur DTH22.

La Figure (III.26) illustre l'architecture de notre système de réseaux de capteurs sans fil, composée de deux entités. D'un côté, nous avons les nœuds capteurs, chargés de mesurer les conditions environnementales telles que l'humidité et la température, et de l'autre, un serveur qui reçoit et traite ces données.



Figure III.26. L'architecture du système de réseaux de capteurs sans fil.

Les nœuds capteurs transmettent les informations recueillies au serveur en utilisant une connexion Wi-Fi sécurisée, appuyée par un protocole TCP/IP fiable ainsi qu'un algorithme de cryptage spécifiquement conçu pour ce système. Sur le serveur, une application développée en C# et s'appuyant sur le .NET Framework s'occupe de déchiffrer et d'authentifier les données avant de les stocker. Ces données sont ensuite gérées via SQL (Structured Query Language), un langage de manipulation de bases de données qui facilite une variété d'opérations et d'analyses. Cette infrastructure permet non seulement de stocker les données environnementales capturées mais aussi d'initier des actions automatisées en fonction de ces données, contribuant ainsi à l'efficacité et l'intelligence du système RCSF mis en place. La maquette expérimentale réalisée, composée de deux nœuds de capteur et du serveur, est illustrée sur la figure (III.27).

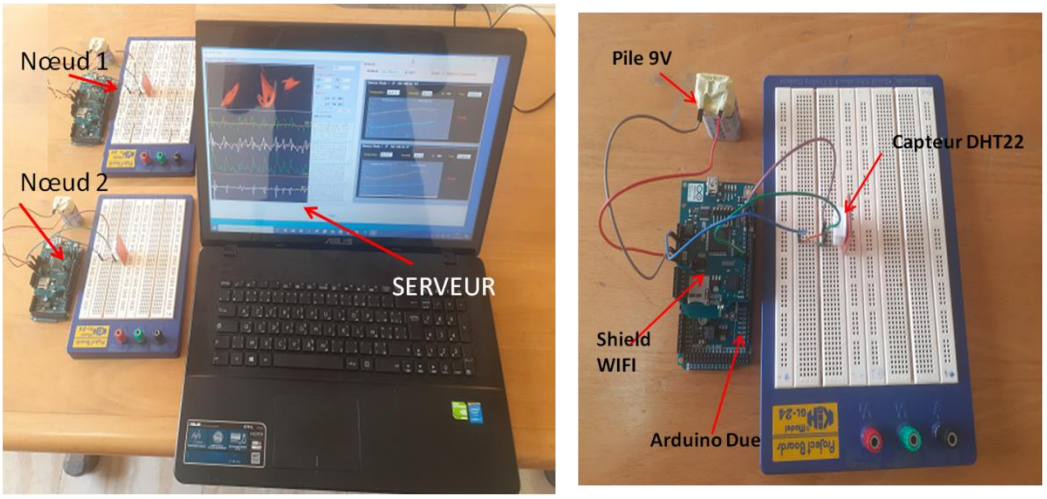


Figure III.27. Photo de la maquette expérimentale.

III.7.1 Description de l'IHM

Dans cette section, nous allons présenter l'application développée, en utilisant le langage de programmation C# (Csharp) sous l'environnement Visual Studio de Microsoft [127], sous forme d'IHM (Interface Homme Machine) pour le déchiffrement et la visualisation des données sécurisées et transmises des deux nœuds, via le Wifi, vers le serveur. La figure (III.28) présente l'IHM développée.

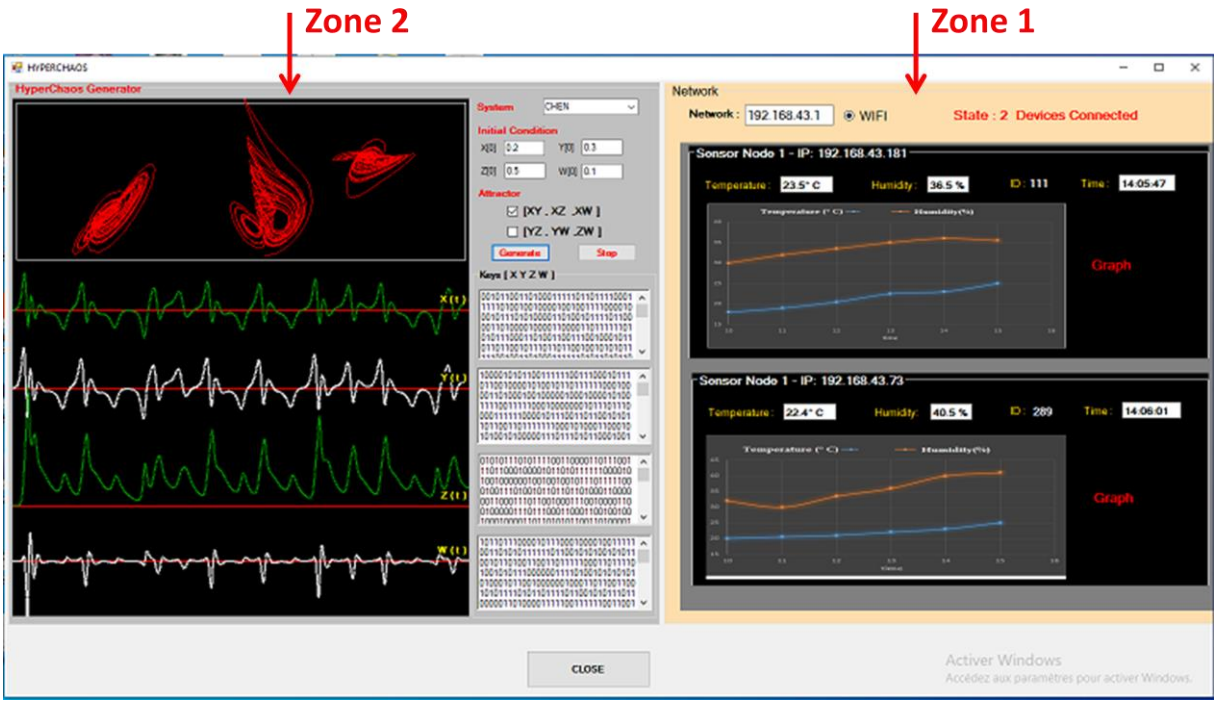


Figure III.28. IHM développée**.**

L'IMH développée est composée principalement de deux zones, une zone1 pour tout ce qui est affichage et visualisation, après décryptage et authentification, des données relatives aux deux nœuds et une zone2 dédiée pour une animation graphique. Cette IHM est interactive et facile d'utilisation. La description de chaque zone est donnée dans ce qui suit.

- **Zone1 de gestion des nœuds**

Cette zone représentée par la figure (III.29) permet de visualiser les valeurs de température, d'humidité et les identifiants (IDs) de chaque nœud après le processus de décryptage. Elle permet également l'affichage de l'évolution des mesures stockées dans une base de données.



Figure III.29. Zone de gestion des nœuds.

- **Zone2 de gestion de l'animation** :

Cette zone qui est représentée sur la figure (III.30) illustre une animation de l'évolution temporelle des signaux et des attracteurs étranges 2D du système hyperchaotiques de Chen, obtenue par la résolution de son système d'équation par la méthode RK4. Un exemple de différentes clés générées (K_1, K_2, K_3, K_4) est également représenté en binaire.



Figure III.30. Zone de gestion de l'animation

III.7.2 Algorithme de Cryptage Hyperchaotique :

Nous introduisons ici un algorithme de cryptage basé sur le principe du chiffrement par flot synchrone. L'objectif est de reproduire le principe du Chiffre de Vernam [95], en générant une suite continue de clés nouvelles, synchronisées entre l'émetteur et le destinataire [128]. Le Chiffre de Vernam, également connu sous le nom de masque jetable, est reconnu pour être théoriquement inviolable [93]. Comme démontré dans la Figure (III.31), il repose sur l'application d'une opération XOR entre le texte original et une clé aléatoire unique, de longueur équivalente à celle du message à chiffrer. Le résultat est un message chiffré dont l'analyse ne révèle aucune information utile au cryptanalyste. Dans le mode du chiffrement par flot synchrone l'émetteur et le récepteur doivent être synchronisés pour pouvoir décrypter correctement. Pour ce faire nous insérons dans chaque trame chiffrée un Vecteur d'Initialisation (VI) en plus qui va assurer la synchronisation des deux générateurs de l'émission et de la réception [129].

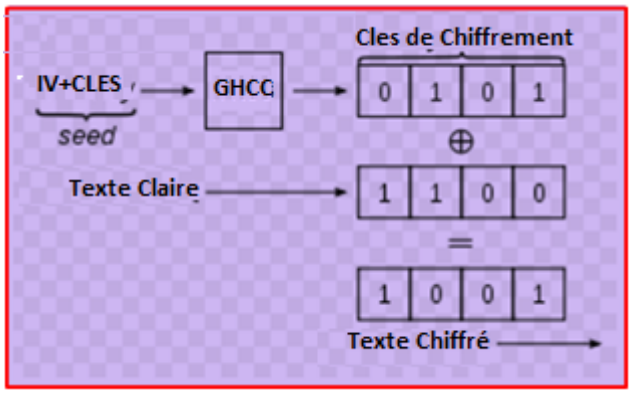


Figure III.31. Principe du chiffrement par flot.

Puisqu'un Vecteur d'Initialisation (VI) non variable et pas assez aléatoire, affaiblit considérablement la robustesse du cryptage ce qui permet aux attaques par dictionnaire et différentielles d'avoir plus de chances de déchiffrer le message crypté [130]. Pour cela nous avons utilisé un VI suffisamment aléatoire et qui est également changé à chaque nouvelle transmission. En effet, pour être sécurisé, il a été prouvé qu'un chiffrement en flux ne devrait pas utiliser la même clé pour crypter plusieurs messages [131]. Afin de masquer le VI dans la trame transmise et d'augmenter les performances de l'algorithme proposé, nous incluons un processus de mélange efficace des données du tableau à transmettre. De plus, une communication sécurisée dans la Maison Intelligente assure l'identification des capteurs légitimes dans l'environnement domestique. Un nœud capteur anonyme peut usurper la communication dans l'environnement domestique et accéder à des données privées. Le schéma de sécurité adopté doit identifier tous les nœuds anonymes, pour cela, chaque dispositif dans le réseau se voit attribuer un identifiant unique et secret ID qui peut être identifié de manière unique par le contrôleur central (serveur). Avant de donner le processus de cryptage, soulignons que les variables en majuscules X_0, Y_0, Z_0, W_0, et les paramètres a, b, c, d et e représenteront la clé secrète tandis que les variables en minuscules x_n, y_n, z_n et w_n représenteront les valeurs correspondantes à la dernière itération du générateur GHCC (dernières solutions). Ainsi, la clé secrète, le vecteur d'initialisation VI sont les entrées du générateur hyperchaotique GHCC. Nous voulons également mentionner que les clés de cryptage K_1 seront utilisées pour crypter la température, K_2 pour l'humidité, K_3 pour l'ID et enfin K_4 pour le CRC (Cyclic Redundancy Check) qui est utilisé comme MAC (Message Authentification Code). Le processus de cryptage est montré dans la Figure (III.32).

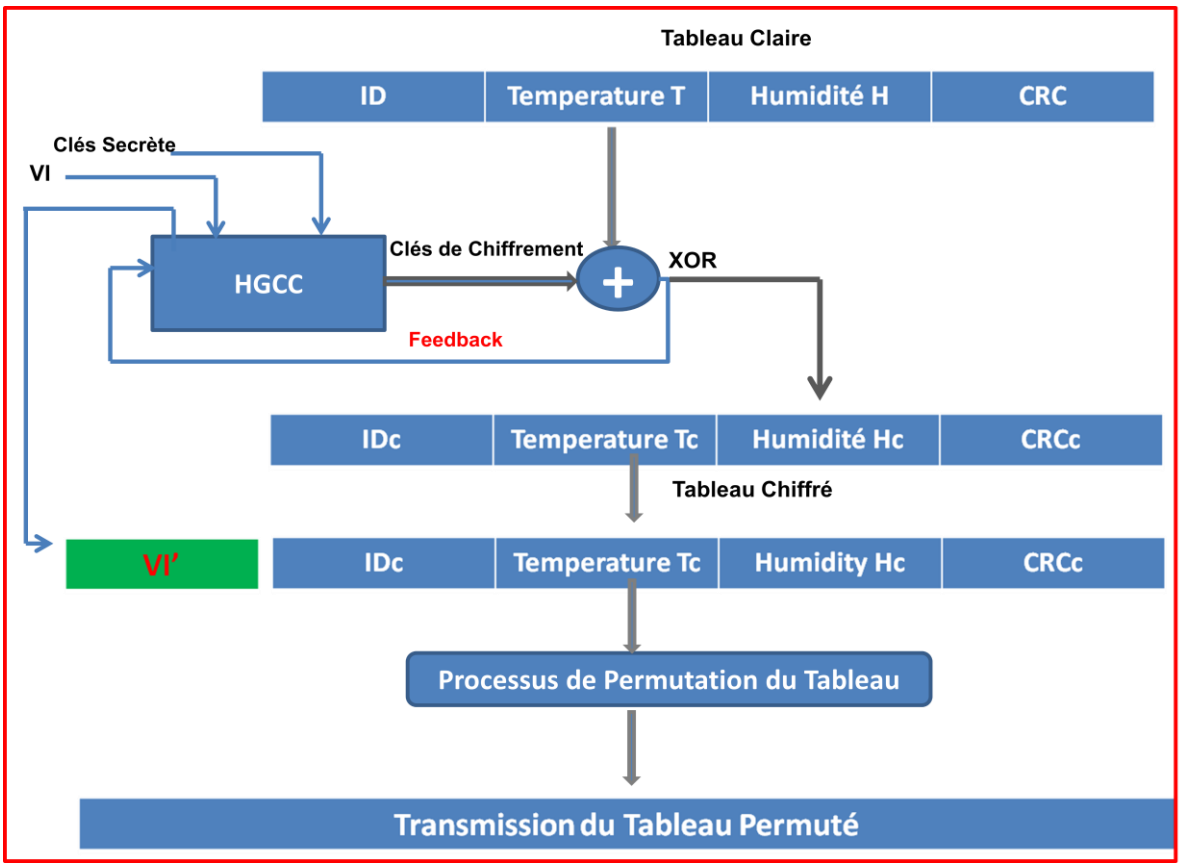


Figure III.32. Processus de cryptage.

L'Algorithme1 ci-dessous donne et illustre plus de détails sur le processus de cryptage.

Algorithm1 : Processus de cryptage

Entrée : température T et humidité H

1 : CRC = (ID ⊕ T ⊕ H) //⊕représente l'opération XOR et ID indique le code d'identification unique et secret de chaque nœud.

2 : V = ID ||T ||H|| CRC // || signifie l'opération de concaténation. Le tableau V forme le texte en claire.

3 : **Init_Vect'**=[x_n, y_n, z_n,w_n] // la dernière itération du générateur GHCC.

4 : **Init_Vect** =**Init_Vect'**⊕[X_0⊕a, Y_0⊕b, Z_0⊕c, W_0⊕d]// X_0, Y_0,Z_0,W_0, a, b, c et d sont la clé secrète.

5 : **C=V ⊕ K** // avec utilisation du **Init_Vect** comme valeurs initiales du générateur GHCC. K représente l'ensemble des clés de chiffrement.

6 : A=**Init_Vect' || C** // A constitue le tableau final chiffré de taille ***L*.**

7 : A ← Permutation du tableau avec PERMUT(***L*)**

Sortie : Envoi du tableau A en utilisant le protocole TCP/IP vers le serveur.

Quant au processus de décryptage, coté serveur, qui est différent du procédé du cryptage il est illustré par la figure (III.33) sous forme d'organigramme et détaillé par l'Algorithme2.

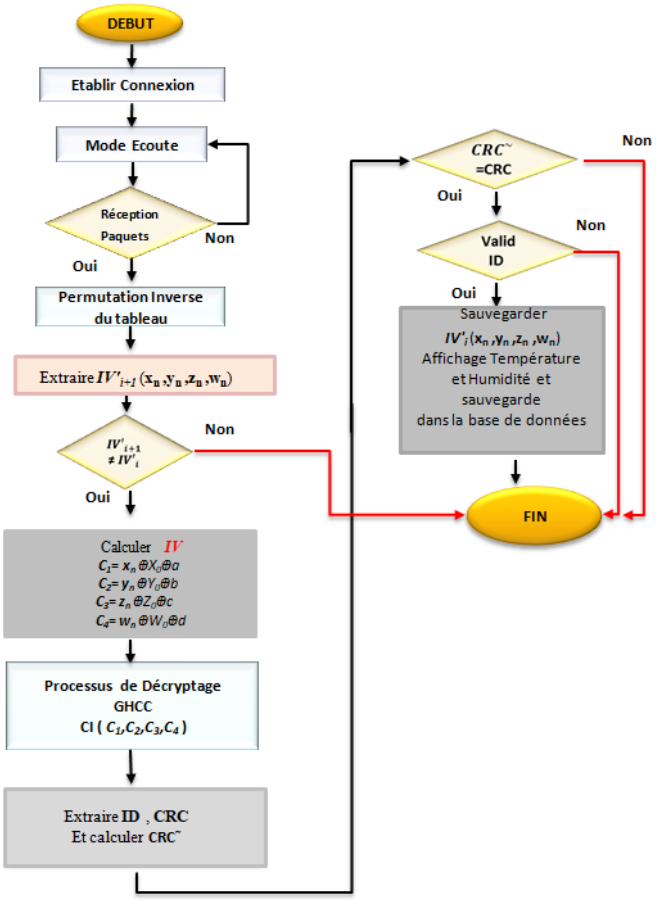


Figure III.33. Organigramme de décryptage.

Algorithme2 : Processus de décryptage

Entrée : Tableau chiffré A

1: A←Permutation inverse PERMUT$(L)^{-1}$.

2 : Extraire Init_Vect'$_{i+1}$= [x_n, y_n, z_n,w_n]. // en comparant Init_Vect'$_{i+1}$ et le dernier Init_Vect'$_i$le serveur vérifie la fraicheur des données.

3 : Si (Init_Vect'$_{i+1}$==Init_Vect'$_i$) // le serveur rejette le message et retourne à1'étape 1.

Si non Init_Vect = Init_Vect'$_{i+1}$⊕[X_0⊕a, Y_0⊕b, Z_0⊕c, W_0⊕d] // X_0, Y_0,Z_0,W_0, a, b, c et d sont la clé secrète.

4 : M=A⊕K. // M tableau déchiffré avec l'ensemble des clés de déchiffrement K.

5 : Calcul du CRC' du tableau M.

6 : Extraire ID et CRC à partir du M.

7 : Si (CRC'≠ CRC) or ID non enregistré dans la base de donnée le serveur rejette le message.

Si non le server sauvegarde **Init_Vect'**$_{i+1}$ et supprime le dernier Init_Vect'$_i$ enregistré.

Sortie: Affichage et enregistrement des mesures de température T et d'humidité H.

Il convient également de noter qu'à chaque mise sous tension, tous les nœuds capteurs exécutent une procédure d'initialisation du générateur HGCC comme montré dans l'Algorithme3. Cela garantit qu'un début de processus de cryptage ne sera pas le même pour chaque nœud, puisque les valeurs du VI'= [x_n, y_n, $z_{n,}w_n$] (conditions initiales) seront toujours différentes pour les générateurs HGCC de chaque nœud ,renforçant ainsi la sécurité du système. Pour cela nous exploitons la fonction randomSeed() d'arduino qui permet d'initialiser le générateur de nombres aléatoires, donné par l'instruction random(), avec une entrée aléatoire comme la valeur renvoyée par la fonction analogRead() sur une broche analogique en l'air.

Algorithm3:Initialisation du générateur GHCC

Entrée : $X_{0,}$ Y_0, Z_0, W_0, *a*, *b*, *c* et *d*. // Clés secrète.

1 : randomSeed(analogRead(A0)). //A0 entrée analogique non connectée

2 : *N*=random(1,1000). // fonction arduino pour générer des nombres pseudo-aléatoires

3 : Itérer le générateur GHCC *N* fois

Sortie : VI'= [x_n, y_n, $z_{n,}w_n$]. // VI' conditions initiales.

III.7.3 Analyse et Performance du Schéma Cryptographique proposé

Dans cette section une analyse de performance de la méthode proposée est effectuée afin de vérifier que toutes les principales exigences de sécurité des données à savoir la confidentialité, l'authentification, la non-répudiation, l'intégrité, la résilience et la fraîcheur des données sont respectées [131].

- ✓ **Confidentialité** : Le schéma de cryptage que nous proposons assure la confidentialité des communications entre les nœuds et le serveur par l'emploi d'un chiffrement de flux synchrone utilisant une grande clé $H=2^{\wedge 640}$ bits. La probabilité qu'un intercepteur réussisse à percer la clé par l'attaque par force brute est donc de 1 sur $2^{\wedge 640}$. Les clés de chiffrement générées par le HGCC, caractérisées par un grand aspect aléatoire, sont prévues pour un usage unique, rendant inutile toute clé potentiellement découverte par un attaquant pour le décryptage des autres messages. Par conséquent, l'obtention de la clé secrète d'une transmission par l'attaquant n'affecte pas les autres clés, assurant ainsi une confidentialité élevée des données.
- ✓ **Authentification et Non-Répudiation** : L'authentification, essentielle dans l'environnement d'une maison intelligente, permet de reconnaître les nœuds légitimes et assure au serveur que les données proviennent bien de la source attendue. Notre

schéma attribue à chaque nœud un ID unique et secret dès l'installation, enregistré par le serveur pour tous les nœuds lors de leur configuration. Un attaquant introduisant un nœud malveillant pourrait tenter de perturber la communication, mais serait rapidement identifié par l'absence de correspondance de l'ID avec ceux des nœuds autorisés, sécurisant ainsi l'échange d'informations.

- ✓ **Intégrité** : L'intégrité, clé de voûte de la sécurité, est préservée en calculant et en joignant un Code d'Authentification de Message (MAC), ici obtenu par le CRC, à chaque message transmis. Le serveur, à la réception, vérifie le CRC avec celui calculé sur le message, permettant d'identifier toute modification malveillante des données.
- ✓ **Fraîcheur des Données** : La fraîcheur garantit l'actualité des données et empêche leur réutilisation malveillante. Le serveur évalue la fraîcheur de chaque message reçu en comparant l'IV' actuel avec le dernier enregistré, refusant les messages dont l'IV' n'a pas varié, assurant ainsi que les données sont récentes et valides.
- ✓ **Résilience** : La résilience est particulièrement importante dans les RCSF, où la compromission d'un seul nœud ne doit pas affecter la sécurité globale. Notre approche diffère des protocoles comme Z-wave et ZigBee, où la compromission d'un nœud met en péril l'ensemble du réseau en révélant les clés. Dans notre cas, seule la clé du nœud compromis est affectée.

- ✓ **Temps d'exécution :** Le tableau (III.8) présente une comparaison entre notre algorithme et d'autres algorithmes de cryptage en terme de temps d'exécution en millisecondes. A partir des informations indiquées dans le tableau, nous déduisons que la vitesse de cryptage du cryptosystème proposé est meilleure que les autres.

Algorithme	Taille des Paquets (Bytes)			Platforme
	16	32	512	
GHCC(proposé)	0.8ms	1.5ms	25.6ms	Arduino DUE
Skipjack [132]	-	6.62ms	103.45ms	Sun Small Pro-grammable object Technology.(SunSPOT)
RC5 [132]	-	14.96ms	235.39ms	
MBCC [132]	-	34.02ms	533.29ms	
BCC [132]	-	220.74ms	3515.43ms	
AES-128 [133]	5.856ms	-	-	Arduino UNO
DES [133]	39.040ms	-	-	

Tableau III.8. Temps d'exécution en millisecondes.

- ✓ **Consommation d'énergie :** La consommation d'énergie en joule est calculée en utilisant la relation suivante :

$$E = P \times T.$$

Avec T : Temps d'exécution en secondes.

Et P : Puissance consommée en watts calculée selon la formule ci-dessous :

$$P = I \times V.$$

Où I est le courant consommé en Ampère et V la tension d'alimentation en volt.

Le principe de mesure du courant est illustré par la figure (III.34). La mesure est réalisée en utilisant le capteur de courant ACS712 qui est basé sur la technique de conversion de courant en un signal de tension proportionnel à l'intensité du courant.

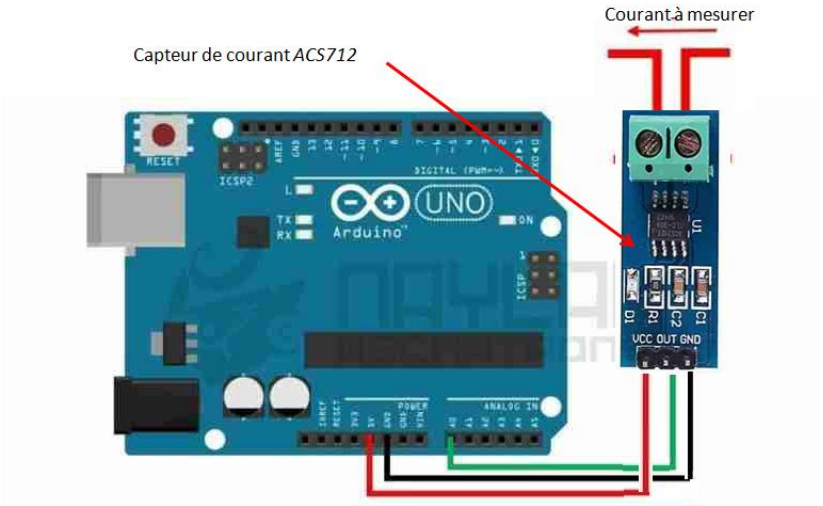


Figure III.34. Principe de mesure du courant consommé**.**

Le tableau (III.9) donne les résultats de consommation d'énergie de notre algorithme ainsi qu'une comparaison avec d'autres algorithmes de cryptage. Les résultats représentés sur le tableau (III.9) montre bien que le cryptosystème proposé consomme moins que les autres.

Algorithme	Taille des Paquets (Bytes)	
	32	512
GHCC(proposé)	2.78 mJ	39.704 mJ
Skipjack [132]	7.45 mJ	120.035 mJ
RC5 [132]	17.904 mJ	284.152 mJ
MBCC [132]	40.646 mJ	645.925 mJ
BCC [132]	267.943 mJ	4387.831 mJ

Tableau III.9. Consommation d'énergie en milli Joule**.**

II.8. Conclusion

Dans ce chapitre, nous nous sommes intéressés à la génération des clés de chiffrement à partir du générateur hyperchaotique de Chen. Le défi était d'assurer un bon choix du générateur et d'assurer de bonnes performances statistiques des clés générées. Le choix du générateur hyperchaotique de Chen est dû au fait qu'il génère un comportement très riche en dynamique chaotique et offre un espace de clés très large, en plus le fait de prendre uniquement les parties fractionnaires des signaux générés, tout en les couplant avec des fonctions booléennes non linéaires nous permet de générer des clés de chiffrement de très bonnes propriétés statistiques. Ce constat est confirmé par le passage avec succès de ces clés de tous les tests statistiques de la batterie de tests du NIST. De plus, pour plus d'assurance, nous avons effectué une analyse de sécurité des données chiffrées en utilisant le masque jetable à base de l'opérateur XOR. Nous avons pris des images de référence comme données à chiffrer. Les résultats obtenus pour le calcul d'entropie, des histogrammes, de corrélation, de la sensibilité à la clé et l'espace des clés valident l'efficacité des clés de chiffrement hyperchaotiques proposées et le procédé du chiffrement à masque jetable spécifique au chiffrement basé-chaos.

Par la suite nous avons décrit et détaillé la plateforme matérielle de RCSF développée composée essentiellement d'une carte Arduino Due, d'un shield WI-Fi et d'un capteur de température et d'humidité DHT22. Puis on a donné et détaillé l'algorithme que nous avons développé, considéré simple et robuste, qui est basé sur le mécanisme de génération de clés de chiffrements efficace du générateur GHCC afin de crypter les données avant leur envoi. A la fin on a montré que la solution proposée répond à toutes les principales exigences de sécurité des données (confidentialité, authentification et non-répudiation, intégrité, résilience et fraîcheur des données) ainsi que le temps de traitement et de la consommation d'énergie.

Chapitre IV

SECURISATION D'UN RESEAU DE CAPTEURS SANS FIL HETEROGENE BASE CHAOS

IV.1. Introduction

Dans ce quatrième chapitre nous présenterons notre deuxième contribution qui consiste à développer et à réaliser un cryptosystème basé chaos pour sécuriser différentes données (température, humidité et flux audio/vidéo) au sein d'un réseau de capteur sans fil hétérogène utilisant plusieurs technologies de communication (Wifi, Bluetooth et Xbee). Pour réaliser ces objectifs, dans un premier temps, nous avons effectué une étude sur les systèmes chaotiques dans le but de déterminer celui répondant au mieux aux exigences d'un chiffrement robuste et léger. Notre choix s'est porté sur le Système Chaotique Unifié comme générateur de clés de chiffrement. Après une présentation et une simulation sous Matlab du générateur Chaotique Unifié nous montrons les signaux temporels et les différents attracteurs caractérisant ce dernier puis une analyse statistique des clés de chiffrement est effectuée en utilisant la batterie de tests de NIST.

Dans un deuxième temps, pour plus d'assurance et de validation, nous avons effectué une analyse de sécurité des données chiffrées en utilisant un masque jetable à base de l'opérateur XOR. Nous avons pris l'image et l'audio comme données à chiffrer. Sachant que le succès et la robustesse d'une méthode de cryptage se résume à sa résistance face aux diverses attaques telles que l'attaque par force brute, l'attaque statistique et l'attaque différentielle. Nous avons effectué quelques tests de sécurité les plus connus et utilisés dans la littérature tel que l'espace des clés, le calcul d'entropie, l'autocorrélation, les histogrammes et la sensibilité des clés. Ces tests ont étés appliqués sur différentes images de référence. Par la suite, nous allons proposer une solution pour l'épineux problème de synchronisation, puis nous allons présenter et décrire le réseau de capteur réalisé ainsi que les différentes Interfaces Homme Machine(IHM) développées. A la fin une analyse de performance des algorithmes proposés est effectuée. Enfin une conclusion clôture ce chapitre.

IV.2. Le système chaotique unifié

En 2002, Lu et Chen ont proposé un nouveau système chaotique qui relie l'attracteur de Lorenz, de Chen et celui de Lu. C'est ce qu'on appelle le système chaotique unifie [134]. Selon la définition de Vanecek et Celikovsky [135]. Le système de Lorenz, de Chen et celui de Lu appartiennent à différents types de topologie. Le modèle mathématique du système chaotique unifié est donné par l'équation suivante :

$$\begin{cases} \dot{x} = (25k + a)(y - x) \\ \dot{y} = (b - 35k)x - xz + (29a - 1)y \\ \dot{z} = xy - c(1 + k)z \end{cases} \quad \text{(IV. 1)}$$

Avec k une valeur entre 0 et 1, a= 10, b=28 et c = 13. Dans cette plage, le système chaotique ` unifié présente des caractéristiques chaotiques. Quand k=0.8, le système chaotique unifié appartient au système de Lu généralisé, quand k varie entre 0 et 0.8, le système chaotique unifié appartient au système de Lorenz généralisé et quand il varie de 0.8 à 1, le système chaotique unifié appartient au système de Chen généralisé. La figure (IV.1) illustre bien le comportement du système chaotique unifié en fonction de la valeur de k.

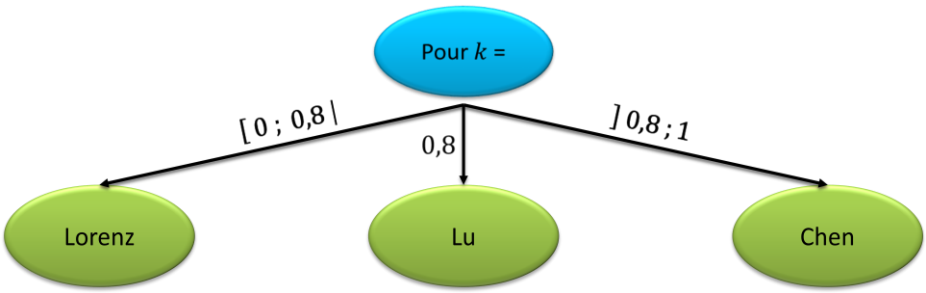


Figure IV.1. Système chaotique unifié en fonction de k.

IV.3. Système Chaotique Unifié Discrétisé(SCUD)

Dans les systèmes embarqués, nous sommes limités par des contraintes telles que la consommation des ressources et le temps d'exécution. Pour cela nous avons choisis de discrétiser le système chaotique unifié à l'aide de la méthode d'Euler [136]**.** Nous obtenons ainsi le système d'équation suivant :

$$\begin{cases} x_{n+1} = x_n + h(25k + a)(y_n - x_n) \\ y_{n+1} = y_n + h[(b - 35k)x_n - x_n z_n + (29a - 1)y_n] \\ z_{n+1} = z_n + h[x_n y_n - c(1 + k)z_n] \end{cases} \quad \text{(IV. 2)}$$

Avec h le pas d'échantillonnage qui est égale à 0.001.

IV.4. Les signaux temporels et les attracteurs.

Les résultats de simulation temporelle sous Matlab de SCUD, avec un pas h = 0.001, a =10, b = 28 et c = 13, x_0 = 0.01, y_0 = 5 et z_0 = 25, sont présentés dans les figures ci-dessous pour différentes valeurs de k.

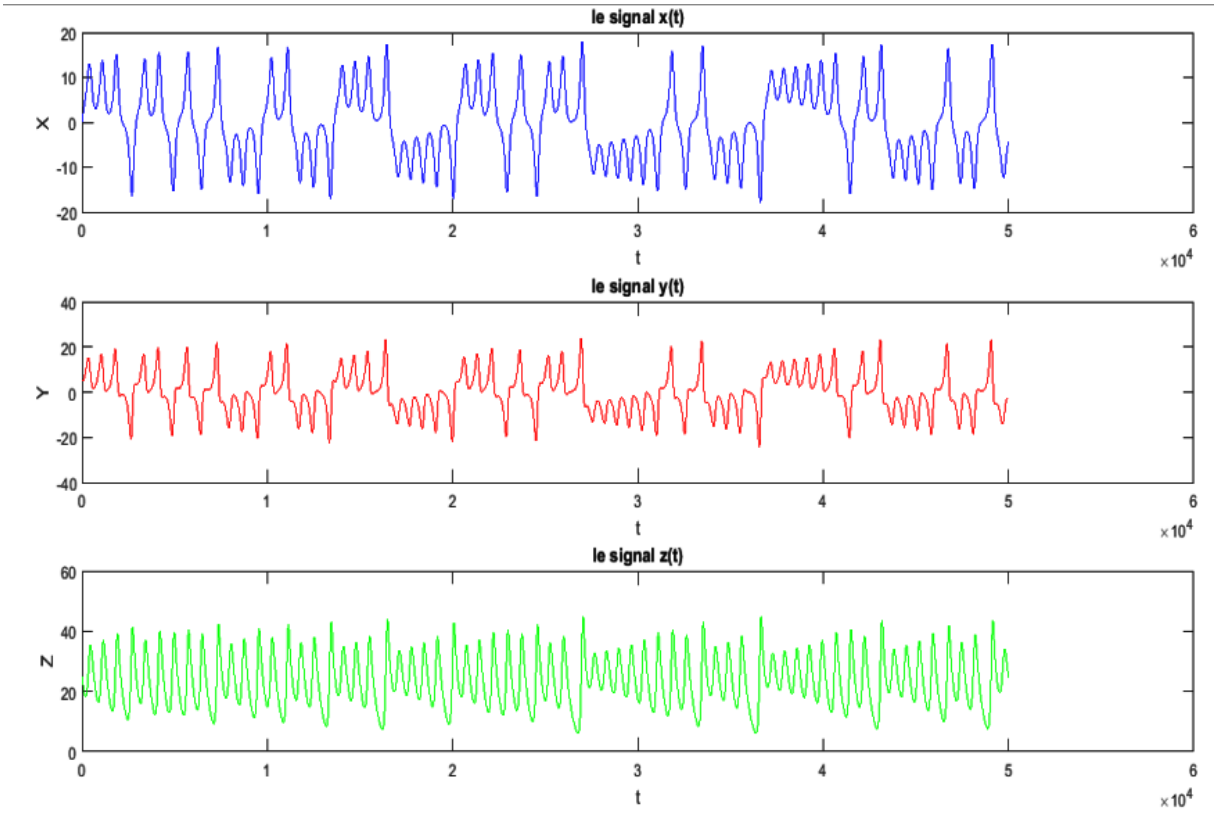


Figure IV.2. Les signaux temporels pour k = 0.

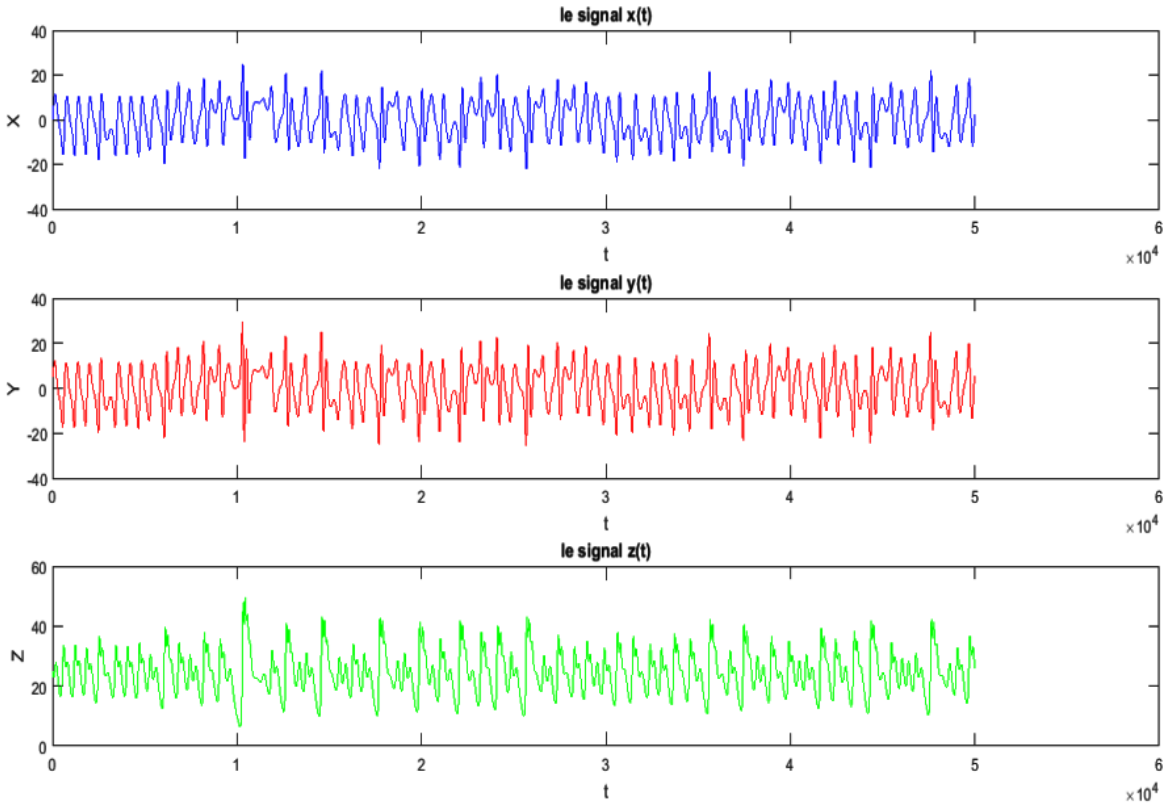


Figure IV.3. Les signaux temporels pour k = 0.8.

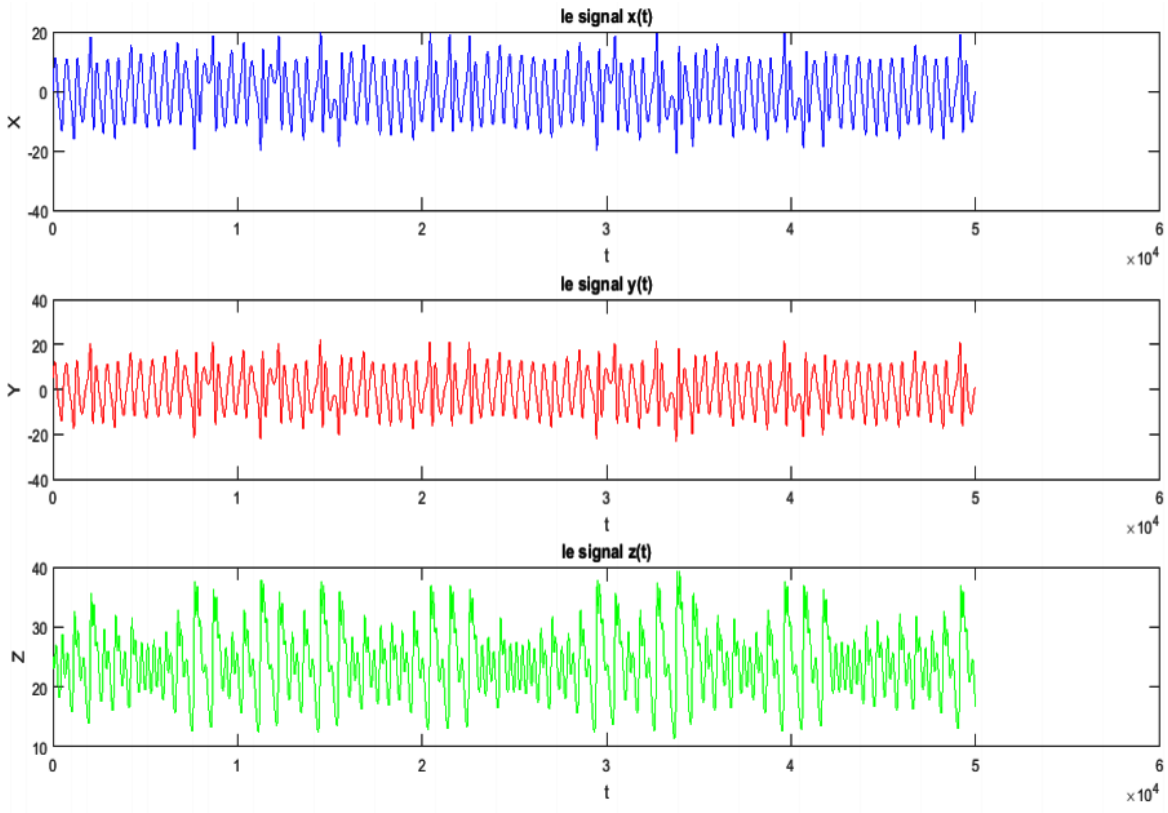


Figure IV.4. Les signaux temporels pour k = 1.

En prenant les mêmes paramètres et conditions initiales, nous représenterons les attracteurs 2D dans les espaces (x,y), (x,z) et (y,z) .

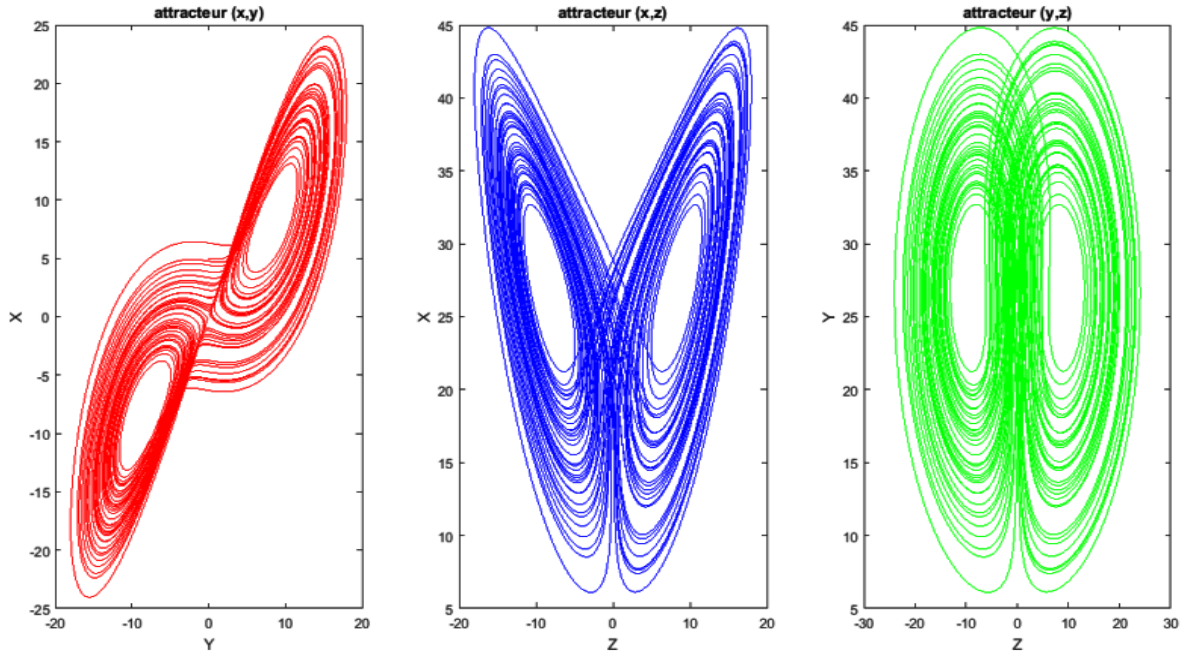


Figure IV.5. Les attracteurs 2D pour k=0.

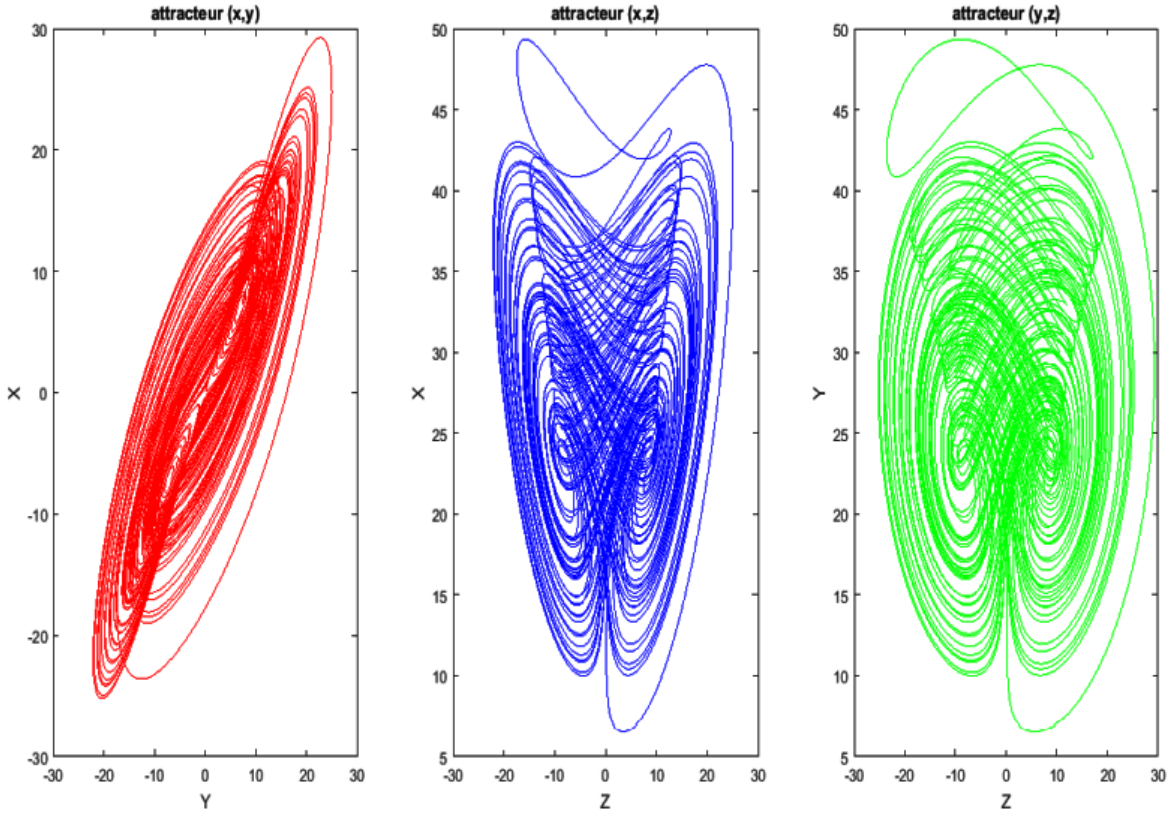


Figure IV.6. Les attracteurs 2D pour k=0.8.

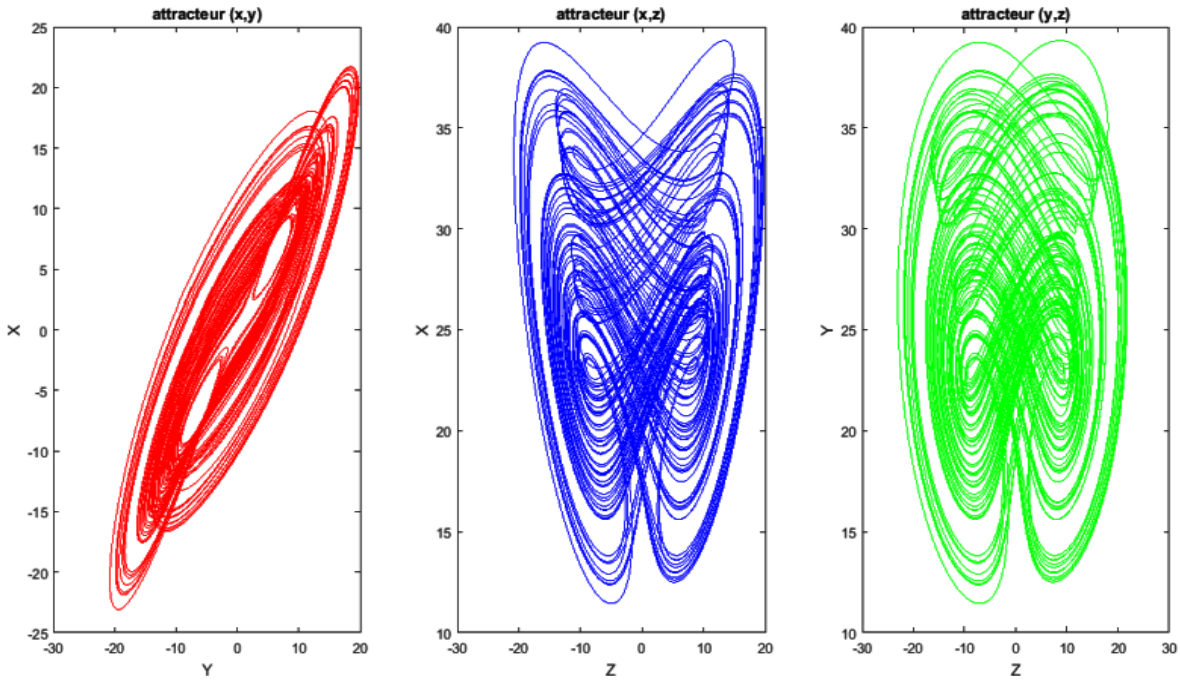


Figure IV.7. Les attracteurs 2D pour k=1.

Les figures suivantes donnent les attracteurs 3D dans l'espace (x,y,z) pour différentes valeurs de k.

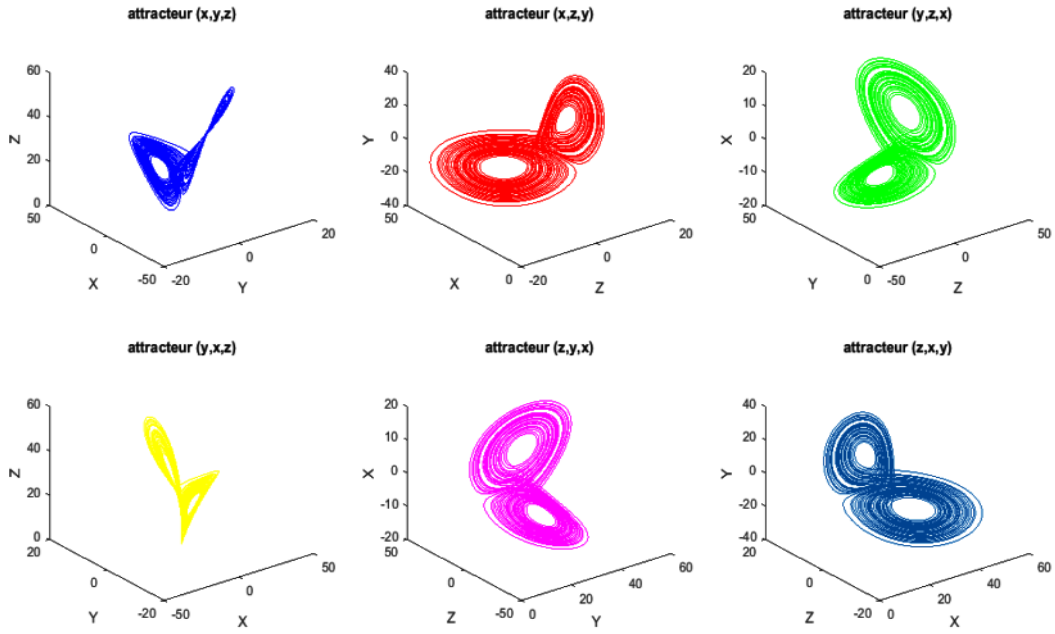


Figure IV.8. Les attracteurs 3D pour k=0.

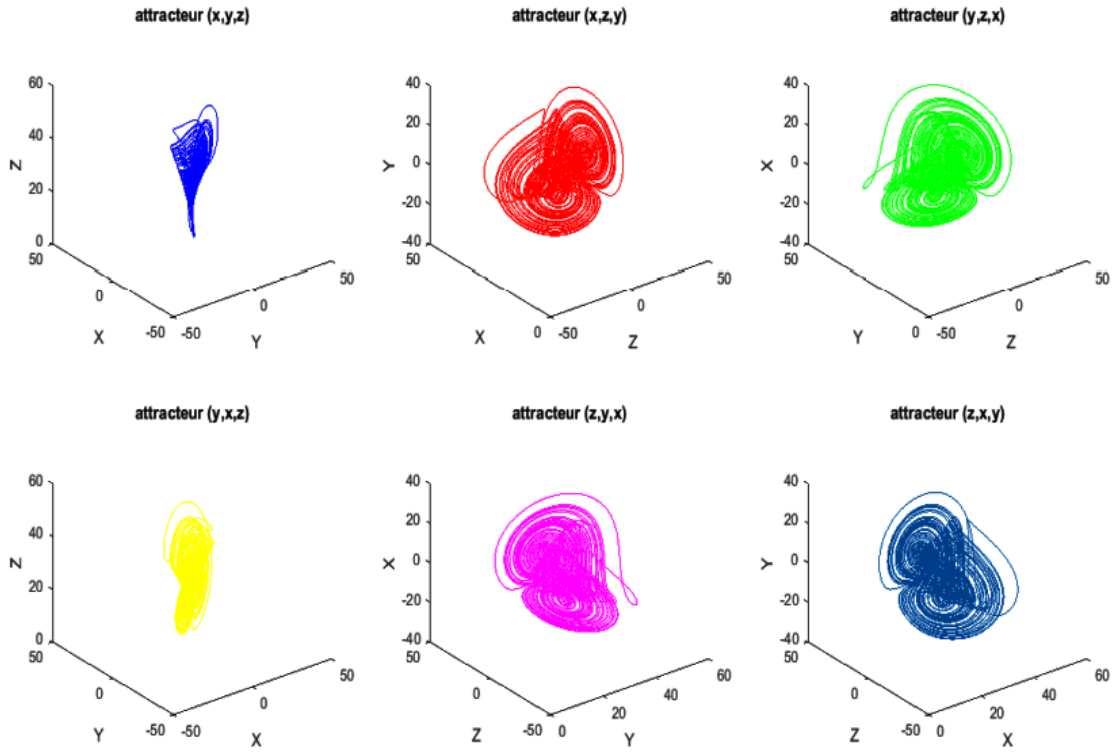


Figure IV.9. Les attracteurs 3D pour k=0.8.

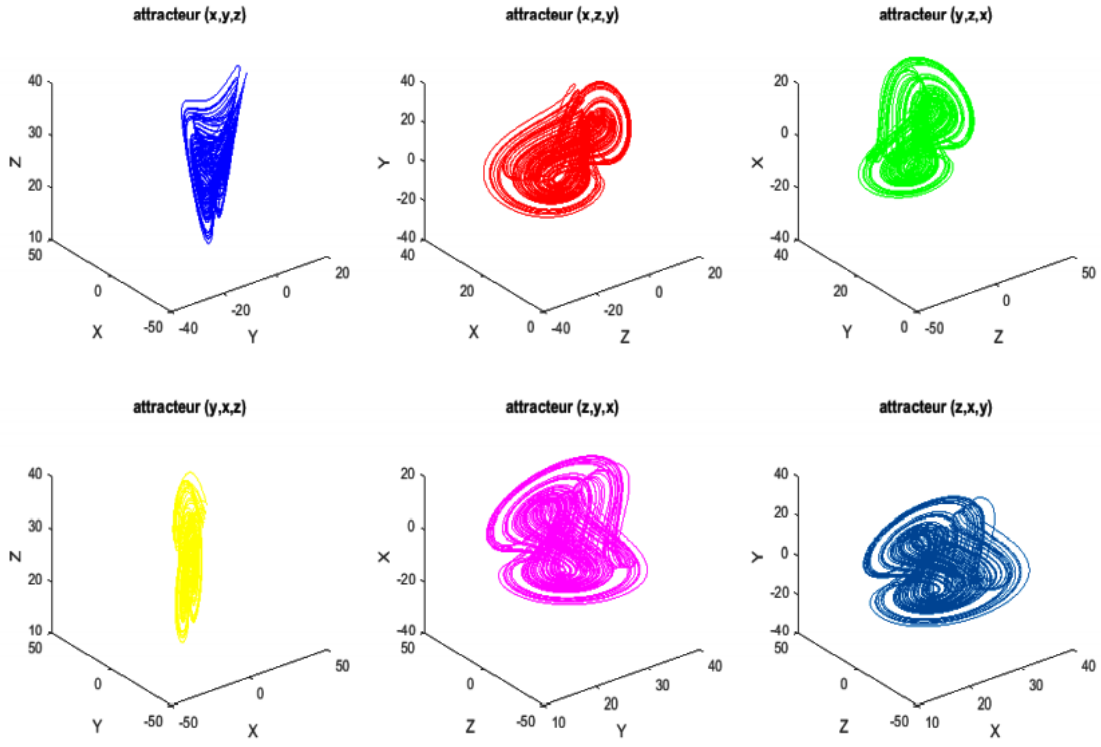


Figure IV.10. Les attracteurs 3D pour k=1.

IV.5. Génération des clés de chiffrement chaotiques

Plusieurs travaux précédents ont démontré que les séquences binaires générées directement à partir des générateurs chaotiques ne sont pas robustes et ne passent pas les tests de NIST, donc ne peuvent pas être considérer comme des clés de chiffrement et un des problèmes qui cause cette dégradation en robustesse est la présence des longues séquences de zéro et de un comme illustré sur la figure (IV.11**)** qui présente un fichier ,sous forme binaire, contenant des clés issues directement de notre générateur.

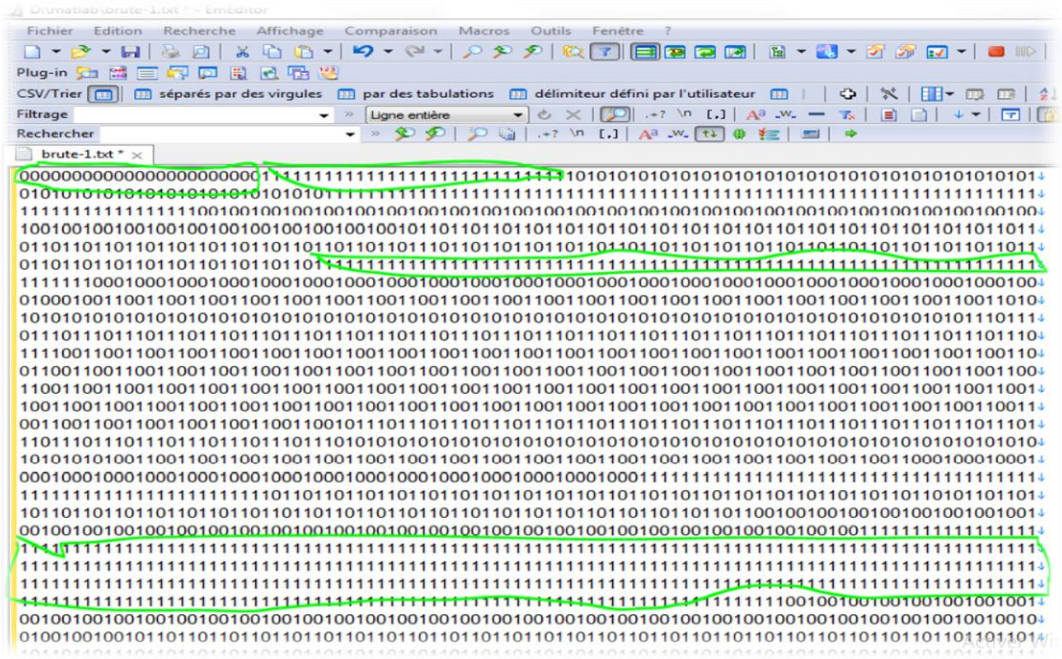


Figure IV.11. Fichier contenant la représentation binaire de la clé X.

Pour cela, plusieurs solutions ont été proposées dans ce sens. On choisit parmi elles la méthode de la partie fractionnaire qui consiste à :

- Augmenter la précision en augmentant la taille des parties fractionnaires : ceci permet de résoudre le problème lié à la dégradation de la dynamique chaotique ;
- Extraction des parties fractionnaires et les prendre comme clés de chiffrement : ceci fait augmenter la complexité des clés et aide à éliminer toute ressemblance entre elles avec la possibilité de combiner les trois clés pour former de très longues clés ;
- Pour augmenter l'espace des clés et pour mieux perturber le générateur, on propose que le paramètre k défini précédemment dans l'équation (IV.1) soit modifié à chaque itération du générateur, et sa valeur sera contrôlée par la carte chaotique discrète logistique Map qui est exprimé par l'équation suivante :

$$v_{n+1} = r * v_n * (1 - v_n) \qquad (IV.3)$$

Avec r = 3.6 et $v \in [0,1]$.

Si la valeur de v(n) varie entre 0 et 0.8 k prendra la valeur 0, si elle varie entre 0.8 et 1 k prendra la valeur 1 et enfin si v(n) est égale à 0.8, k prendra la valeur 0.8.

IV.5.1. Principe du générateur chaotique

En tenant compte de ces différentes recommandations, nous proposons d'appliquer cette technique au système chaotique unifié discrétisé. Ainsi le générateur de clés de chiffrement peut être modélisé par le système d'équations suivant :

$$\begin{cases} X = PF(x) \\ Y = PF(y) \\ Z = PF(z) \end{cases} \qquad (IV.4)$$

Avec : X = PF(x) = abs(x) –floor(abs(x)).

Où abs(x) définit la valeur absolue de x, floor(x) représente la partie entière de x.

Les *x*, *y* et *z* représentent les signaux du système chaotique unifié discrétisé tandis que les X, Y et Z sont les clés de chiffrement codées sur 32 bits. Nous pouvons ainsi illustré la génération des clés de chiffrement X, Y et Z par le schéma bloc de la figure (IV.12). Celle-ci présente ce que nous allons appeler Générateur basé-Système Chaotique Unifié Discrétisé (GSCUD).

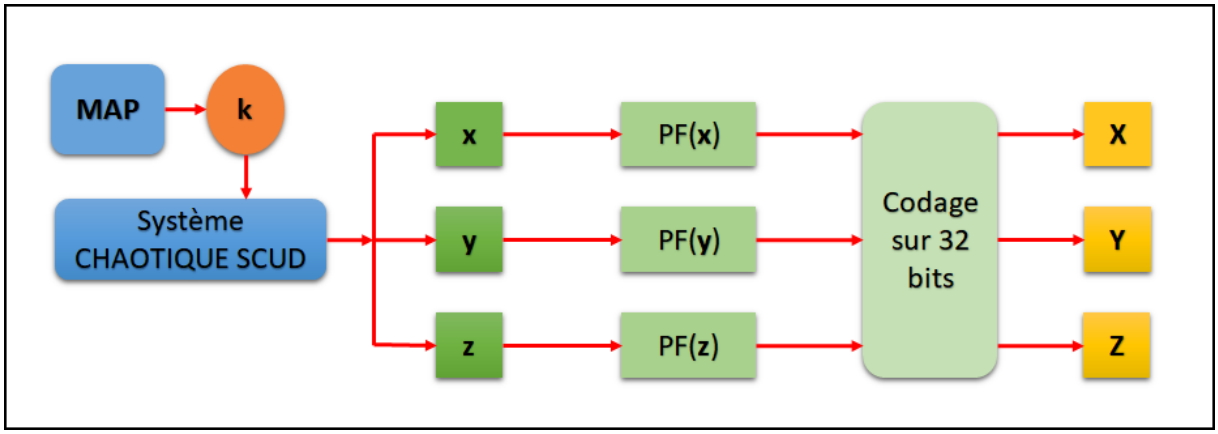


Figure IV.12. Schéma Bloc du GSCUD.

IV.5.2. Signaux temporels et attracteurs du GSCUD

Les résultats de simulation temporelle, sous Matlab, des signaux chaotiques *X*, *Y* et Z pour les mêmes valeurs des paramètres et conditions initiales tout en prenant v(0)=0.4 et r=3.6 pour la carte MAP sont présentés dans les figures (IV.13) et (IV.14). Ainsi, la figure (IV.13) illustre l'évolution temporelle des signaux chaotiques *X*, *Y* et *Z* respectivement alors que la figure (IV.14) présente les attracteurs chaotiques 2D dans les espaces (*X*-*Y*), (*X*-*Z*) et (*Y*-*Z*) ainsi que l'attracteur 3D dans l'espace(*X*-*Y*-*Z*).

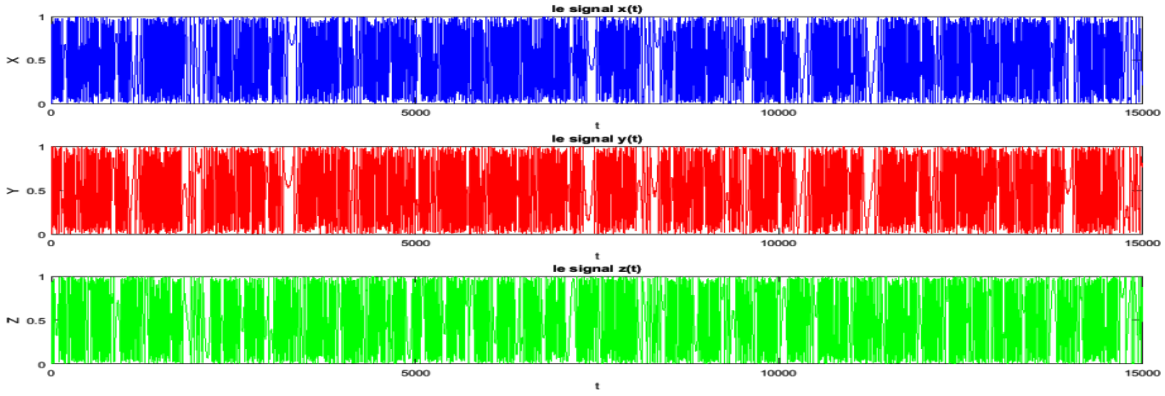


Figure IV.13. Signaux temporels des signaux *X*(*t*), *Y*(*t*) et Z_1(*t*) du GSCUD.

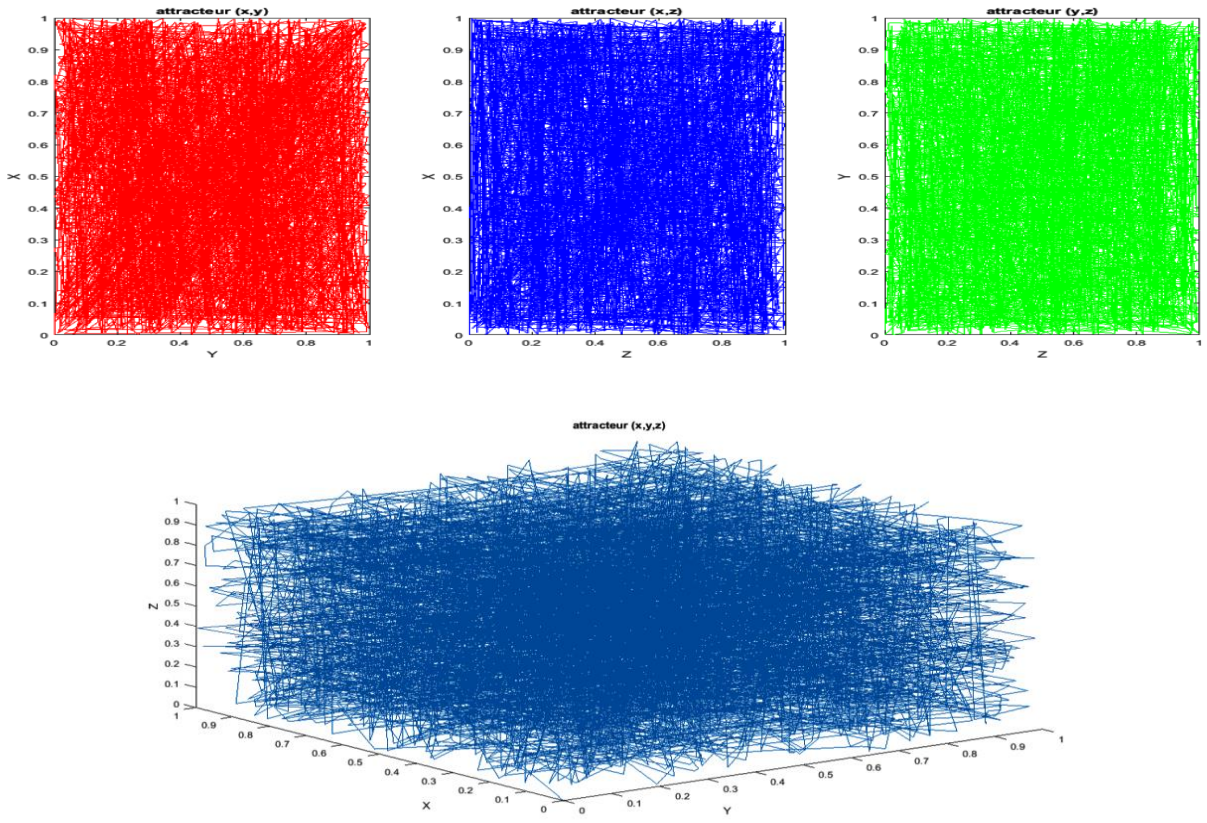


Figure IV.14. Attracteurs 2D et 3D du GSCUD.

Les résultats de la figure (IV.13) montrent bien que les signaux chaotiques *X*, *Y* et *Z* se caractérisent par une grande variabilité temporelle, semblable au bruit. De même, les trajectoires de leurs attracteurs 2D et 3D occupent complètement les espaces de phases.

IV.5.3. Tests statistiques

Pour évaluer le critère aléatoire des clés générées, nous avons effectué une analyse statistique en utilisant la batterie de tests statistiques du NIST. Notre analyse a été faite sur 300 séquences de longueur d'un million de bits (1000000 bits). Les résultats des tests sont présentés dans le tableau (IV.1).

Tests		P_value_T		
N°	Type de test	*X*	*Y*	Z
1	FrequencyTest	0.8953	0.8987	0.8091
2	Block FrequencyTest	0.9444	0.9953	0.9092
3	Cumulative Sums Up	0.9206	0.8331	0.9797
4	Cumulative Sums Down	0.8621	0.7078	0.8695
5	RunsTest	0.8772	0.8369	0.8994
6	Long Runs of One's Test	0.8696	0.9190	0.9146
7	Rank Test	0.9349	0.8821	0.8518
8	Discrete Fourier Transform	0.9234	0.9454	0.9565
9	Non-overlapping Template Matching	0.9142	0.8860	0.8834
10	Overlapping Template Matching	0.9873	0.9409	0.8548
11	Maurer's"Universal Statistical"	0.9674	0.8521	0.9876
12	Approximate Entropy	0.5517	0.8518	0.8718
13	Random Excursions Test	0.5591	0.8695	0.8957
14	Random Excursions Variant Test	0.8634	0.4099	0.9669
15	Serial Test1	0.8735	0.9810	0.9620
16	Serial Test2	0.9099	0.9264	0.8259
17	Linear complexityTest	0.9957	0.9977	0.9913

Tableau IV.1. Résultats des tests statistiques du NIST.

Nous remarquons bien que tous l'ensemble des clés *X*, *Y* et *Z* générées par le GSCUD ont passés les 17 tests de NIST avec succès. Ce qui montre clairement que les signaux de sorties du générateur GSCUD ont un bon caractère aléatoire.

IV.6. Structure du cryptosystème proposé

Pour le schéma global du cryptosystéme (chiffrement/déchiffrement), nous gardons la même démarche que celle adoptée pour le cryptosysteme du chapitre (III) qui consiste en fait

à rebouclé (feedback) le message chiffré vers l'entrée pour perturber le générateur et ainsi complexifier au maximum sa dynamique chaotique (voir Figure III.13).

IV.7. Analyse de sécurité d'images chiffrées

IV.7.1. Analyse différentielle

Le tableau (IV.2) donne les résultats de l'analyse différentielle du cryptosysteme proposé.

Image	**Taille de l'image**	**NPRC (%)**	**UACI (%)**	**DH (%)**
Lena	512x512x3	99.567	33.298	0.498
Barbara	512x512x3	99.6069	33.570	0.50
Pepper	512x512x3	99.604	33.379	0.499

Tableau IV.2. Résultats d'analyse du NPCR, l'UACI et DH.

Les résultats montrent bien la robustesse du cryptosystème proposé contre l'attaque différentielle. En effet, on constate bien que les valeurs du NPCR et de l'UACI sont très proches, respectivement, des valeurs optimales de 99,61% et de 33,46% et que les valeurs des DH sont presque égales à 50%. A titre comparatif, nous donnons dans le tableau (IV.3) les valeurs de NPCR, UACI et de DH, de quelques cryptosystèmes basés chaos connus dans la littérature. On voit bien que les résultats obtenus sont assez comparables.

Algorithme	Image	UACI (%)	NPCR (%)	DH
GSCUD	Lena	33.298	99.567	0.498
GHCC	Lena	33.405	99.597	0.499483
Crypto-C [106]	Lena	33.462	99.609	0.499912
Ref.[117]	Lena	33.4618	99.6085	-
GSCUD	Peppers	33.379	99.604	0.4991
GHCC	Peppers	33.478	99.605	0.499
Ref.[116]	Peppers	33.4211	99.6185	-
GSCUD	Barbara	33.570	99.6069	0.50
GHCC	Barbara	33.470	99.6269	0.500401
Ref.[118]	Barbara	33.48	99.60	0.4995

Tableau IV.3. Comparaison du NPRC, UACI et DH avec quelques références.

IV.7.2. Test d'entropie

Nous donnons dans le tableau (IV.4) les résultats de calcul d'entropie obtenus pour quelques images. D'après les résultats nous remarquons que la valeur de l'entropie est très proche de 8, cela signifie que le système de chiffrement est robuste face à une attaque d'entropie.

Image	Entropie (Image claire)	Entropie (Image chiffrée)
Lena	7.7725	7.9989
Cameraman	6.7379	7.9991
Peppers	7.6698	7.9997
Airplane	6.6639	7.9995

Tableau IV.4. Résultats du test d'entropie.

IV.7.3. Espace des clés

Le générateur GSCUD proposé possède cinq paramètres de contrôle a, b, c et k ainsi que trois conditions initiales x_0, y_0 et z_0. Nous pouvons ajouter le pas d'intégration h de la méthode Euler utilisée pour résoudre le système d´équation du générateur. Ainsi, nous obtenons un total de huit paramètres codés sur 32bits. Donc l'espace des clés H du cryptosysteme proposé peut être calculé comme suit :

$$H(a, b, c, k, x, y, z) = ((2)^{32})^{8} = (2)^{256} \qquad (IV.5)$$

Pour rappel dans la littérature, pour qu'un cryptosysteme basé-chaos résiste à l'attaque par force brute, il faut que son espace des clés soit supérieur ou égal à 2^{128} [109]. Donc l'espace des clés de notre cryptosysteme est assez large pour résister à l'attaque par force brute.

IV.7.4. Sensibilité des clés

Le résultat du test de sensibilité aux conditions initiales montre que le déchiffrement avec la valeur $X_0 = 0.01 + 10^{-12}$ de l'image chiffrée (Lena) avec la valeur X_0=0.01 ne permet pas la reconstitution correcte de l'image originale. De même pour le test de sensibilité aux paramètres, pour un déchiffrement avec la valeur $a = 10 + 10^{-14}$ de l'image chiffrée (Baboon) avec la valeur a=10. Donc, notre cryptosysteme est extrêmement sensible à la moindre variation soit dans les conditions initiales soit dans les paramètres de contrôle.

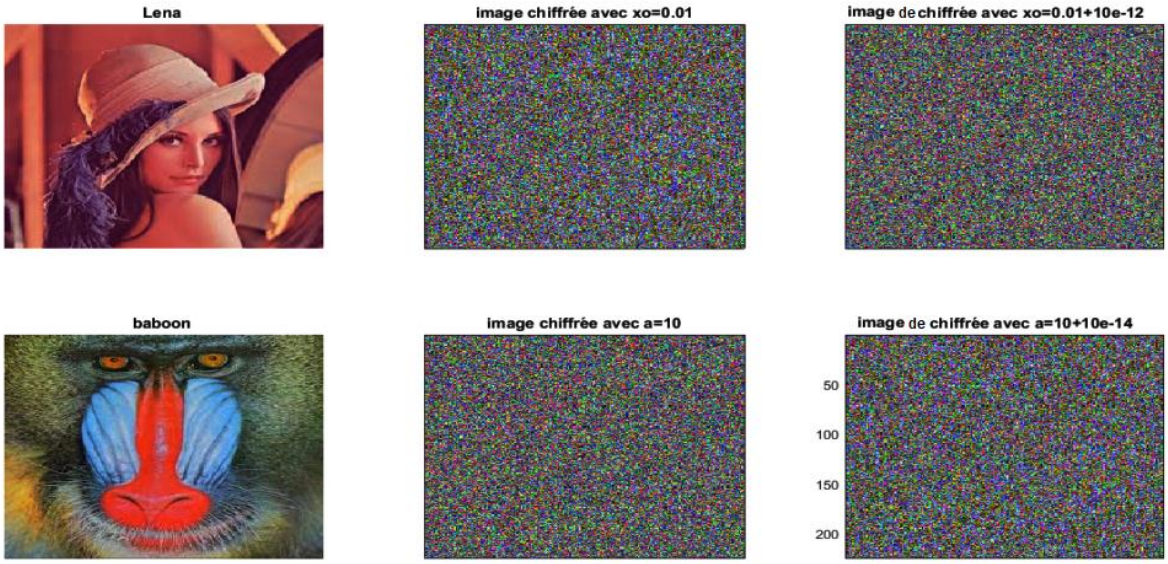


Figure IV.15. Sensibilité aux conditions initiales (haut) et aux paramètres (bas).

IV.7.5. Histogramme

Les histogrammes des images chiffrées (Lena et Mandrill) représentés sur les figures (IV.16) et (IV.17) présentent des niveaux d'amplitudes relativement plats par rapport à ceux des images originales, ce qui confirme leurs aspects aléatoires.

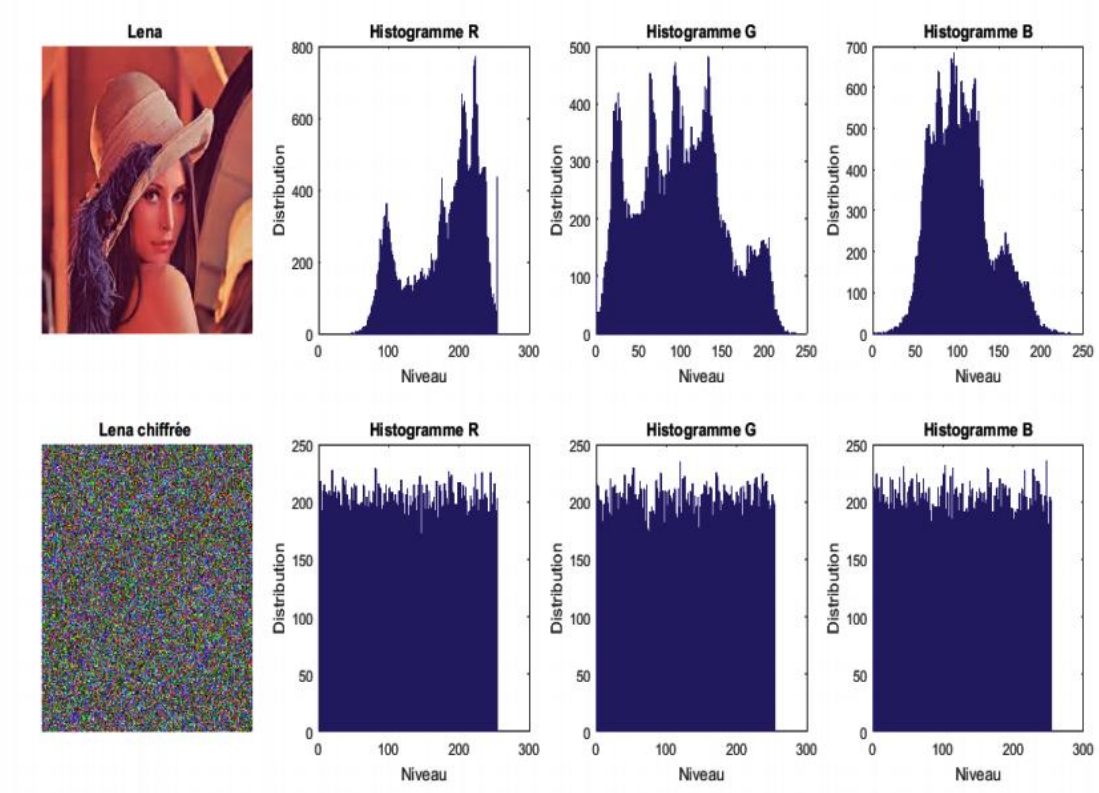


Figure IV.16. Analyse d'histogrammes d'image (Lena).

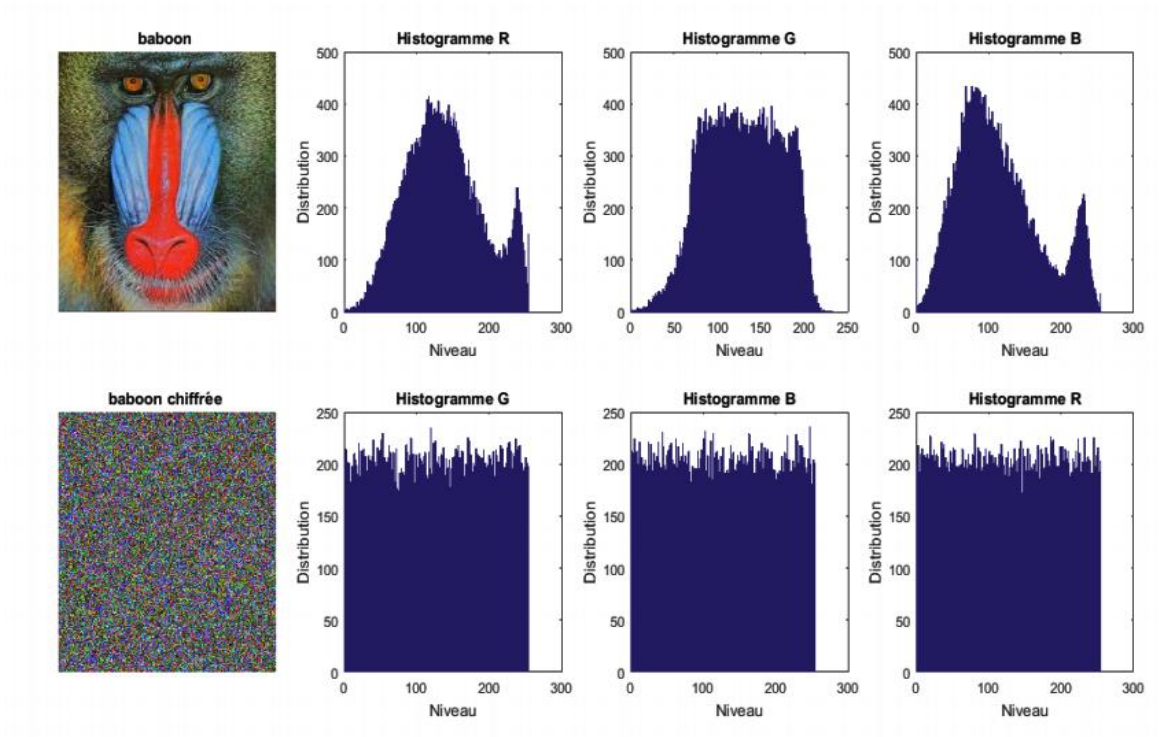


Figure IV.17. Analyse d'histogrammes d'image (Mandrill).

Le test de chi-carré, défini précédemment dans le chapitre II, appliqué sur les deux images est donné par le tableau (IV.5).

Image chiffrée	Test chi-carré
Lena 512*512*3	268.2738
Baboon 512*512*3	269.3847

Tableau IV.5. Test de chi-carré.

On voit bien sur le tableau que les valeurs des chi-carré calculés χ^2_{exp} sont inférieurs à la valeur du chi-carré théorique $\chi^2_{th} = 293.24$, ce qui confirme bien, statistiquement, l'uniformité des histogrammes des images chiffrées et montre que le cryptosystème proposé est robuste face à l'attaque d'histogramme.

IV.7.6. Corrélation des pixels adjacents

Les figures (IV.18) et (IV.19) montrent la corrélation des pixels adjacents dans les trois directions pour les images claires et chiffrées Peppers et Airplane respectivement. On remarque aisément que les nuages de points obtenus pour les images chiffrées sont totalement dispersés. Ce qui signifie que les pixels adjacents sont complètement décorrelés. Par conséquent, aucune estimation de la valeur d'un pixel à partir des valeurs de ses pixels adjacents n'est possible. Ceci indique l'efficacité du cryptosystème proposé.

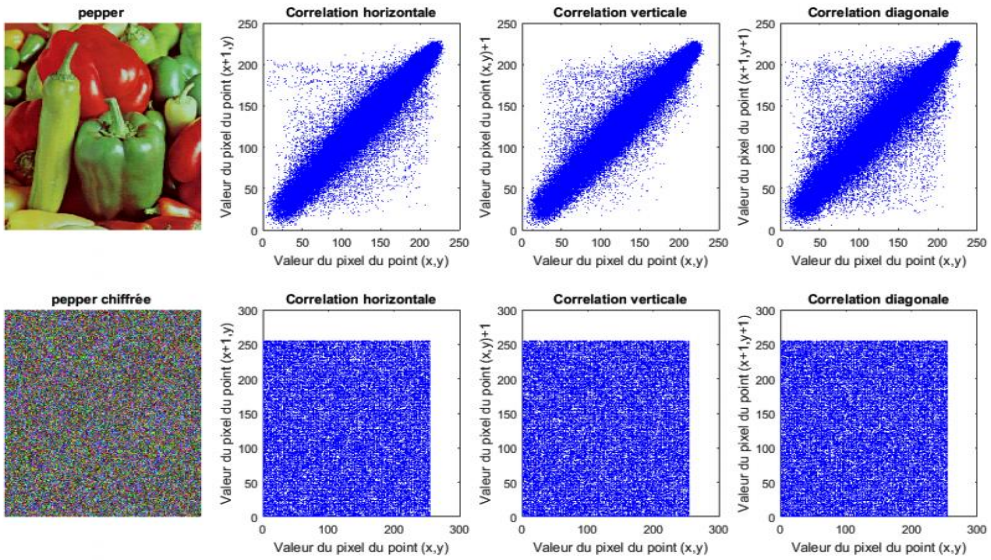


Figure IV.18. Corrélation des pixels adjacents dans l'image Originale et chiffrée (Peppers).

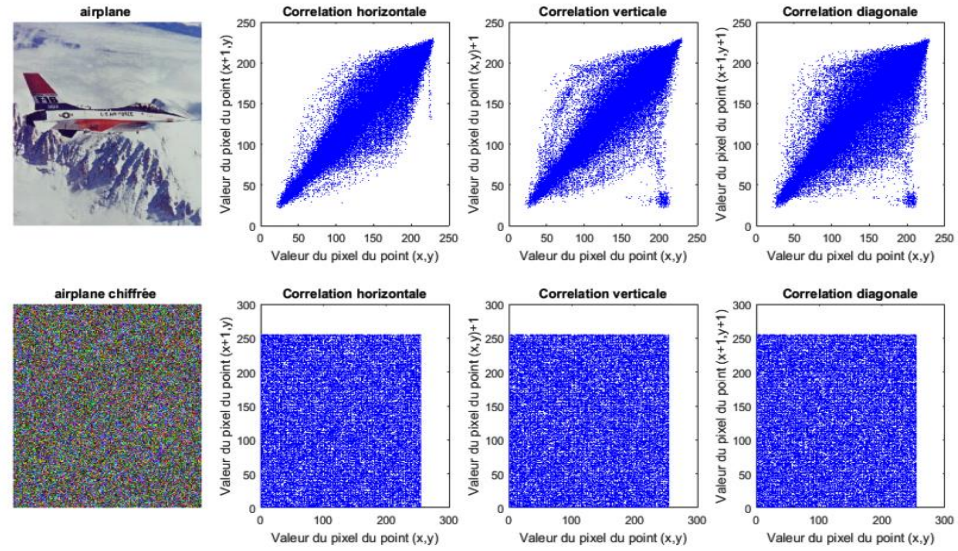


Figure IV.19. Corrélation des pixels adjacents dans l'image originale et chiffrée (Airplane).

Le tableau (IV.6) donne les coefficients de corrélation des pixels adjacents, dans les trois directions (horizontale, verticale et diagonale), calculés pour quelques images avec différents cryptosystèmes y compris le nôtre. Les résultats obtenus sont comparables.

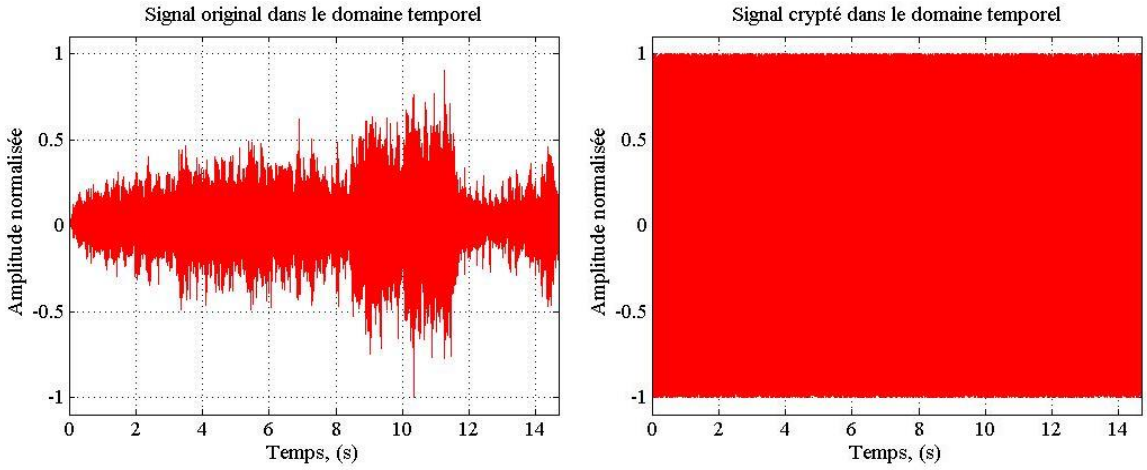
Algorithme	**Image**	**Correlation image originale**			**Correlation image chiffrée**		
		H	**V**	**D**	**H**	**V**	**D**
GSCUD	Lena	0.9752	0.98530	0.96483	-0.00033	-0.00054	-0.00212
GHCC	Lena	0.9752	0.98530	0.96483	-0.00022	-0.00043	-0.00162
Ref.[118]	Lena	0.9850	0.9719	0.9593	0.0093	0.0032	-0.0019
Ref.[119]	Lena	0.9685	0.96252	0.96462	0.0005	-0.0025	0.0028

Ref.[120]	Lena	0.9859	0.9741	0.9618	0.0052	00.0112	0.0034
GSCUD	Barbara	0.8927	0.95421	0.88393	-0.00004	0.0029	-0.00026
GHCC	Barbara	0.8927	0.95421	0.88393	-0.00003	0.00129	-0.00016
Ref.[118]	Barbara	0.9542	0.8927	0.8839	0.0022	0.0053	-0.0005
Ref.[121]	Barbara	0.9588	0.8991	0.8733	-0.0002	0.0081	-0.0011

Tableau IV.6. Les coefficients de corrélation des pixels adjacents.

IV.8. Analyse de sécurité du signal audio

Dans ce qui suit, nous allons évaluer l'efficacité et la robustesse de notre cryptosystème pour le chiffrement de la parole et ceci, en analysant un signal audio original représenté sur la figure (IV.20 (a)), plus le signal audio chiffré qui est représenté sur la figure (IV.20. (b)) dans le domaine temporel.

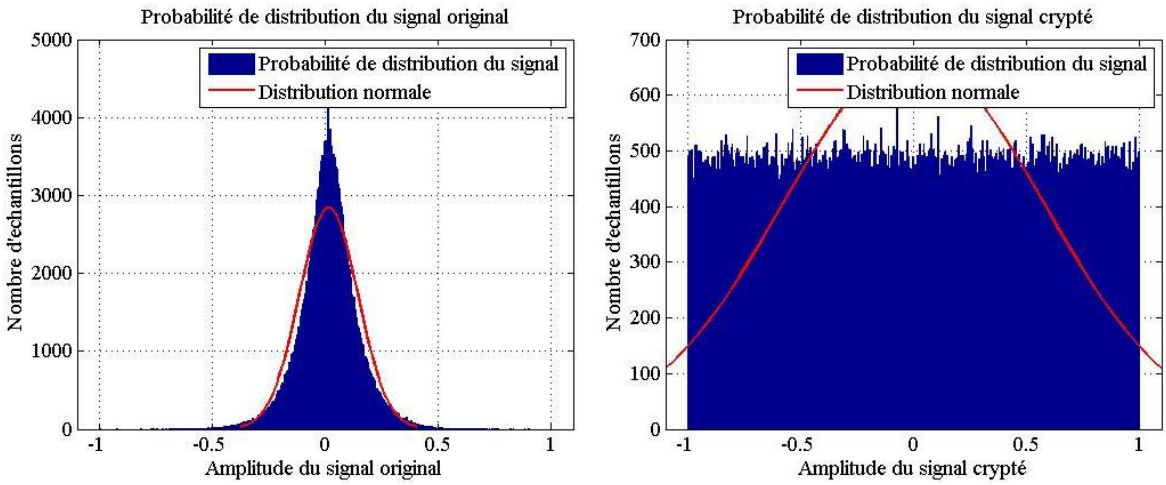


(a) (b)

Figure IV.20. Chiffrement de la parole : (a) signal parole original et (b) signal chiffré.

IV.8.1. Analyse d'histogramme

L'Histogramme est un outil d'étude statistique qui est utilisé pour illustrer l'efficacité de la diffusion de l'information dans les données chiffrées apportée par le cryptosystème utilisé. L'histogramme du signal original se présente sous forme d'une raie (figure IV.21(a)), alors que celui du signal chiffré est uniforme (figure IV.21(b)). Ceci illustre parfaitement la diffusion de l'information dans le chiffré qu'apporte notre cryptosystème.

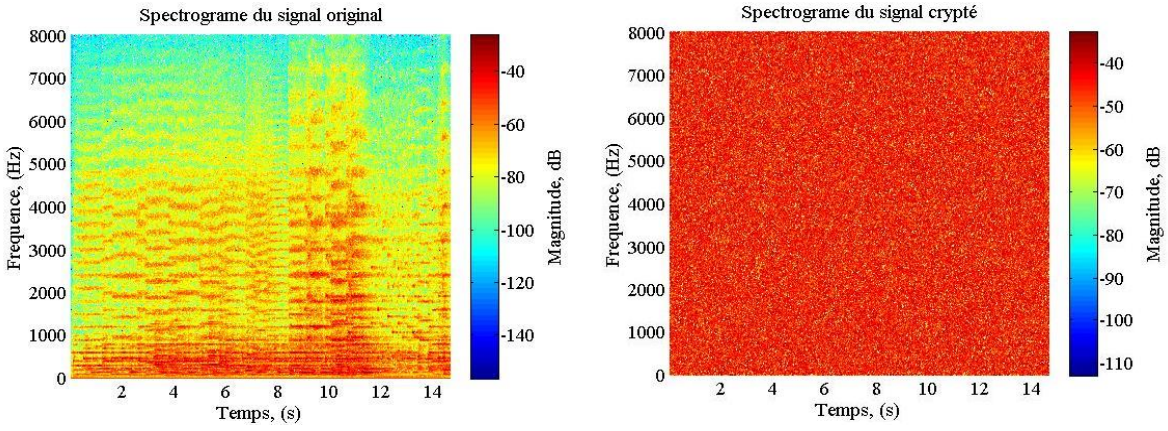


(a) (b)

Figure IV.21. Analyse d'histogramme du chiffrement de la parole : (a) l'histogramme du signal parole original et (b) l'histogramme du signal chiffré.

VI.8.2. Analyse de spectrogramme

L'analyse de la distribution temps-fréquence ou spectrogramme du signal de parole original (figure IV.22(a)) et chiffré (figure IV.22(b)), montre que le signal chiffré a un comportement fréquentiel uniforme en fonction du temps, contrairement au signal original, ou nous remarquons une certaine évolution régulière des fréquences en fonction du temps.



(a) (b)

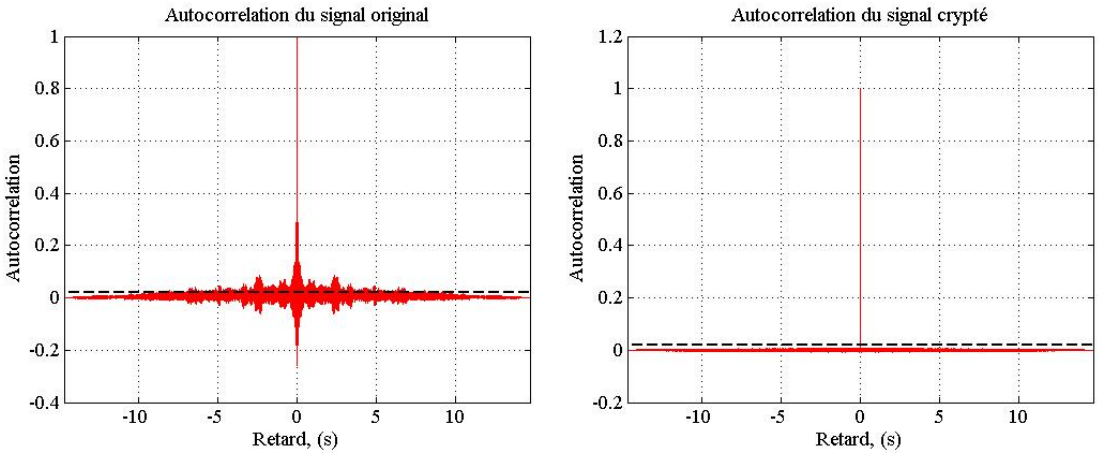
Figure IV.22. Spectrogramme : (a) signal parole original et (b) signal chiffré.

IV.8.3. Fonction d'autocorrélation

La fonction d'autocorrélation est un outil mathématique souvent utilisé en traitement du signal. Elle traduit la similitude d'un signal au niveau de la forme en fonction d'un décalage temporel τ. C'est une mesure de la ressemblance du signal avec lui-même au cours du temps.

Intuitivement, la corrélation est maximale pour un décalage nul ($\tau = 0$). Le principe de la corrélation est de pouvoir extraire l'information prédictive future à partir des valeurs précédentes du signal.

La figure (IV.23) présente la fonction d'autocorrélation du signal audio original (a) et son chiffré (b). Nous remarquons bien que l'autocorrélation du signal chiffré est similaire à celle d'un bruit blanc du fait qu'elle présente un pic en zéro et est nulle ailleurs. Par conséquent, la prédiction du signal à partir de son passé n'est pas possible.



(a) (b)

Figure IV.23. Autocorrélation : (a) signal audio original et (b) signal audio chiffré.

IV.9. Structure et déploiement de l'architecture proposée

La Figure (IV.24) illustre l'architecture de notre système de réseaux de capteurs sans fil, composée de quatre entités. D'un côté, nous avons deux nœuds de capteurs, chargés de mesurer l'humidité et la température (DTH22), un Smartphone pour la transmission du flux audio/vidéo temps réel et de l'autre côté un serveur qui reçoit et traite ces données. Comme présenté sur la figure (IV.24) l'architecture proposée utilise trois différentes technologies de transmission en l'occurrence le Wifi pour le flux audio/vidéo, le Bluetooth et l'Xbee pour les mesures d'humidité et de température.

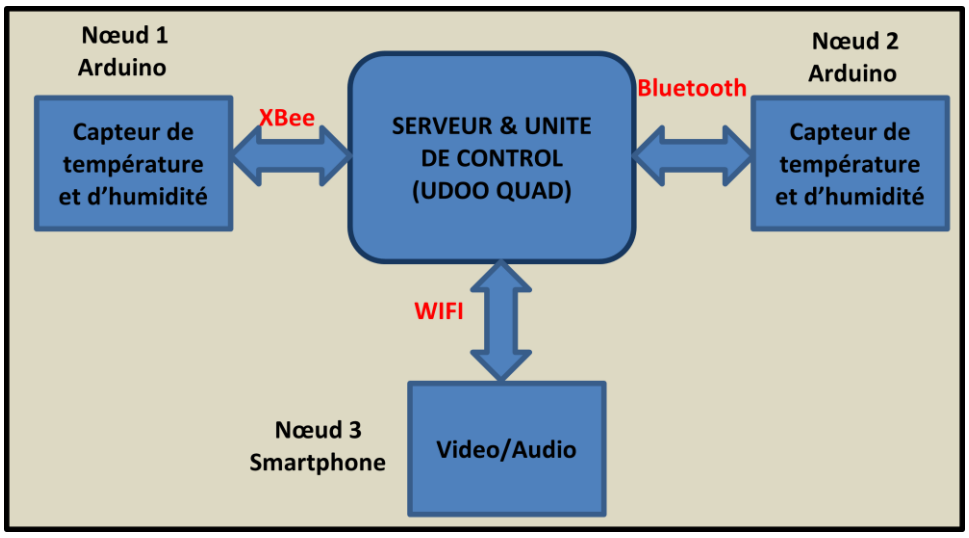


Figure IV.24. Bloc diagramme du système.

Maintenant, nous allons d´écrire les cryptosystèmes proposés pour chiffrer et transmettre les données issues des différents capteurs ainsi que le processus de déchiffrement réalisé au niveau du serveur comme schématisé sur la figure (IV.25).

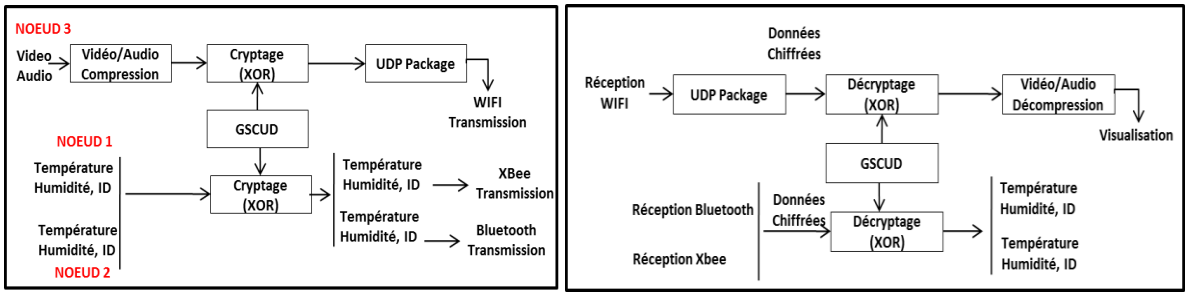


Figure IV.25. Processus de chiffrement/déchiffrement**.**

Mais d'abord nous allons définir ce qui est commun pour eux à savoir :

1. Nous allons définir pour chaque nœud (capteur) un identifiant « ID » unique et secret, pour qu'il puisse s'identifier au niveau du contrôleur central. Cela permet de vérifier l'authentification qui est un des critères de sécurité ;

2. Nous allons proposer une solution pour le problème de la synchronisation ;

3. Nous allons décrire les fonctions de permutations utilisées.

IV.9.1. Solution proposée pour le problème de la synchronisation

Pour notre travail, nous implémentons le même générateur chaotique avec les mêmes conditions initiales et paramètres au niveau de l'émetteur ainsi que le récepteur. Nous proposons comme méthode de synchronisation l'utilisation d'un compteur « CTR » au niveau

de l'émetteur. Le rôle de ce compteur se manifeste au niveau de la condition initiale du générateur chaotique par exemple pour x(0). Initialement, le CTR est égale à un et à chaque fois qu'on termine le chiffrement d'une donnée, le compteur s'incrémente de un (CTR = CTR + 1). La relation qui lie la condition initiale à la valeur du CTR est définie par :

$$x(0) = x(0) + \beta \times CTR \qquad (IV.6)$$

Avec x(0) est la condition initiale, β un facteur que nous définissons et CTR la valeur du compteur. Le β pour notre cas, est considéré comme étant une clé secrète connue uniquement par l'émetteur et le récepteur. Sa valeur doit être extrêmement petite et cela pour deux raisons:

1. Le générateur chaotique est extrêmement sensible au moindre changement dans les conditions initiales ;

2. Un grand β peut causer la perte du régime chaotique. Voici un schéma illustrant le procédé à l'émission :

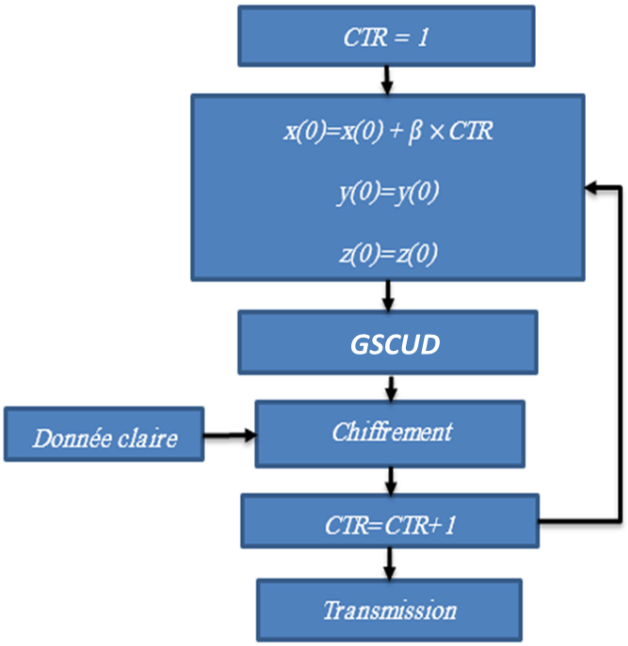


Figure IV.26. Principe de la synchronisation coté émission.

Lors de la transmission, la trame envoyée contiendra la valeur du compteur pour que le récepteur puisse la récupérer. Ainsi à la réception, le récepteur extrait d'abord la valeur du CTR pour pouvoir déchiffrer correctement comme montré sur la figure (IV.27).

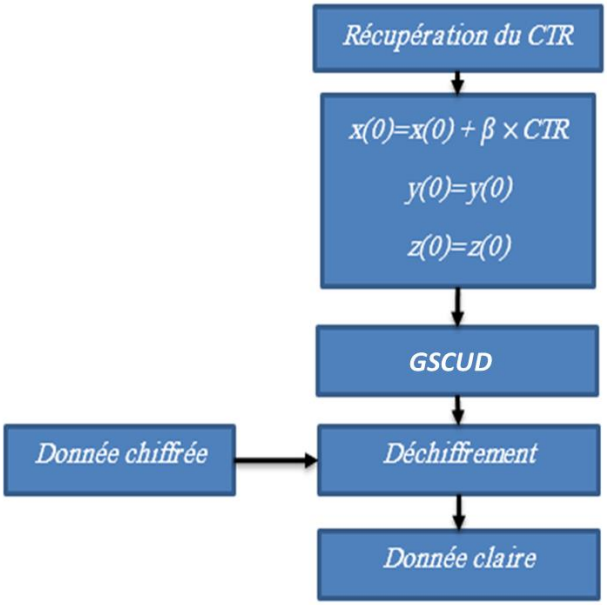


Figure IV.27. Principe de la synchronisation coté réception.

IV.9.2. Fonctions de Permutations

En cryptographie la technique de diffusion est souvent réalisée par des permutations de blocs ou des permutations binaires. Dans le cryptosysteme proposé, nous allons intégrer deux fonctions de permutations, une utilisée dans le chiffrement du flux audio/vidéo et l'autre dans le processus du chiffrement des mesures de température et d'humidité.

1. Fonction de permutation FP1.

La première fonction de permutations FP1 qui sera appliquée dans le chiffrement des données de température et d'humidité est basée sur le codage affine décrit par la relation suivante :

$$f(x)=(a*x+b)\ mod\ (N). \qquad \text{(IV.7)}$$

Avec a et b des entiers représentant la clé et N la taille du tableau à permuter.

Notons que pour a=1, b=3 et N=26 on obtient le célèbre chiffre de César.

Un exemple de permutation d'un tableau de longueur N=16 est représenté dans la figure (IV.28) pour a=7 et b=2.

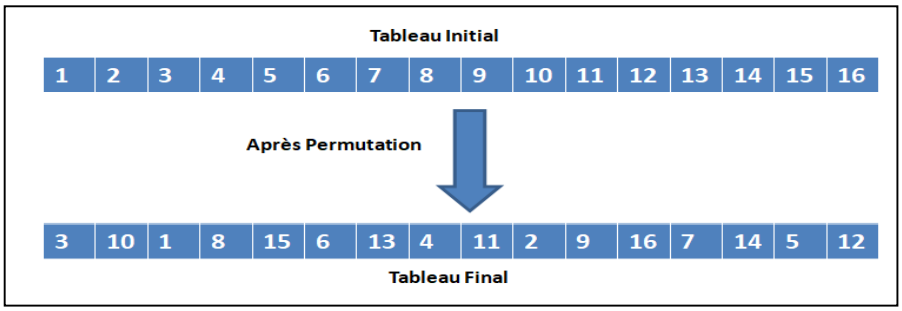


Figure IV.28. Exemple de permutation FP1 d'un tableau.

1. Fonction de permutation FP2

La deuxième fonction de permutations FP2 qui sera appliquée dans le chiffrement du flux audio/vidéo, inspirée directement de l'algorithme du chiffrement par flot RC4 [137] (Rivest Chipher 4), est donnée par l'algorithme suivant :

Algorithm1 : Algorithme de Permutation

Entrée : tableau ***A*** de taille ***N,*** carte chaotique Map(r, x_0)

pour i = 0 jusqu'à N_{-1} faire:

K = Map(r, x_0)*1000 ; // K est la clé générée par le générateur chaotique Map de paramètre r et de condition initiale x_0 .

$j = (j + i + K)\ mod\ \boldsymbol{N}$;

Permuter tab[i] et tab[j] ;

fin

Sortie : tableau ***A*** permuté de taille ***N***

Un exemple de permutation d'un tableau de longueur N=576 est illustré dans la figure (IV.29).

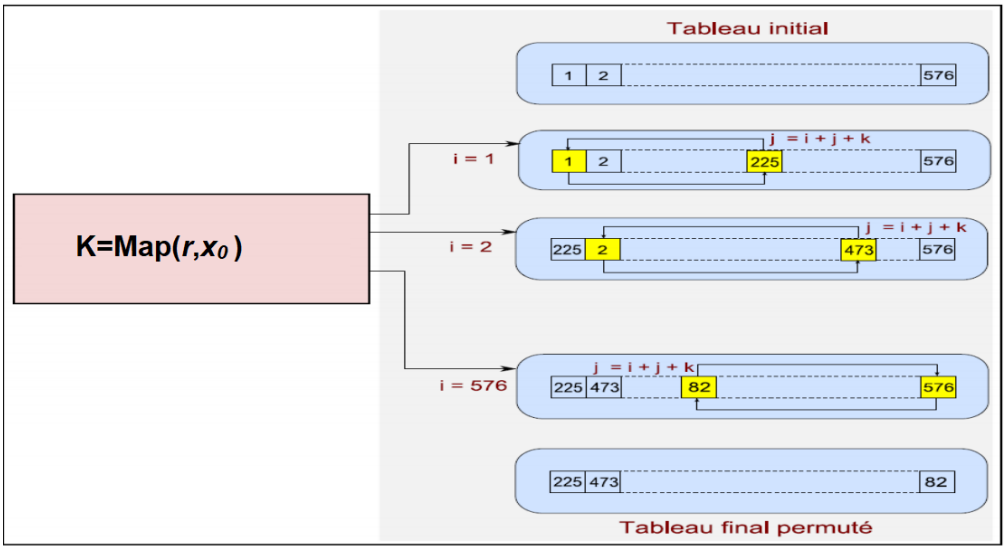


Figure IV.29. Exemple de permutation FP2 d'un tableau.

IV.10. Cryptosysteme pour la sécurisation des données du DHT22

Dans cette section, nous abordons la sécurisation des données issues des capteurs de température et d'humidité qui seront transmises à travers l'Xbee et le Bluetooth. Pour cela, nous désignons la température par la lettre T, l'humidité par la lettre H, le compteur par CTR, l'identifiant secret par ID et le check somme qui va servir pour le contrôle d'intégrité par CRC. Nous allons coder T, H et CTR chacun sur quatre octets, le CRC sur deux octets et enfin l'ID sur deux octets.

IV.10.1. Au niveau de l'émission

Le processus de chiffrement est représenté sur la figure (IV.30).

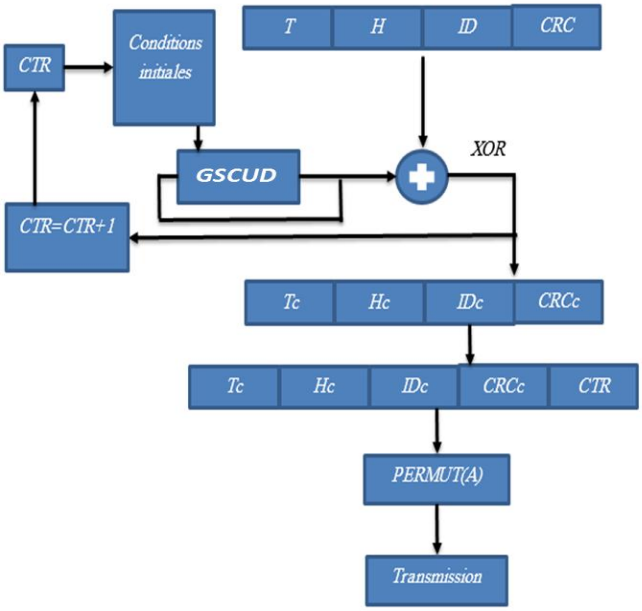


Figure IV.30. Processus de cryptage.

L'Algorithme2 ci-dessous donne et illustre plus de détails sur le processus de cryptage.

Algorithm2 : Processus de cryptage

Entrée : température T et humidité H

1 : CRC = (ID ⊕ T ⊕ H) //⊕ représente l'opération XOR et ID indique le code d'identification unique et secret de chaque nœud.

2 : V = ID ||T ||H|| CRC // || signifie l'opération de concaténation. Le tableau V forme le texte en claire.

3 : **C=V ⊕ K** // K représente l'ensemble des clés de chiffrement issues du générateur GSCUD avec la condition initiale $x(0) = x(0) + \beta \times CTR$.

4 : A= CTR|| **C** // A constitue le tableau final chiffré de taille ***L*.**

5 : A ← Permutation du tableau avec FP1(***L*)**

6 : CTR= CTR+1

Sortie : Envoi du tableau A vers le serveur.

IV.10.2. Au niveau de la réception

Le processus de décryptage, coté serveur est illustré par la figure (IV.31) sous forme d'organigramme et détaillé par l'Algorithme 3.

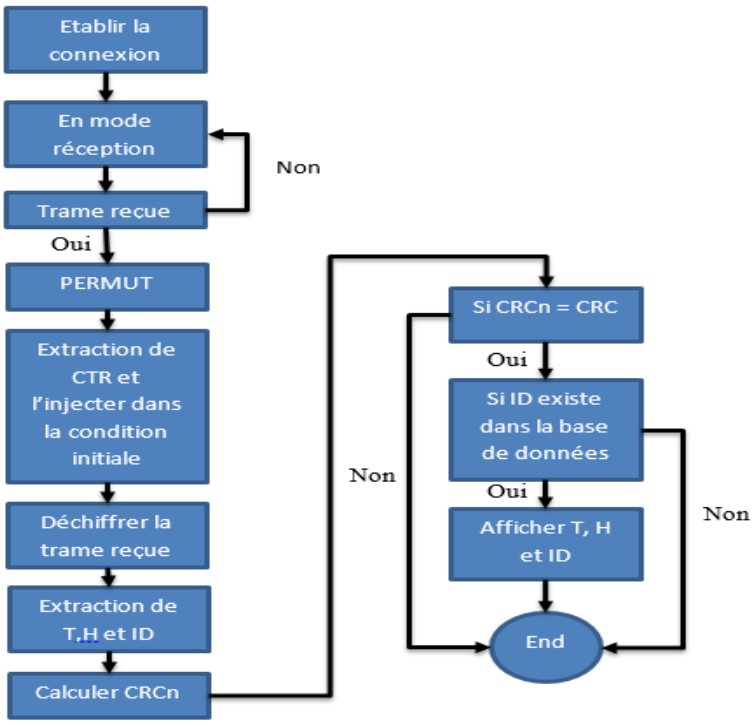


Figure IV.31. Processus de décryptage.

Algorithme3 : Processus de décryptage

Entrée : Tableau chiffré A

1 : A←Permutation inverse FP($\boldsymbol{L}$)$^{-1}$.

2 : Extraire la valeur du CTR $_{i+1}$ // en comparant CTR $_{i+1}$ et le dernier CTR $_{i}$ enregistré le serveur vérifie la fraicheur des données.

3 : Si (CTR $_{i+1}$ == CTR $_{i}$**)** // le serveur rejette le message et retourne à1'étape 1.

Si non x(0) = x(0) + β× CTR $_{i+1}$.

4 : M=A⊕K. // M tableau déchiffré avec l'ensemble des clés de déchiffrement K.

5 : Calcul du CRC_n du tableau M.

6 : Extraire ID et CRC à partir du M.

7 : Si (CRC_n≠ CRC) or ID non enregistré dans la base de donnée le serveur rejette le message.

Sinon le server sauvegarde CTR $_{i+1}$ et supprime le dernier CTR $_{i}$ enregistré.

Sortie: Affichage et enregistrement des mesures de température T et d'humidité H.

IV.11. Cryptosystème pour la sécurisation des données du smartphone

Dans cette section, nous allons nous intéresser à la sécurisation du flux audio/vidéo issu du smartphone qui sera transmis vers le serveur via une connexion WIFI en exploitant le protocole UDP.

IV.11.1. Sécurisation du flux vidéo

Le processus de chiffrement du flux vidéo est illustré sur la figure (IV.32).

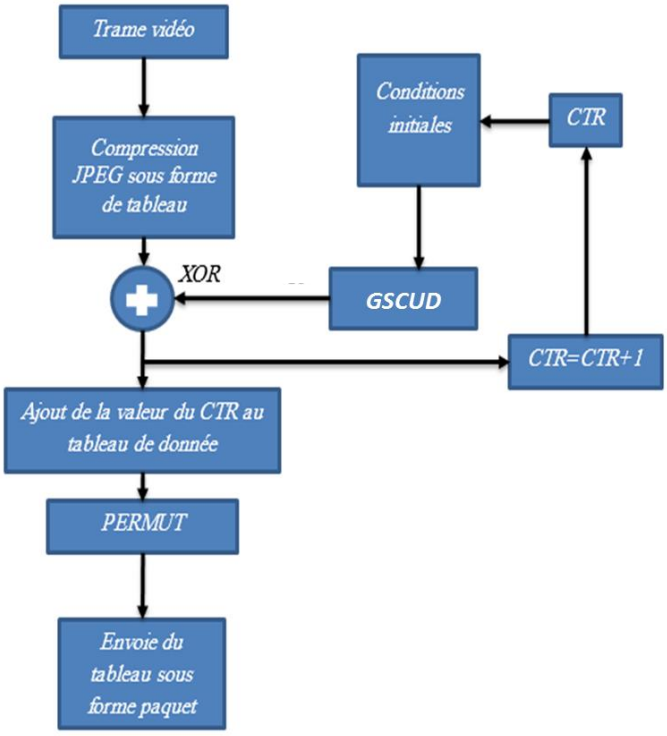


Figure IV.32. Processus de chiffrement de la vidéo.

Ainsi le chiffrement consiste à :

- Acquisition de l'image à partir du flux vidéo de la camera.
- Compression de l'image sous le format jpeg (Join Picture Expert Group).
- L'image compressée sera convertie en un tableau d'octets.
- Injection de la valeur du CTR dans les conditions initiales de GSCUD.
- Xor de chaque octet avec l'une des clés du GSCUD.
- Ajout de la valeur du CTR au tableau de données.
- Permutation du tableau de données en utilisant la fonction FP2.
- Envoi de l'image chiffrée sous forme de paquets UDP.

Quant au processus de déchiffrement il est décrit sur la figure (IV.33).

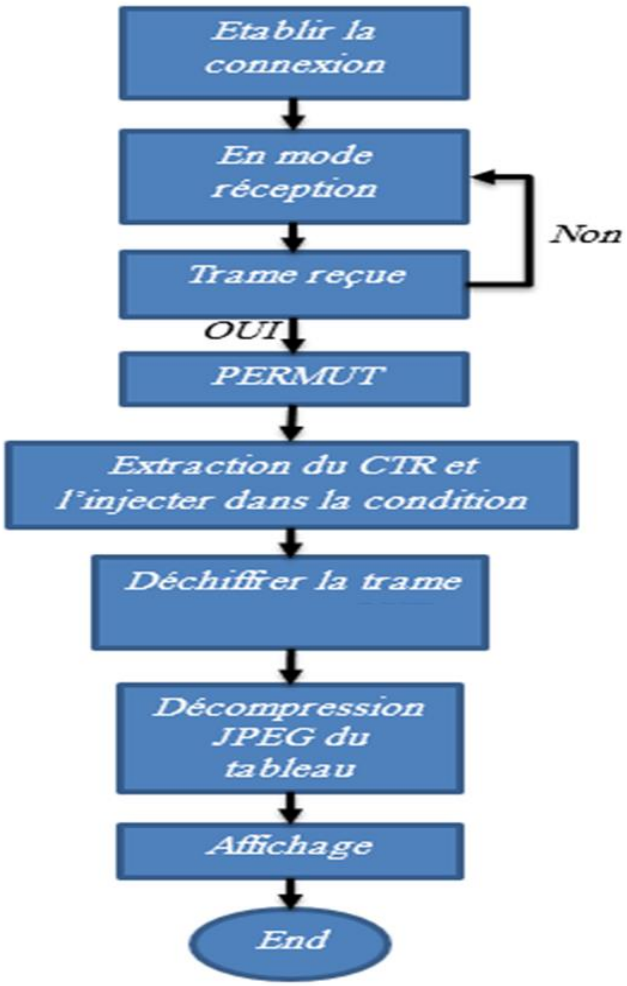


Figure IV.33. Processus de déchiffrement de la vidéo.

Le déchiffrement consiste à :

- Une fois que la trame de donnée est reçue, on permute le tableau en utilisant la fonction inverse $FP2^{-1}$.
- On extrait la valeur du l'ID pour identification.
- On extrait la valeur du CTR pour l'injecter en $x(0) = x(0) + \beta \times CTR$.
- Lancement du générateur GSCUD avec les bonnes conditions initiales.
- Xorer les octets reçus avec les différentes clés générées.
- Décompression JPEG et affichage de l'image.

IV.11.2. Sécurisation du flux audio

Le processus de chiffrement/déchiffrement du flux audio est identique à celui de chiffrement/déchiffrement du flux vidéo à l'exception du type de compression utilisée qui est, en l'occurrence, la norme de compression audio G.711 définie par l'UIT-T (Union

Internationale des Télécommunications) [138]. Cette norme définit deux algorithmes de compression qui sont :

- La loi μ, utilisée principalement en Amérique du Nord et au Japon
- La loi A, utilisée en Europe et dans le reste du monde.

Ces deux lois sont en fait, des systèmes de quantification logarithmique du signal audio.

Toutefois, la loi A est plus simple et moins complexe que la loi μ.

IV.12. Procédure d'initialisation du générateur GSCUD

Il convient également de noter qu'à chaque mise sous tension, tous les nœuds capteurs exécutent une procédure d'initialisation du générateur **GSCUD.** Cela garantit qu'un début de processus de cryptage ne sera pas le même pour chaque nœud, puisque les valeurs de CTR, par conséquence les conditions initiales, seront toujours différentes pour les générateurs GSCUD de chaque nœud, renforçant ainsi la sécurité du système. Pour cela pour les nœuds de capteurs nous utilisons une entrée aléatoire, comme la valeur renvoyée par la fonction analogRead() sur une broche analogique laissée en l'air, pour initialiser le compteur CTR.

Ainsi la valeur initiale du CTR est donnée par :

$$CTR=(analogRead(A1))*1000 \quad (IV.8)$$

Avec A1 entrée analogique non connectée.

Concernant l'initialisation du générateur **GSCUD** pour le chiffrement du flux audio/vidéo nous exploitons la donnée très bruitée et aléatoire des accéléromètres intégrés dans le Smartphone [139].

Ainsi la valeur initiale du CTR est donnée par :

$$CTR= floor(\lambda*10^{10}) mod\ (2^{32}) \quad (IV.9)$$

Avec λ valeur de l'accéléromètre en m/s^2 et floor(u) représente la partie entière de u.

IV.13. Réseau de capteur réalisé

Dans cette section, nous allons décrire le réseau de capteur réalisé en détaillant les différents nœuds, le type de transmission de données utilisé ainsi que la carte UDOO QUAD qui se charge de la réception et la gestion des données reçues de chaque nœud. L'affichage

des différentes données se fera sur une IHM dédiée que nous avons développée en langage de programmation JAVA.

IV.13.1. Maquette expérimentale

La maquette expérimentale réalisée comprenant les différents nœuds ainsi que la carte UDOO QUAD qui fait office de serveur est illustrée sur la figure (IV.34).

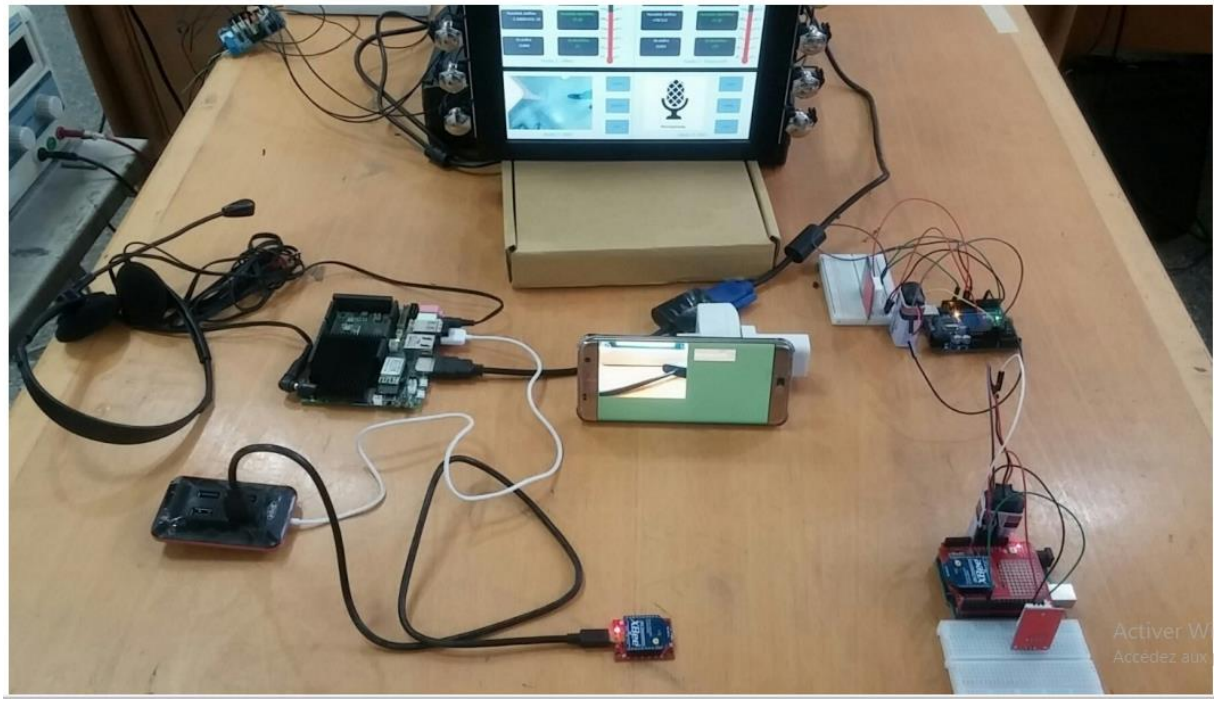


Figure IV.34. Maquette expérimentale.

IV.13.2. Description des différents nœuds

La figure (IV.35) montre la disposition et la composition des différents nœuds utilisés pour la réalisation des réseaux de capteurs sans fil.

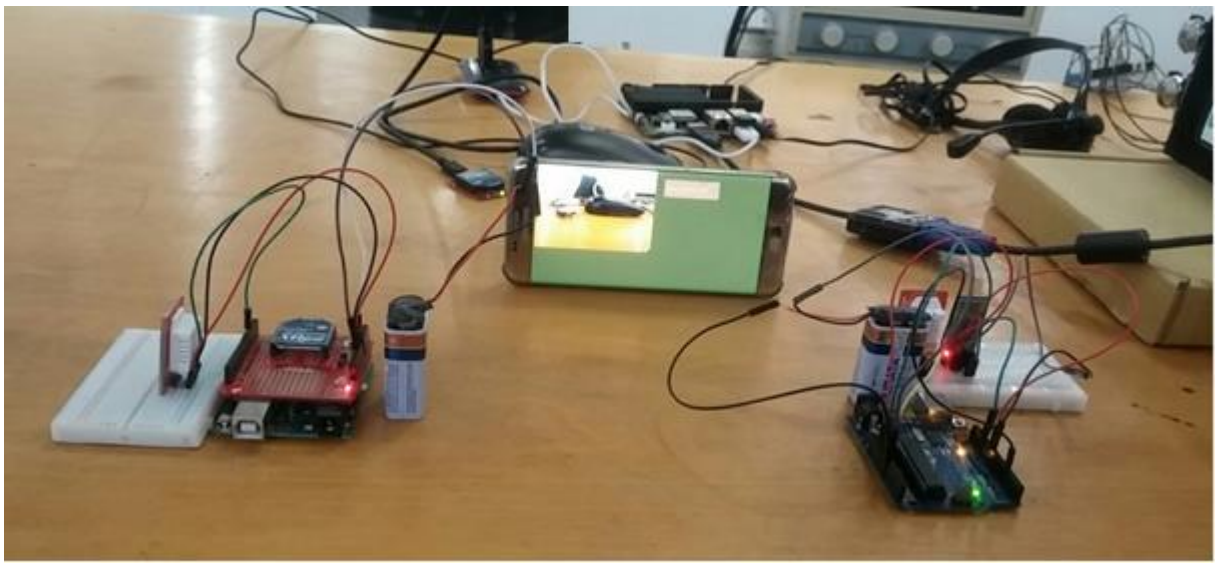


Figure IV.35. L'ensemble des nœuds.

Le détail sur la composition de chaque nœud est comme suite.

Nœud 1 :

La figure (IV.36) montre les différents composants utilisés pour réaliser ce nœud qui sont :

1. Le capteur de température et d'humidité DHT22.

2. Un module Bluetooth HC05.

3. Une carte arduino Uno.

4. Une batterie 9 volts pour alimenter la carte arduino.

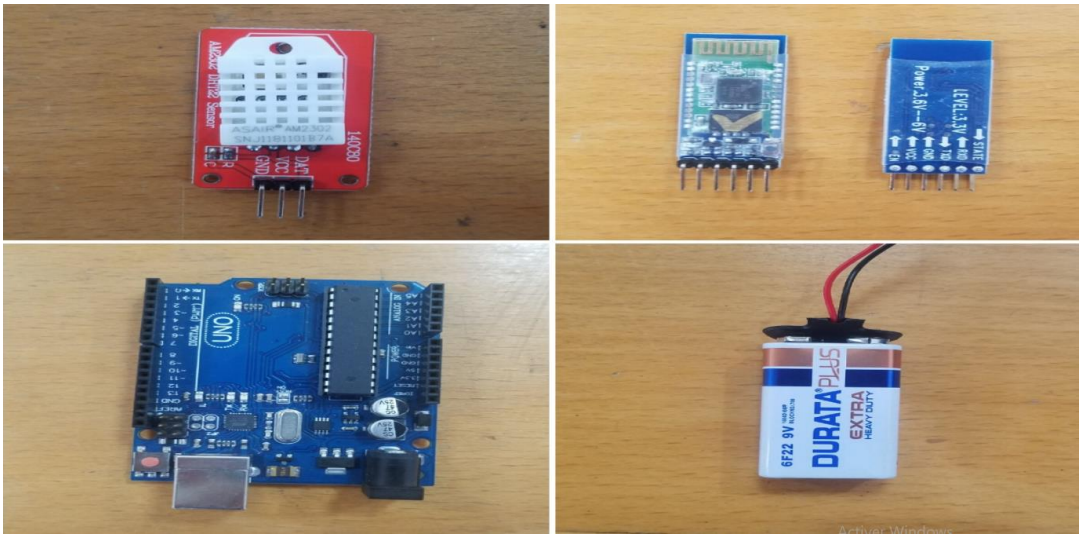


Figure IV.36. Composants du nœud 1.

Le DHT22 et le HC05 sont relies à la carte arduino comme le montre la figure suivante :

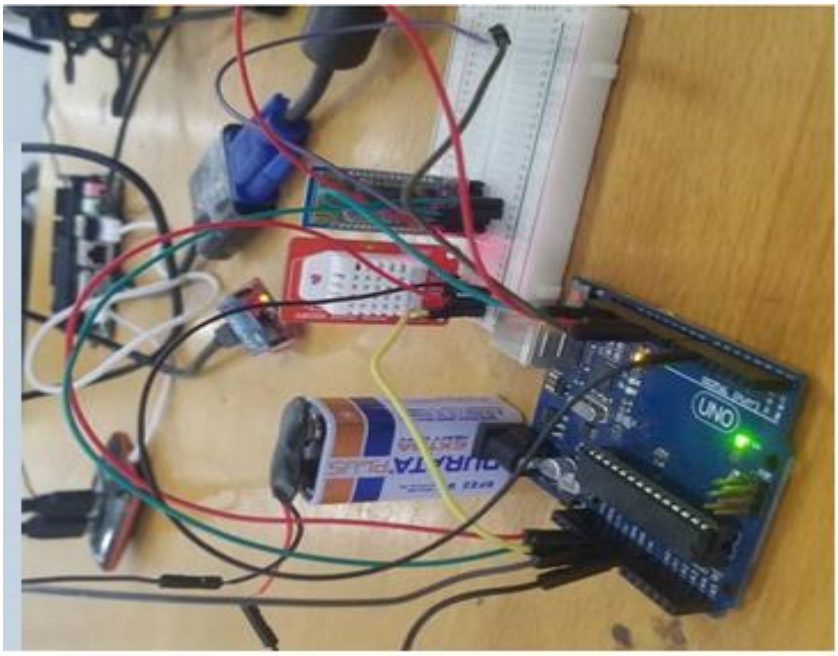


Figure IV.37. Réalisation du nœud 1.

La carte arduino récupère les valeurs de la température et de l'humidité et utilise le cryptosysteme pour le chiffrement pour enfin transmettre les données grâce au module HC05.

Nœud 2 :

Les différents composants utilisés pour réaliser ce nœud sont les mêmes que ceux du nœud1 à l'exception de la présence d'un module XBee à la place du module Bluetooth. La figure (IV.38) montre comment sont reliés le DHT22 et le module XBee à la carte arduino.

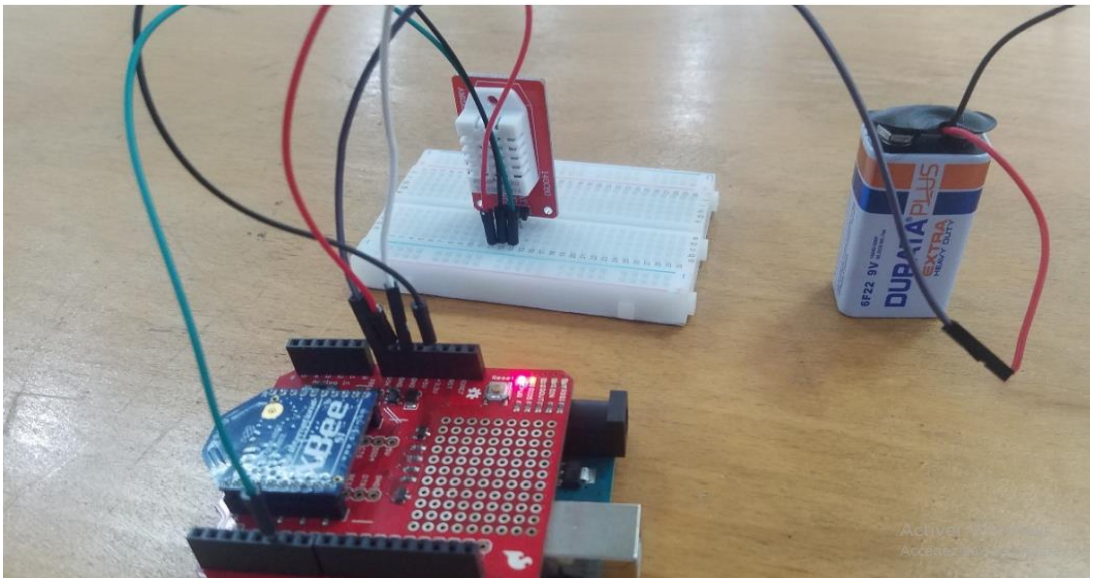


Figure IV.38. Réalisation du nœud 2.

La carte arduino récupère les valeurs de la température et de l'humidité et utilise le crypto système pour le chiffrement puis transmet les données à travers le module XBee.

Nœud 3 :

Ce nœud est constitué principalement d'un Smartphone qui permet la transmission audio/vidéo en temps réel à travers le WIFI. La figure (IV.39) montre l'interface de l'application réalisée.

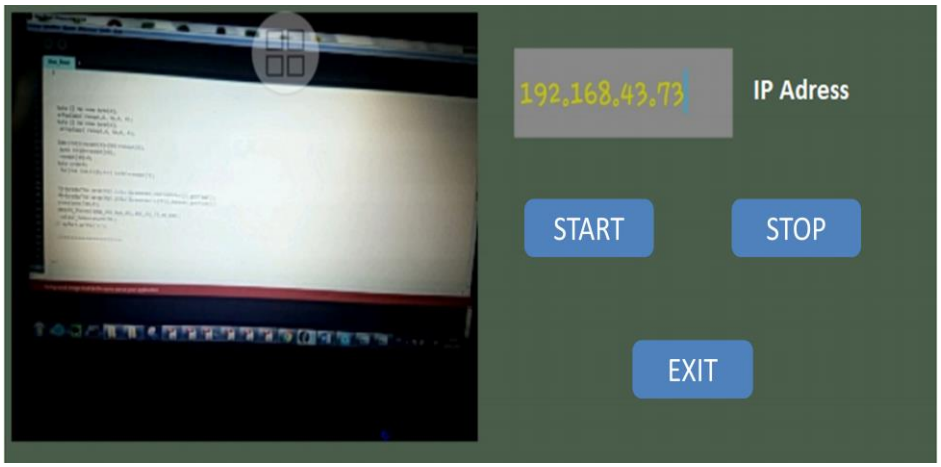


Figure.IV.39. L'application Android développée.

IV.13.3. Description de la carte UDOO QUAD (Serveur)

UDOO QUAD est un mini-ordinateur capable d'exécuter un système d'exploitation Android ou Linux. Il est équipé d'un processeur ARM i.MX6 Freescale et doté d'un microcontrôleur intégré Arduino Due basé sur le processeur ATMEL SAM3X8E, le tout monté sur la même carte [140] comme illustré sur la figure suivante.

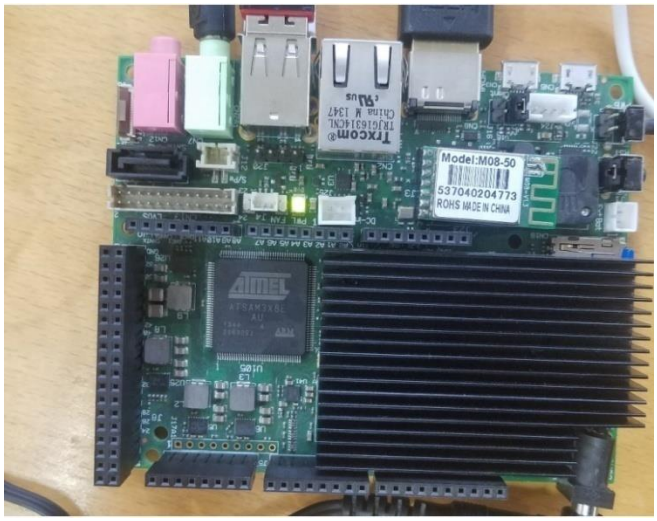


Figure IV.40. La carte UDOO QUAD.

Pour pouvoir recevoir les différentes données transmises par les différents nœuds, on a associé à la carte UDOO un module XBee et un module Bluetooth. Pour la réception des

données du nœud 3 la carte dispose d'un module Wifi intégré. On a aussi rajouté un écran pour l'affichage et un casque audio comme illustré sur la figure (IV.41).

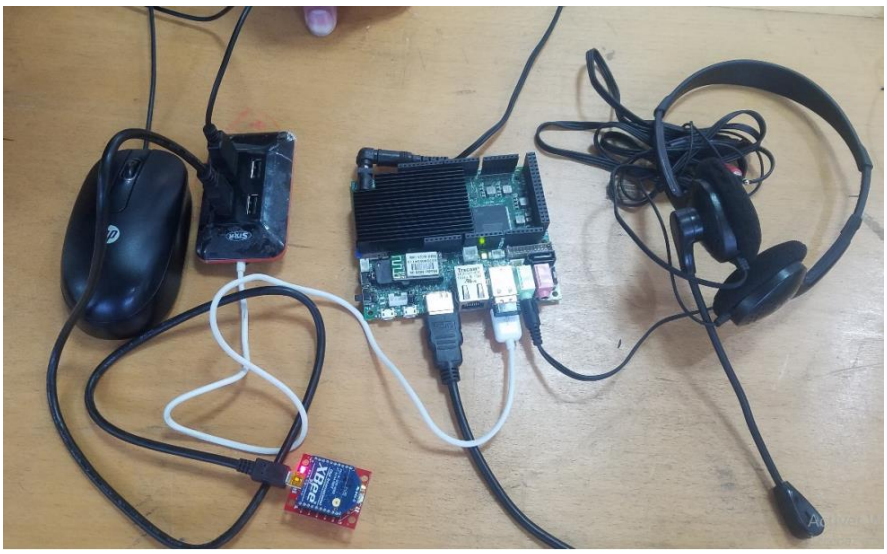


Figure IV.41. Montage du serveur.

IV.13.4. IHM développée

Au niveau de la carte UDOO QUAD, nous avons développé, une interface IHM qui affiche les données reçues avant et après déchiffrement correspondantes aux différents nœuds.

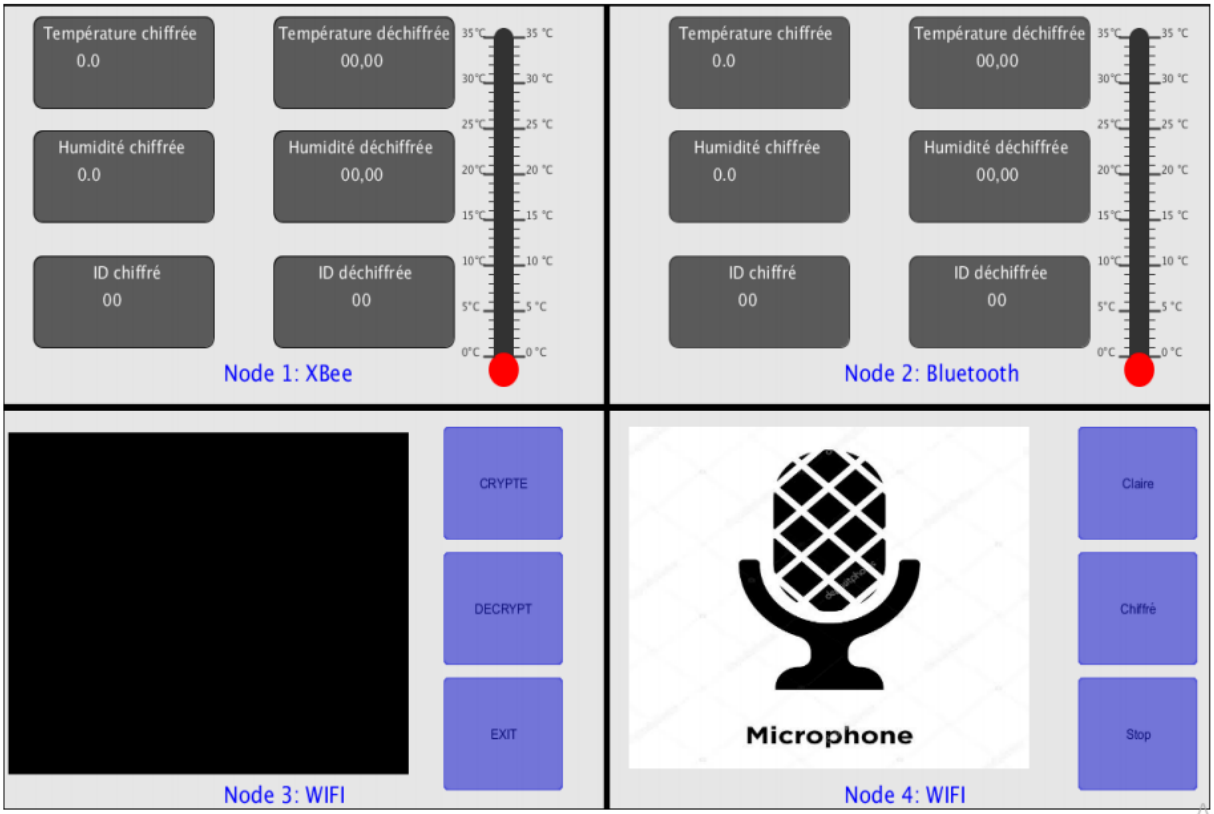


Figure IV.42. IHM serveur.

IV.14. Tests de validation et résultats

Dans cette section, nous allons montrer des figures qui illustrent la réception des données. La figure (IV.43) montre L'IHM complète.

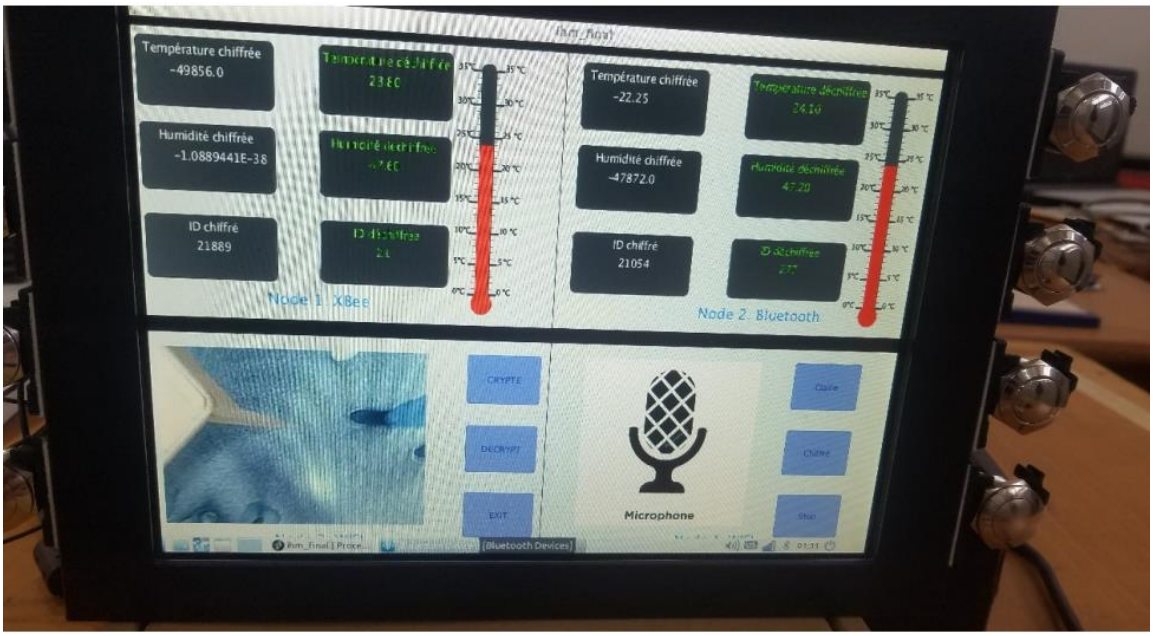


Figure IV.43. IHM complète.

Comme le montre la figure (IV.43), toutes les données déchiffrées sont affichées en temps réel et la gestion du processus du déchiffrement est faite d'une manière intelligente. En effet, bien que nous ayons trois normes de transmission sans fil différentes (WIFI, Bluetooth et XBee), c-a-d un réseau de capteurs hétérogène, notre programme affiche tous les résultats sans interruption, même si une donnée d'un vecteur est perdue, le reste des données continuent à s'afficher correctement. La figure (IV.44) affiche les données chiffrées et déchiffrées reçues des nœuds 1et 2.

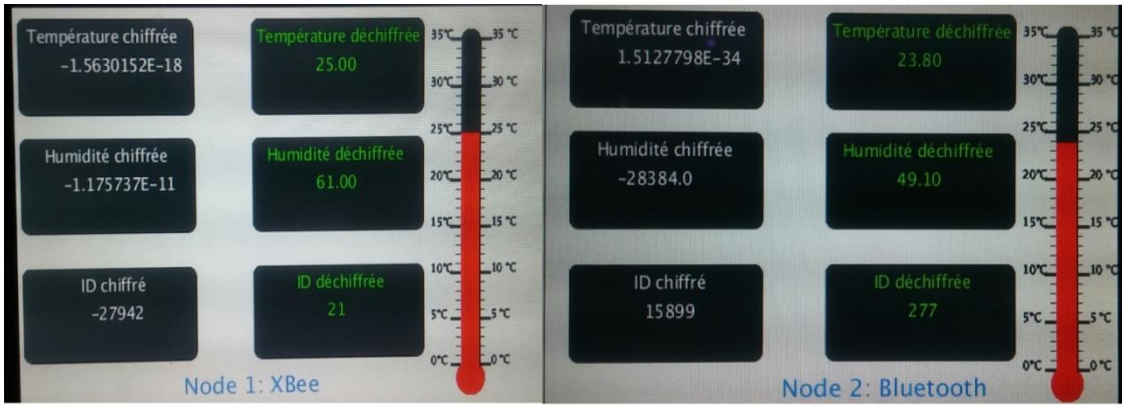


Figure IV.44. Données des nœuds 1 et 2.

La figure (IV.45) montre la vidéo chiffrée et déchiffrée issue du smartphone via la connexion WIFI. Le résultat ainsi obtenu montre clairement que le serveur arrive à déchiffrer convenablement et afficher la vidéo en temps réel avec une bonne qualité et faible latence. Ce test nous a permis en outre, de valider le bon fonctionnement et la robustesse de la technique de synchronisation proposée.

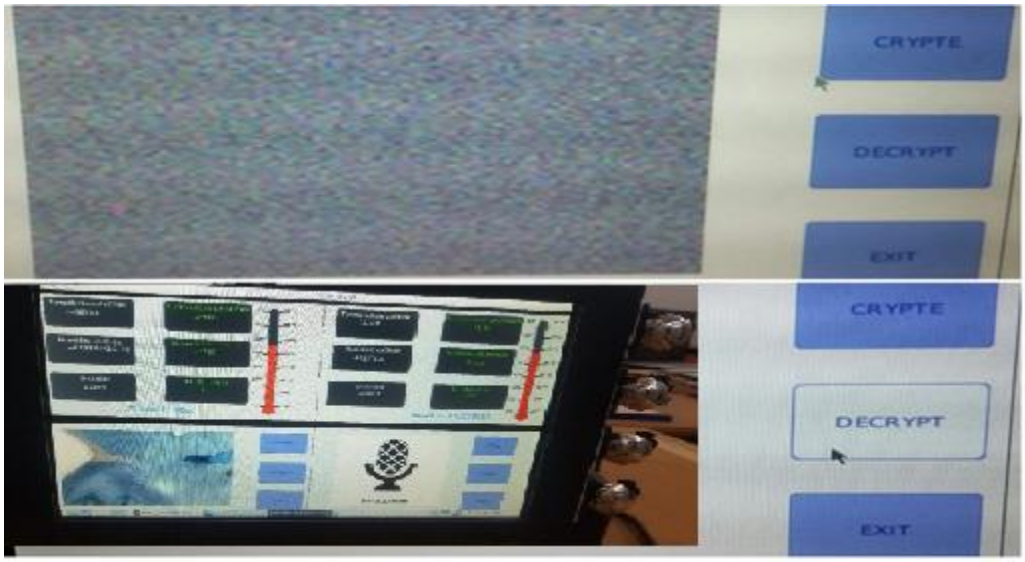


Figure IV.45. Données du nœud 3 audio/vidéo temps réel.

IV.15. Analyse et Performance du Schéma Cryptographique proposé

Dans cette section une analyse de performance de la méthode proposée est effectuée afin de vérifier que toutes les principales exigences de sécurité des données à savoir la confidentialité, l'authentification, la non-répudiation, l'intégrité, la résilience et la fraîcheur des données sont respectées.

- ✓ **Confidentialité** : Le schéma de cryptage que nous proposons assure la confidentialité des communications entre les nœuds et le serveur par l'emploi d'un chiffrement utilisant une grande clé $H=2^{\wedge 256}$ bits. La probabilité qu'un intercepteur réussisse à trouver la clé par attaque par force brute est donc de 1 sur $2^{\wedge 256}$. Les clés de chiffrement générées par le GSCUD, caractérisées par un grand aspect aléatoire, sont prévues pour un usage unique, rendant inutile toute clé potentiellement découverte par un attaquant pour le décryptage de d'autres messages. Par conséquent, l'obtention de la clé secrète d'une transmission par l'attaquant n'affecte pas les autres clés, assurant ainsi une confidentialité élevée des données.

- ✓ **Authentification et Non-Répudiation** : L'authentification, essentielle dans l'environnement d'une maison intelligente, permet de reconnaître les nœuds légitimes et assure au serveur que les données proviennent bien de la source attendue. Notre schéma attribue à chaque nœud un ID unique et secret dès l'installation, enregistré par le serveur pour tous les nœuds lors de leur configuration. Un attaquant introduisant un nœud malveillant pourrait tenter de perturber la communication, mais serait rapidement identifié par l'absence de correspondance de l'ID avec ceux des nœuds autorisés, sécurisant ainsi l'échange d'informations.

- ✓ **Intégrité** : L'intégrité est préservée en calculant et en joignant un Code d'Authentification de Message (MAC), ici obtenu par le CRC, à chaque message transmis. Le serveur, à la réception, vérifie le CRC avec celui calculé sur le message, permettant ainsi d'identifier toute modification malveillante des données.

- ✓ **Fraîcheur des Données** : La fraîcheur garantit l'actualité des données et empêche leur réutilisation malveillante. Le serveur évalue la fraîcheur de chaque message reçu en comparant le CTR actuel avec le dernier enregistré, refusant les messages dont le CTR n'a pas varié, assurant ainsi que les données sont récentes et valides.

- ✓ **Résilience** : La résilience est particulièrement importante dans les RCSF, où la compromission d'un seul nœud ne doit pas affecter la sécurité globale. Dans notre cas, seule la clé du nœud compromis est affectée.

- ✓ **Temps d'exécution :** Le tableau (IV.7) présente une comparaison entre notre algorithme et d'autres algorithmes de cryptage en termes de temps d'exécution en millisecondes. A partir des informations indiquées dans le tableau, nous déduisons que le cryptosystème proposé présente de bonnes performances en termes de temps d'exécution.

Algorithme	Taille des Paquets (Bytes)			Platforme
	16	32	512	
GSCUD(proposé)	2.8ms	5.8ms	90.4ms	Arduino UNO
GHCC(proposé)	0.8ms	1.5ms	25.6ms	Arduino DUE
Skipjack [132]	-	6.62ms	103.45ms	Sun Small Pro-grammable object
RC5 [132]	-	14.96ms	235.39ms	

MBCC [132]	-	34.02ms	533.29ms	Technology.(SunSPOT)
BCC [132]	-	220.74ms	3515.43ms	
AES-128 [133]	5.856ms	-	-	Arduino UNO
DES [133]	39.040ms	-	-	

Tableau IV.7. Temps d'exécution en millisecondes.

- ✓ **Consommation d'énergie :** La consommation d'énergie en joule est calculée en suivant la même procédure que celle utilisée dans le chapitre III. Le tableau (IV.8) donne les résultats de consommation d'énergie de notre algorithme ainsi qu'une comparaison avec d'autres algorithmes de cryptage. Les résultats représentés sur le tableau (IV.8) indique bien que le cryptosystème proposé présente de bonnes performances en termes de consommation d'énergie.

Algorithme	Taille des Paquets (Bytes)	
	32	512
GSCUD(proposé)	6.89mJ	118.4 mJ
GHCC(proposé)	2.78 mJ	39.704 mJ
Skipjack [132]	7.45 mJ	120.035 mJ
RC5 [132]	17.904 mJ	284.152 mJ
MBCC [132]	40.646 mJ	645.925 mJ
BCC [132]	267.943 mJ	4387.831 mJ

Tableau IV.8. Consommation d'énergie en milli Joule.

IV.16. Conclusion

Dans ce dernier chapitre, nous avons présenté notre deuxième contribution qui consiste en la réalisation d'un cryptosysteme chaotique pour sécuriser les données échangées dans un réseau de capteur sans fil utilisant différentes technologies de communication. D'abord nous nous sommes intéressés au système chaotique unifié discrétisé qu'on a utilisé comme générateur de clés de chiffrements. Ces clés de chiffrements que nous avons générées ont passé avec succès les tests de NIST ce qui confirme leurs bonnes propriétés statistiques. Par la suite nous avons utilisé ces clés pour chiffrer des images de références ainsi que des fichiers

audio. Nous avons ensuite effectué une analyse de sécurité des données chiffrées. Les résultats obtenus pour le calcul d'entropie, des histogrammes, de corrélation, de la sensibilité à la clé et l'espace des clés valident l'efficacité des clés de chiffrement. Après avoir rappelé le problème de la synchronisation chaotique, nous avons exposé et expliqué la solution qu'on a adoptée pour résoudre et contourner ce problème par l'utilisation d'un compteur qu'on incrémente à la fin de chaque opération de chiffrement. Par la suite nous avons décrit et détaillé la composition de la plateforme matérielle de RCSF, hétérogène, réalisée qui comprend trois nœuds et un serveur.

Ensuite, nous avons présenté, sous forme d'organigramme et d'algorithmes, les procèdes de chiffrement et de déchiffrement basés-chaos des différentes données. Puis on s'est focalisé sur la présentation et la description des différentes IHM que nous avons développées. Nous avons pu démontrer la fiabilité et la robustesse du chiffrement/déchiffrement basé chaos de tout type de données numériques transmises en temps réel à travers différents canaux de transmission sans fil. A la fin on a montré que la solution proposée répond bien à toutes les principales exigences de sécurité des données (confidentialité, authentification et non-répudiation, intégrité, résilience et fraîcheur des données) ainsi que le temps de traitement et de la consommation d'énergie.

CONCLUSION GENERALE ET PERSPECTIVES

Les travaux présentés dans cette thèse portent sur le développement de deux cryptosystemes basés chaos pour la sécurisation des réseaux de capteur sans fil dans le contexte des maisons intelligentes. Pour atteindre les objectifs tracés, dans un premier temps, il fallait chercher et proposer des générateurs de clés de chiffrement dynamiques ayant de bonnes propriétés statistiques et qui assurent un chiffrement robuste. Pour le premier cryptosysteme, le choix s'est focalisé sur l'utilisation du système hyperchaotiqus de Chen, tandis que le deuxième cryptosysteme utilise le système chaotique unifié discrétisé. Ces deux générateurs offrent l'avantage de générer des séquences binaires proches de l'aléatoire avec une extrême sensibilité à la moindre variation des valeurs de leurs paramètres et des conditions initiales. Ensuite, il fallait trouver une solution robuste pour l'épineux problème des communications basées-chaos se résumant à la synchronisation chaotique, qui se pose avec acuité pour assurer un déchiffrement correct et efficace.

Les deux cryptosysteme adoptés, nous ont permis d'obtenir des clés avec de bonnes propriétés statistiques, ce qui est confirmé par ailleurs par les tests du NIST. Pour plus de validation, nous avons effectué une analyse de sécurité (différentielle et statistique) des données chiffrées en utilisant le masque jetable à base de l'opérateur XOR. Nous avons pris des images et des fichiers audio comme données à chiffrer. Les résultats obtenus pour le calcul d'entropie, des histogrammes, de corrélation, la sensibilité à la clé et l'espace des clés valident et confirment l'efficacité des clés de chiffrement des deux cryptosystèmes.

Afin de rendre la tâche d'un éventuel attaquant plus difficile et plus ardue, nous avons intégré des fonctions de permutation qui permettent de rajouter une couche de sécurité en plus. En effet, ces fonctions assurent, entre autre, un mélange aléatoire des segments de données et la dissimilation des mots codes (servant à la synchronisation) dans les données chiffrées. Ce qui a permis d'augmenter considérablement le degré de sécurité et de robustesse des cryptosystéme proposés.

Puis nous avons présenté et détaillé l'architecture des deux plateformes matérielles développées, basées sur des RCSF (Réseaux de Capteur San Fil) ainsi que les différentes IHM

développées. Par la suite une description et une analyse des performances des algorithmes servant pour le cryptage des données, avant leurs transmissions, est effectuée.

Enfin, on a montré que les solutions proposées répondent à toutes les principales exigences de sécurité des données (confidentialité, authentification et non-répudiation, intégrité, résilience et fraîcheur des données) ainsi que le temps de traitement et de la consommation d'énergie.

A la fin nous avons effectué plusieurs tests de validation des différentes applications développées. Les tests réalisés comportaient :

- La transmission chiffrée du flux audio/vidéo temps réels entre le nœud et le serveur connecté à travers le réseau Wifi.
- Le transfert des mesures de températures et d'humidités cryptées via le Wifi, Xbee et le Bluetooth.

Les résultats expérimentaux obtenus dans cette thèse sont très satisfaisants et concluants démontrant ainsi l'efficacité de notre approche.

Comme perspectives, il est envisagé d'effectuer une cryptanalyse plus poussée des cryptosystemes proposés ainsi que leurs déploiements dans des réseaux de capteurs étendus utilisant l'Intelligence Artificielle. Aussi il serait intéressant d'utiliser les réseaux de neurones pour générer des systèmes chaotiques avec un espace de clés plus large.

BIBLIOGRAPHIE

[1]G.S.Poh, P. Gope, J.Ning, PrivHome: "Privacy-Preserving Authenticated Communication in Smart Home Environment", *IEEE Transactions on Dependable and Secure Computing*, 2019. https://doi.org/10.1109/tdsc.2019.2914911.

[2]Mansour, M et al. Internet of Things: "A Comprehensive Overview on Protocols, Architectures, Technologies, Simulation Tools, and Future Directions.", *Energies*, 16(8), pp 3465, 2023. doi: https://doi.org/10.3390/en16083465.

[3]Sudha, K.S., Jeyanthi, N. Similarity "Learning-Based Supervised Discrete Hash Signature Scheme for authentication of smart home scenario." *Int.j.inf.tecnol*, 14,3377–3389,2022. https://doi.org/10.1007/s41870-022-00920-1.

[4]Stolojescu-Crisan, C.;Crisan, C.; Butunoi, B.-P. "An IoT-Based Smart Home Automation System". *Sensors*, 21, 3784, 2021. https://doi.org/10.3390/s21113784.

[5]M.Jenal,AN.Omar,MA.Hisham,WN.Noh,ZA.Razali."Smart Home Controlling System.", *J of Electro Voltand App3(1)*:92-104, 2022.

[6]H. Ghayvat, S.. Mukhopadhyay, J. Liu, A. Babu, E. Elahi, X. Gui, "Internet of Things for smart homes and buildings: Opportunities and Challenges," *Aust. J. Telecommun. Digit. Econ.* 3, 33, 2015. doi:10.18080/ajtde.v3n4.23.

[7]CB.Kolanur,RM. Banakar,G.Rajneesh, "Design of IoT-based platform development for Smart Home Appliance Control", *Int J Phy: Conf Ser* 199(1):012052, 2021.

[8]Yasmin, N., Gupta, R. "Modified lightweight cryptography scheme and its applications in IoT environment". *Int. j. inf. tecnol.* **15**, 4403–4414, 2023. https://doi.org/10.1007/s41870-023-01486-2.

[9]Abi Sen, A.A., Eassa, F.A., Jambi, K. *et al.* "Preserving privacy in internet of things: a survey."*Int. j. inf. tecnol.* **10**,189–200, 2018. https://doi.org/10.1007/s41870-018-0113-4.

[10]F.K. Santoso, N.C.H. Vun, "Securing IoT for smart home system", *in: Int. Symp. Consum. Electron., IEEE*: pp. 1–2, 2015. doi:10.1109/ISCE.2015.7177843.

[11]M.Abu-Tair, S. Djahel, P. Perry , B. Scotney , U. Zia , J. Martinez Carracedo , A. Sajjad , "Towards Secure and Privacy-Preserving IoT Enabled Smart Home: Architecture and Experimental Study."*Sensors*, 20, 6131, 2020. doi:10.3390/s20216131.

[12]C. Wheelus, X. Zhu, "IoT Network Security: Threats, Risks, and a Data-Driven Defense Framework" . *IoT*, 1, 259–285, 2020. doi:10.3390/iot1020016

[13]Y. Liu, J. Zhang, J. Zhan, "Privacy protection for fog computing and the internet of things data based on blockchain." *Cluster Computing*, 24:1331–1345, 2021 https://doi.org/10.1007/s10586-020-03190-3.

[14]Gulganwa, P., Jain, S. EES-WCA: "energy efficient and secure weighted clustering for WSN using machine learning approach". *Int. j. inf. tecnol.* 14, 135–144, 2022. https://doi.org/10.1007/s41870-021-00744-5.

[15]H.Mrabet, S.Belguith, A.Alhomoud, A. Jemai , "A Survey of IoT Security Based on a Layered Architecture of Sensing and Data Analysis." *Sensors*, 20, 3625, 2020. doi:10.3390/s20133625.

[16]A. Ghani, K. Mansoor, S. Mehmood, S.A. Chaudhry, A.U. Rahman, M. Najmus Saqib, "Security and key management in IoT-based wireless sensor networks: An authentication protocol using symmetric key*," Int. J. Commun. Syst. 32*, e4139, 2019. doi:10.1002/dac.4139.

[17]JN .Mamvong, GL .Goteng,B. Zhou,Y. Gao, "Efficient Security Algorithm for Power

Constrained IoT Devices." *IEEE Internet of Things Journal*, 8 (7). pp. 5498-5509. ISSN 2327-4662, 2020.

[18] L. Kocarev, "Chaos-based cryptography: a brief overview," *IEEE Circuits Syst. Mag. 1* 6–21, 2001. doi:10.1109/7384.963463.

[19] Z.Rahman, X. Yi , M. Billah , M. Sumi,A. Anwar, A. "Enhancing AES Using Chaos and Logistic Map-Based Key Generation Technique for Securing IoT-Based Smart Home." *Electronics*, 11, 1083, 2022. https://doi.org/10.3390/ electronics11071083.

[20] Umar, T., Nadeem, M. & Anwer, F." Modified chaotic quadratic map with improved robust region." *Int. j. inf. Tecnol*, 2023. https://doi.org/10.1007/s41870-023-01578-z.

[21] Maurya, R., Rao, G.V.E. & Rajitha, B. "Visual cryptography for securing medical images using a combination of hyperchaotic-based pixel, bit scrambling, and DNA encoding." *Int. j. inf. tecnol.* 14, 3227–3234, 2022. https://doi.org/10.1007/s41870-022-01029-1.

[22] C. Meshram , C.C. Lee , I. Bahkali, A.L. Imoize , "An Efficient Fractional Chebyshev Chaotic Map-Based Three-Factor Session Initiation Protocol for the Human-Centered IoT Architecture." *Mathematics*, 11, 2085, 2023. https:// doi.org/10.3390/math11092085.

[23] Ons Jallouli. "Chaos-based security under real-time and energy constraints for the Internet of Things." *Thèse de doctorat en science*. Université de Nantes, 2017.

[24] YAHI Amina. " Développement d'algorithmes de cryptage d'images à base des suites chaotiques". *Thèse de doctorat en science.* Université MOHAMED EL BACHIR EL IBRAHIMI -BORDJ BOU ARRERIDJ, 2023.

[25] H.A. Abdullah, H.N. Abdullah, W.A. Mahmoud Al-Jawher, "A hybrid chaotic map for communication security applications," *Int. J. Commun. Syst*. 33, e4236, 2020. doi:10.1002/dac.4236.

[26] J.-C. Hsueh, V.H.-C. Chen, "An ultra-low voltage chaos-based true random number generator for IoT applications," *Microelectronics J.* 87, 55–64, 2019. doi:10.1016/J.MEJO.2019.03.013.

[27] S. Dhall, S.K. Pal, K. Sharma, "Cryptanalysis of image encryption scheme based on a new 1D chaotic system," *Signal Processing*. 146, 22–32, 2018. doi:10.1016/j.sigpro.2017.12.021.

[28] C. Bisdikian, "An overview of the Bluetooth wireless technology," *in IEEE Communications Magazine, vol*. 39, no. 12, pp. 86-94, Dec 2001.

[29] MACIEJ KRANZ. "building the internet of things: implement new business models, disrupt competitors, transform your industry". *1er edition.Weley*, ISBN: 978-1-119-28566-3, 2016.

[30] McEwen, Adrian, and Hakim Cassimally. " Designing the Internet of Things ".*Wiley*, , ISBN 978-1-118-43062-0, 2014.

[31] Francis Donnat, "Droit européen de l'internet: réseaux, données, services". *Paris, LGDJ*, 2018.

[32] R. R. Harmon, E. G. Castro-Leon and S. Bhide, " Smart cities and the Internet of Things," *2015 Portland International Conference on Management of Engineering and Technology (PICMET)*, Portland, OR, USA, pp. 485-494, 2015.

[33] P. Agbaje, A. Anjum, A. Mitra, E. Oseghale, G. Bloom and H. Olufowobi, "Survey of Interoperability Challenges in the Internet of Vehicles," in *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 12, pp. 22838-22861, Dec 2022.

[34] B. Di Martino, M. Rak, M. Ficco, A. Esposito, S.A. Maisto, S. Nacchia, “Internet of things reference architectures, security and interoperability: A survey,Internet of Things”, Volumes 1–2, Pages 99-112,ISSN 2542-6605, 2018.

[35] S. Singh and N. Singh, "Internet of Things (IoT): Security challenges, business opportunities & reference architecture for E-commerce,", *International Conference on Green Computing and Internet of Things (ICGCIoT)*, Greater Noida, India, pp. 1577-1581, 2015.

[36] Tawalbeh, Lo’ai, FadiMuheidat, MaisTawalbeh, and MuhannadQuwaider. "IoT Privacy and Security: Challenges and Solutions" *Applied Sciences* 10, no. 12: 4102, 2020.

[37] Miao Yun and Bu Yuxin, "Research on the architecture and key technology of Internet of Things (IoT) applied on smart grid," *2010 International Conference on Advances in Energy Engineering*, Beijing, China, pp. 69-72, 2010.

[38] ZHU, Zhen, HUANG, Ren-Gen, et al. “Study on the IoT architecture and access technology.” *In: 2017 16th International Symposium on Distributed Computing and Applications to Business, Engineering and Science (DCABES). IEEE*, p. 113-116, 2017.

[39] LAVRIC, Alexandru, PETRARIU, Adrian I., et POPA, Valentin. “Long range sigfox communication protocol scalability analysis under large-scale, high-density conditions.” *IEEE Access*, vol. 7, p. 35816-35825, 2019.

[40] STOČES, Michal, VANĚK, Jiří, MASNER, Jan, et al. “Internet of things (iot) in agriculture selected aspects.” *Agris on-line Papers in Economics and Informatics*, vol. 8, no 1, p. 83-88, 2016.

[41] L. Zhou and H. -C. Chao, "Multimedia traffic security architecture for the internet of things," *in IEEZHU, Zhen, HUANG, Ren-Gen, et al. Study on the IoT architecture and access technology*. In: 2017 *16th International Symposium on Distributed Computing and Applications to Business, Engineering and Science (DCABES). IEEE*, 2017. p. 113-116.*E Network*, vol. 25, no. 3, pp. 35-40, May-June 2011.

[42] Ibrahim Mashal, Osama Alsaryrah, Tein-Yaw Chung, Cheng-Zen Yang, Wen-HsingKuo, Dharma P. Agrawal, “Choices for interaction with things on Internet and underlying issues,Ad Hoc Networks”, Volume 28, Pages 68-90, ISSN 1570-8705, 2015.

[43] KIM, Wan-Soo, LEE, Won-Suk, et KIM, Yong-Joo. “A review of the applications of the internet of things (IoT) for agricultural automation”. Journal of Biosystems Engineering, vol. 45, p. 385-400, 2020.

[44] A. Lavric and A. I. Petrariu, "LoRaWAN communication protocol: The new era of IoT," *2018 International Conference on Development and Application Systems (DAS)*, Suceava, Romania, pp. 74-77, 2018YUEJUN, Zhang et MINGGUANG, Wu. “Design of wireless remote module in X-10 intelligent home.” *In IEEE International Conference on Industrial Technology. IEEE*, p. 1349-1353, 2005.

[45] LIANG, Lulu, ZHENG, Kai, SHENG, Qiankun, et al. “A denial of service attack method for an iot system.” *In 8th international conference on Information Technology in Medicine and Education (ITME). IEEE*, p. 360-364, 2016.

[46] AL-KASHOASH, Hayder AA et KEMP, Andrew H. “Comparison of 6LoWPAN and LPWAN for the Internet of Things”. *Australian Journal of Electrical and Electronics Engineering*, vol. 13, no 4, p. 268-274, 2016.

[47] SOBIN, C. C. “A survey on architecture, protocols and challenges in IoT.” *Wireless Personal Communications*, vol. 112, no 3, p. 1383-1429, 2020.

[48] YASSEIN, MuneerBani, MARDINI, Wail, et KHALIL, Ashwaq. " Smart homes automation using Z-wave protocol ".*In International Conference on Engineering & MIS (ICEMIS), IEEE*, p. 1-6 , 2016.

[49] MERABET, Fatma. "Solutions de sécurité pour l'internet des objets dans le cadre de l'assistance à l'autonomie à domicile". *Thèse de doctorat. Université de Limoges; Université Mouloud Mammeri* (Tizi-Ouzou, Algérie), 2021.

[50] ZHU, Quanyan, SAAD, Walid, HAN, Zhu, et al. "Eavesdropping and jamming in next-generation wireless networks: A game-theoretic approach." *In MILCOM 2011 Military Communications Conference. IEEE*, p. 119-124, 2011.

[51] DUTTA, Nishigandhaet SINGH, MoirangthemMarjit. "Wormhole attack in wireless sensor networks: a critical review." *Advanced Computing and Communication Technologies: Proceedings of the 11th ICACCT*, p. 147-161, 2018.

[52] PUNDIR, Sumit, WAZID, Mohammad, SINGH, DeveshPratap, et al. "Designing efficient sinkhole attack detection mechanism in edge-based IoT deployment". *Sensors*, vol. 20, no 5, p. 1300, 2020.

[53] SHOBANA, M. et RATHI, S. "Iot malware: An analysis of iot device hijacking." *Int. J. Sci. Res. Comput. Sci. Eng. Inf. Technol*, vol. 5, p. 2456-3307, 2018.

[54] CHEN, Qifeng, CHEN, Haoming, CAI, Yanpu, et al. "Denial of service attack on IoT system." *In 9th international conference on information technology in medicine and education (ITME), IEEE*, p. 755-758, 2018.

[55] NASSIRI ABRISHAMCHI, Mohammad Ali et ZAINAL, Anazida. "Review of Smart Home Privacy-Protecting Strategies from a Wireless Eavesdropping Attack." *In : The International Conference on Data Science and Emerging Technologies, Singapore : Springer Nature Singapore*, p. 154-167, 2022.

[56] CERVANTES, Christian, POPLADE, Diego, NOGUEIRA, Michele, et al. "Detection of sinkhole attacks for supporting secure routing on 6LoWPAN for Internet of Things." *In: 2015 IFIP/IEEE International Symposium on Integrated Network Management (IM), IEEE*, p. 606-611, 2015.

[57] HU, Yih-Chun, PERRIG, Adrian, et JOHNSON, David B. "Wormhole attacks in wireless networks". *IEEE journal on selected areas in communications*, vol. 24, no 2, p. 370-380, 2006.

[58] SYLLA, Tidiane. "Sécurité et vie privée centrées sur l'utilisateur dans l'IoT. " *Thèse de doctorat. Bordeaux*, 2021.

[59] EL-HAJJ, Mohammed, FADLALLAH, Ahmad, CHAMOUN, Maroun, et al. "A survey of internet of things (IoT) authentication schemes".*Sensors*, vol. 19, no 5, p. 1141, 2019.

[60] NAJMI, Kholoud Y., ALZAIN, Mohammed A., MASUD, Mehedi, et al. "A survey on security threats and countermeasures in IoT to achieve users confidentiality and reliability." *Materials Today : Proceedings*, vol. 81, p. 377-382, 2023.

[61] SONI, Dipaet MAKWANA, Ashwin. "A survey on mqtt: a protocol of internet of things (iot)." *In: International conference on telecommunication, power analysis and computing techniques (ICTPACT-2017)*, p. 173-177, 2017.

[62] RAHMAN, Reem Abdul et SHAH, Babar. "Security analysis of IoT protocols: A focus in CoAP". *In: 3rd MEC international conference on big data and smart city (ICBDSC). IEEE*, p. 1-7, 2016.

[63] ANSARI, Danish Bilal, REHMAN, Atteeq-Ur, et ALI, Rizwan. “Internet of things (iot) protocols: a brief exploration of mqtt and coap.” *International Journal of Computer Applications*, vol. 179, no 27, p. 9-14, 2018.

[64] YAKUPOV, Dmitry. “Overview and comparison of protocols Internet of Things: MQTT and AMQP”. *International Journal of Open Information Technologies*, vol. 10, no 9, p. 90-98, 2022.

[65] MISHRA, Biswajeebanet KERTESZ, Attila. “The use of MQTT in M2M and IoT systems: A survey”. *IEEE Access*, vol. 8, p. 201071-201086, 2020.

[66] CONZON, Davide, BOLOGNESI, Thomas, BRIZZI, Paolo, et al. “The VIRTUS middleware: An XMPP based architecture for secure IoT communications.” *In: 21st International Conference on Computer Communications and Networks (ICCCN), IEEE*, p. 1-6, 2012.

[67] MALIK, Muhammad Imran, MCATEER, Ian Noel, HANNAY, Peter, et al. “XMPP architecture and security challenges in an IoT” *ecosystem.In proceeding of the 16th Australlian Information Security Management Conference* (pp. 62-73), 2018.

[69] S.Sadoudi, “Système de Communication Chaotique Embarqué pour la Sécurisation de la Transmission de Données”, *Thèse de doctorat*, Ecole Militaire Polytechnique, 2013.

[70] James Gleick, “La théorie du chaos — vers une nouvelle science,” *Flammarion*, 1991.

[71] E. Cherrier. “ Estimation de l'état et des entrées inconnues pour une classe de systèmes non linéaires”, *Thèse de doctorat*, 2006.

[72] T. E. N. Lorenz,“ Deterministic non periodic flow,” *Journal of the Atmospheric Sciences*, Vol.20, No. 2, pp.130-141, March 1963.

[73] T.Y. Li, J. A. Yorke, “Period Three Implies Chaos”, *The American Mathematical Monthly*, Vol. 82, No.10. pp.985-992, December 1975.

[74] L. O. Chua, “Chua’s circuit: Ten years later,” *IEICE Trans. Fundamentals*, Vol. E77-A, pp. 1811-1822, November 1994.

[75] E. Ott,C. Grebogi and J.A. Yorke “Controling chaos”, *Physical Review Letters*,64(11),1196-1199, 1990.

[76] K. M. Cuomo and A. V. Oppenheim, “Circuit implementation of synchronized chaos with applications to communications,” *Phy. Rev*. Lett., 71, pp. 65-68, 1993.

[77] G. D. Van Wiggeren and R. Roy,“Communication with chaotic lasers,” *Science*, 279,1198–1200, February 1998.

[78] A. Panas., B. Kyarginsky, N. Maksimov, “Single-transistor microwave chaotic oscillator,” *In Proc. of NOLTA'2000*, Dresden, Germany, pp. 445- 448, 2000.

[79] A. Dmitriev, M. Hasler, A.I. Panas, K.V. Zakharchenko, “Basic Principles Of Direct Chaotic Communications,” *Nonlinear Phenomena in Complex Systems*, Vol. 4, No. 1, 2002.

[80] Tao Yang, Chai Wah Wu and Leon O. Chua, “Cryptography based on chaotic systems”, *IEEE Transaction on Circuits and Systems—I: fundamental theory and applications*, Vol.44, No.5, pp.469-472, May 1997.

[81] J. Bricmont, “ Introduction à la dynamique non linéaire”, *PHYS 2111*, 2010.

[82] M.S. Azzaz, “Implantation paramétrable d’un nouvel algorithme de cryptage symétrique basé chaos par inclusion au sein d’une architecture reconfigurable de type FPGA”, *Thèse de doctorat en science*, Ecole Militaire Polytechnique, Alger, 2012.

[83] E. Ghys, "L'attracteur de Lorenz, paradigme du chaos", *Seminaire Poincaré 14*, pp.1-52, 2010.

[84] P. Gaspard, "Rossler Systems", *Encyclopedia of Nonlinear Science, Alwyn Scott Editor*, pp. 808-811, New York, 2005.

[85] P. Wu, "Rechearch on Digital Image Watermark Encryption Based on Hyperchaotic", *Thèse doctorat en science*, University of Derby, May 2012.

[86] M. J. Rostami, et al « A novel parallel image encryption with chaotic windows based on logistic map », *Comput. Electr. Eng.*, vol. 62, p. 384‑400, août 2017, doi: 10.1016/j.compeleceng.(2017).04.004.

[87] H. A. Abdullah, H. N. Abdullah, et W. A. Mahmoud Al-Jawher, « A hybrid chaotic map for communication security applications », *Int. J. Commun. Syst.*, vol. 33, no4, p. e4236, mars 2020, doi: 10.1002/dac.4236.

[88] F. Anstett, "Les Systèmes Dynamiques Chaotiques pour le Chiffrement: Synthèse et Cryptanalyse", *Thèse doctorat en science*, Université Henri Poincaré, Nancy, 2006.

[89] H.D. Abarbanel, R.Brown, and M.Kennel, " Lyapunov exponents in chaotic systems: their importance and their evaluation using observed data," *International Journal of Modern Physics B*, vol. 5, no. 09, pp. 1347–1375, 1991.

[90] David Louapre. Théorie du chaos et effet papillon. science étonnante, 2018.

[91] H.NOURA, "Conception et simulation des generateurs,crypto-systèmes et fonctions de hachages basés chaos performants ", *Thèse doctorat en science*, Université de Nantes, 2012.

[92] A. Menezes, P. VanOorschot, S. Vanstone, "Handbook of applied cryptography", *by CRCPress*, 1997.

[93] C. E. Shannon, "Communication Theory of Secrecy Systems", *Bel System Technical Journa*l,Vol.28, No.4, pp.656-715, 1949.

[94] W. Diffie et M. E. Hellman,"New directions in cryptography." *IEEE Transac-tions on Information Theory*, IT-22(6):644-654, 1976.

[95] Gilbert S. Vernam,"Cipher printing telegraph systems for secret wire and radio telegraphic communications." *IEEE Transactions of the American Institute of Electrical Engineers*, 55:109-115, 1926.

[96] M. Bensalah, "Protocoles RFID Pour L'Authentification Sur Les Courbes Elliptiques", *Thèse doctorat* , Ecole Militaire Polytechnique, Alger, 2014.

[97] T. Yang, "A survey of chaotic secure communication systems,"*Int. J. of Computational Cognition*, Vol. 2, No. 2, pp. 81–130, June 2004.

[98] G. Vidal,a, M.S. Baptista, and H. Mancini, "A fast and light stream cipher for smartphones", *Eur. Phys. J. Special Topics*, e2014-02185-y, 2014.

[99] G. Alvarez and S. Li, "Some basic cryptographic requirements for chaos-based cryptosystems," *Int. J. of Bifurcation and Chaos*, 16(8), pp. 2129-2151, 2006.

[100] FIPS 140-2, "Announcing the standard for Security Requirement for Cryptographic Modules", *U.S. Departement of commerce*," 2002.

[101] G. Marsaglia, "Diehard: a battery of tests of randomness," http://stat.fsu.edu/ geo/diehard.html, 1996.

[102] Anwendungshinweise und Interpretation zum Schema (AIS), 2011.

[103] A. Rukhin1, J. Soto, J. Nechvatal, M. Smid, E. Barker, S. Leigh, M. Levenson, M. Vangel, D. Banks, A. Heckert, J. Dray and S. Vo, "A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications," *NIST Special Publication 800-22 Revision 1a*, April 2010.

[104] Eli Biham and Adi Shamir. "Differential cryptanalysis of DES-like cryptosystems". *Journal of CRYPTOLOGY*, 4(1):3–72, 1991.

[105] Yue Wu, Joseph P Noonan, and Sos Agaian. " NPCR and UACI randomness testsfor image encryption ". *Cyber Journals: Multidisciplinary Journals in Science andTechnology, Journal of Selected Areas in Telecommunications (JSAT)*, pages 31–38, 2011.

[106] M.Farajallah, " Chaos-based crypto and joint crypto-compression systems for images and videos", *Thèse doctorat en science*, Université de Nantes, 2015.

[107] Xingyuan Wang, Dapeng Luan, and Xuemei Bao. "Cryptanalysis of an image encryption algorithm using chebyshev generator". *Digital Signal Processing*, 25:244–247, 2014.

[108] M. R. Sabuncu, « Entropy-based Image Registration », *PhD Thesis*, Université de Princeton, USA, 2006.

[109] J. Wang, L.Liu. « A Novel Chaos-Based Image Encryption Using Magic Square Scrambling and Octree Diffusing». *Mathematics*, 10, 457, 2022, https://doi.org/10.3390/math10030457.

[110] Jolfaei, Alireza, « Robust Encryption Schemes for 3D Content Protection», 2016, doi: 10.25904/1912/3357.

[111] M. Sonka, V. Hlavac, R. Boyle, « Image processing, analysis, and machine vision», *Fourth edition. Stamford, CT, USA: Cengage Learning*, 2015.

[112] K. L. Wuensch. Chi-square tests. In International Encyclopedia of Statistical Science, pages 252–253. Springer, 2011.

[113] S.H. Chen and J. Lü, "Synchronization of an uncertain unified system via adaptive control", *Chaos, Solitons and Fractals*, 14, pp. 643–647, 2002.

[114] [RFC 1321, 1992] R. Rivest, "The MD5 message-digest algorithm", *IETF Network W orking Group*, RFC 1321, 1992.

[115] [RFC 3174, 2001] RFC 3174, US, "Secure Hash Algorithm 1", http://www.faqs.org/rfcs/rfc3174.html.

[116] Zhu, S.; Deng, X.; Zhang,W.; Zhu, C. "Image Encryption Scheme Based on Newly Designed Chaotic Map and Parallel DNA Coding". *Mathematics*, 11, 231, 2023, https://doi.org/10.3390/math11010231.

[117] Moya-Albor, E.;Romero-Arellano, A.; Brieva, J.;Gomez-Coronel, S.L. "Color Image Encryption Algorithm Based on a Chaotic Model Using the Modular Discrete Derivative and Langton's Ant". *Mathematics*,11, 2396, 2023, https://doi.org/10.3390/math11102396.

[118] Wajih El hadj Youssef et al. "A Secure Chaos-based Lightweight Cryptosystem for the Internet of Things". *IEEE Access*, 2023 . doi: 10.1109/ACCESS.2023.3326476.

[119] W. Zhou, X. Wang, M. Wang, and D. Li, "A New Combination Chaotic System and its Application in a New Bit-level Image Encryption Scheme," *Opt. Lasers Eng*, vol. 149, Art. no. 106782, 2022.

[120] L. Zhu, D. Jiang, J. Ni, X. Wang, X. Rong, M. Ahmad, andY. Chen, "A Stable Meaningful Image Encryption Scheme using the Newly-designed 2D Discrete Fractional-Order Chaotic Map and Bayesian Compressive Sensing", *Signal Process*, vol. 195, Art. no. 108489, 2022.

[121] X. Wang and M. Zhang, "An Image Encryption Algorithm based on New Chaos and Diffusion Values of a Truth Table", *Inf. Sci*, vol. 579, pp. 128–149, 2021.

[122] P.L. Carroll and L.M. Pecora , " Synchronization in chaotic systems " *Phy.Rev.Lett*.,Vol.64,NO.8,pp.821-824, 1990.

[123] Guy Pujoll . "Les Réseaux"*EYROLLES* , Edition 2011.

[124] F. Almasalha , R.H. Beltran and A.A. Khokhar " Partial Encryption of Entropy-Coded Video Compression Using Coupled Chaotic Maps ". *Entropy*, 16, 5575-5600, 2014.

[125] A. Mosenia, N.K. Jha, "A Comprehensive Study of Security of Internet-of-Things", *IEEE Trans. Emerg.Top. Comput*. 5 586–602, 2017, doi:10.1109/TETC.2016.2606384.

[126] Q. Jing, A. V. Vasilakos, J. Wan, J. Lu, D. Qiu, "Security of the Internet of Things: perspectives and challenges", *Wirel.Networks*. 20 2481–2501, 2014, doi:10.1007/s11276-014-0761-7.

[127] Le site officiel de Microsoft : http://msdn.microsoft.com.

[128] El-Razouk, H., Reyhani-Masoleh, A., & Gong, G. " New Implementations of the WG Stream Cipher*". IEEE Transactions on Very Large Scale Integration (VLSI) Systems* V.22 N.9, pp. 1865-1878, 2014.

[129] Fontaine, C. " Synchronous Stream Cipher". *Encyclopedia of Cryptography and Security*, 1274-1275, 2011.

[130] Preneel, B. " An introduction to modern cryptology". *Dept. Electrical Engineering-ESAT/COSIC*, 2007.

[131] Borisov,N.;Goldberg,I.;Wagner,D."Intercepting mobile communications:The insecurity of 802.11". *In Proceedings of the 7th Annual International Conferenceon Mobile Computing and Networking (MOBICOM)*,Rome,Italy,16–21, pp.180–189, July 2011.

[132] M. Sharafi et al.:" Low Power Cryptography Solution Based on Chaos Theory in Wireless Sensor Nodes". *IEEE Access*, vol.7, 2019.

[133] A. Sukiatmodjo, Y. Dwi Setianto , "Speed and Power Consumption Comparison between DES and AES Algorithm in Arduino". *Scientific Journal of Informatics*, Vol.6, 2019.

[134] S. Kehui : "Secure Communication : Principles and Technologies", China, 2016.

[135] A. Vanecek, S. Celikovsky : "Control Systems : From Linear Analysis to Synthesis of Chaos. London, Petrice-Hall, 1996.

[136] Rui Miguel Silva, Rui Gustavo Crespo, Mario SerafimNunes, "Enhanced chaotic stream cipher for WSNs", *IEEE International Conference on Availability, Reliability and Security*, p 210-215, February 2010.

[137] B. Schneier, " Cryptographie appliquée : Algorithmes, protocoles et codes source en C". *Vuibert. 2emeedition*, 2001.

[138] Le site officiel de l'UIT-T : http://www. UIT-T. com.

[139] Miguel Garcia-Bosque et al "Application of a MEMS-Based TRNG in a Chaotic Stream Cipher". *Sensors*, 17, 646, 2017, doi:10.3390/s17030646.

[140] https://www.udoo.org.
[141] https://fr.digi.com/blog/post/iot-solutions-for-transportation.
[142] https://www.actu-environnement.com/ae/news/smart-city-ville-intelligente-data-donnees-transport-energie-28845.php4.
[143] https://www.veyan.fr/securite-des-objets-connectes-dans-le-domaine-de-la-sante/.
[144] https://greenerideal.com/guides/smart-home-101-how-to-get-started-with-home-automation/.

Publication

Samir Benzegane, Mounir Bouhedda, Noureddine Batel “ Hyperchaos-based cryptosytem to secure WSN data transfer for smart home application ” Int. j. inf. tecnol.2024. https://doi.org/10.1007/s41870-024-01787-0.