Sécurisation par dynamiques chaotiques des réseaux locaux sans fil au niveau de la couche MAC

Ghada Zaïbi

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par :

Université Toulouse 2 Le Mirail (UT2 Le Mirail)

Cotutelle internationale Ecole Nationale d'ingénieur de Sfax, Université de Sfax TUNISIE

Présentée et soutenue par :
Ghada ZAÏBI
le jeudi 6 décembre 2012

Titre :

Sécurisation par dynamiques chaotiques des réseaux locaux sans fil au niveau de la couche MAC

École doctorale et discipline ou spécialité :
EDSYS : Systèmes embarqués 4200046

Unité de recherche :
IRIT - Institut de Recherche en Informatique de Toulouse - UMR 5505

Directeur(s) de Thèse :

M. PEYRARD Fabrice
Mme FOURNIER-PRUNARET Danièle

Jury (noms, prénoms et qualité des membres) :

M. Abdennaceur KACHOURI (Professeur, ISSIG-Gabès), Président
M. Safwan EL ASSAD (Maître de conférences - HDR, Polytech'Nantes), Rapporteur
M. Kamel BESBES (Professeur, FSc-Monastir), Rapporteur
Mme. Danièle FOURNIER-PRUNARET (Professeur, INSA-Toulouse), Co-directrice de Thèse
M. Mounir SAMET (Professeur, ENI-Sfax), Directeur de Thèse
M. Fabrice PEYRARD (Maître de conférences - HDR, Université Toulouse II), Directeur de Thèse

République Tunisienne
Ministère de l'Enseignement Supérieur,
de la Recherche Scientifique

Université de Sfax
École Nationale d'Ingénieurs de Sfax

Ecole Doctorale
Sciences et Technologies

Thèse de *DOCTORAT*

Génie électrique

N° d'ordre: 246– 2012

THESE

Présentée à

L'École Nationale d'Ingénieurs de Sfax

En vue de l'obtention du

DOCTORAT

Dans la discipline *Génie électrique*

Par

Ghada ZAIBI

(Assistante en Génie électrique à l'ENIM)

Sécurisation par dynamiques chaotiques des réseaux locaux sans fil au niveau de la couche MAC

Soutenue le 6 Décembre 2012, devant le jury composé de :

M.	**Abdennaceur KACHOURI** (Professeur, ISSIG-Gabès)	Président
M.	**Safwan EL ASSAD** (Maître de conférences-HDR, Polytech'Nantes)	Rapporteur
M.	**Kamel BESBES** (Professeur, FSc-Monastir)	Rapporteur
Mme.	**Danièle FOURNIER-PRUNARET** (Professeur, INSA-Toulouse)	Co-Directrice de Thèse
M.	**Mounir SAMET** (Professeur, ENI-Sfax)	Directeur de Thèse
M.	**Fabrice PEYRARD** (Maître de conférences-HDR, U.Toulouse II)	Directeur de Thèse

À mon père Mohamed

À ma mère Radhia

À mes frères Amine & Oussama

Remerciements

Je me disais toujours qu'allais-je écrire dans le remerciement, ça me souciait parfois, et je me disais que je vais dire telle ou telle chose, que je ne vais pas oublier ce moment ou cet obstacle. Mais les heures passent, les heures deviennent des jours et les jours des mois et les mois des années….Et le jour où on me dit : Voilà, tu dois tout rédiger, on est pressé par le temps, je rentre dans ce cercle vicieux de stress, tête vide, sècheresse de mots et d'esprit. Je découvre à quel point cette tâche me parait difficile aussi bien qu'elle était facile au moment où tout à débuter.

Cette thèse s'inscrit dans le cadre d'une thèse en cotutelle entre l'Ecole Nationale d'Ingénieur de Sfax relevant de l'Université de Sfax et l'Université de Toulouse II le Mirail.

J'ai effectué les travaux de recherche au sein du Laboratoire d'Electronique et des Technologies de l'Information (LETI) en coopération avec le Laboratoire Toulousain de Technologie et d'Ingénierie des Systèmes de Toulouse (LATTIS) à l'INSA de Toulouse.

Je remercie vivement Monsieur Kamel Besbes, Professeur à la faculté des sciences de Monastir, et Monsieur Safwan el Assad, Maître de conférence à la Polytech'Nantes, d'avoir acceptés de rapporter ce travail.

J'exprime également mes sincères remerciements et ma gratitude à Madame Danièle Fournier-Prunaret, Professeur à l'INSA de Toulouse, et à Monsieur Fabrice Peyrard, Maître de conférences à l'université Toulouse II, pour leurs qualités pédagogiques et humaines, pour leurs aides, leurs conseils, leur encouragement et leur disponibilité durant ma thèse.

Je souhaite remercier vivement Monsieur Mounir Samet, Professeur à l'ENIS-Sfax, et Monsieur Abdennaceur Kachouri, Professeur à l'ISSIG-Gabès, pour leurs qualités pédagogiques et humaines et pour leur confiance qui m'ont accordé. Tout au long de mon parcours, depuis le PFE jusqu'à la thèse, vous étiez toujours là à me supporter, à m'encourager et à me montrer le bon chemin.

A mes chers encadrants, vous m'avez appris, vous m'avez porté assistance, vous m'avez orienté, je vous dis merci.

Durant mes séjours à l'INSA j'ai rencontré plusieurs personnes sympathiques avec qui j'ai partagé de bons moments : Côme, Hironori, Mohamed Lamine, Marc, Yanjun, Xavier et Xinwei. Je n'oublie pas Jihane et Karim, je voyais en vous le Maghreb United dans toute sa différence et sa richesse de culture. Merci également à Joëlle, Karima et Taha pour leur disponibilité.

Merci à Nada Rabhi ma chère amie et mon alliée des combats cybernétiques contre l'obscurantisme et les visions anéantissant notre société moderne. Je garde nos beaux souvenirs à Toulouse et en Tunisie dans l'espoir de les revivre dans les nouveaux projets à entreprendre.

Mes amis à l'ENIS et au LETI je ne vous oublie pas : Abir, Amira, Asma, Dalenda, Dorra, Emna, Fatma, Jihène. B (Djo), Jihène. M, Moncef, Nawel, Nejah, Olfa (Welfa), Salwa et Wafa, on compte sur vous pour la réforme et l'émancipation des universités tunisiennes.

Kaouther mon amie à l'ENIM, tu m'as rendue l'ENIM agréable par ta présence et ton support moral. Si tu es en train de lire ces mots, alors je sais bien que je suis enfin en vacances.

Je remercie infiniment ma famille, mes parents et mes frères à qui je dois tout ce que je suis aujourd'hui. Sans vous, je n'irai nulle part dans ce monde cruel. Dans mes moments de joie vous étiez là et dans les moments où je pouvais plus continuer, où je menaçais de jeter l'éponge, je me relève à chaque fois grâce à vous et pour vous. Amin, merci pour les discussions à plusieurs sujets politiques, religion, littérature et art, ça commençait toujours par une question qui se répète plusieurs fois pour m'en sortir de mon silence de mort, de mon

détachement du monde réel (parfois je naviguais dans le vide) et finissait par mon fameux «je dois revenir à mon travail ». Merci pour être un vrai frère qui me supporte à fond malgré tous mes défauts et mes cruautés. Oussama, nos parties de Cossack qui parfois finissait par des querelles parce que je n'aime pas les critiques du coach. C'est inné, c'est ma nature, j'aime la démocratie mais j'exerce la dictature, comme tout ce peuple que je peux considérer schizophrène comme moi d'ailleurs. Merci pour être mon petit frère, le plus parfait de tous les enfants et le plus gentils de tous les petits frères.

Maman, je te dois mon existence et ma réussite, est ce qu'il est temps que je grandisse dans tes yeux? Je serai pour toujours la petite Ghada qui a besoin de conseils et d'amour.

Papa, toi mon encadrant qui ne figure pas dans les membres de jury, mon guide et mon professeur. Tes sacrifices et tes efforts me sont chers, cette thèse est pour toi.

Je ne veux pas que ce remerciement soit une épopée, je vous souhaite alors une bonne lecture de cet humble manuscrit.

« Le chaos est souvent source de vie alors que l'ordre génère les habitudes. »

Henry Brooks Adam

Table des matières

Table des figures

Liste des tableaux

Abréviations

ACL	Access Control List
ADC	Analog to Digital Converter
AES	Advanced Encryption Standard
ARIB	Association of Radio Industries and Businesses
BER	Bit Error Rate
CBC-MAC	Cipher Block Chaining-MAC
CCS-PRBG	Couple Chaotic Systems Based PRBG
CRC	Cyclic Redundancy Check
CSK	Chaotic Shift Keying
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance
CTR	Counter Mode
CTR CBC-MAC (CCM)	Counter - Cipher Block Chaining-MAC
DCSK	Differential Chaotic Shift Keying
DES	Data Encryption Standard
DNSK	Differential Noise Shift Keying
ECKBA	Enhanced Chaotic Key-Based Algorithm
Erfc	Complementary error function
ETSI	European Telecommunication Standard Institute
GPS	Global Positioning System
HiperLAN	High Performance Local Area Network
HisWAN	High-Speed Wireless Area Network
IEEE	Institute of Electrical and Electronic Engineering
Igamc	Incomplete gamma function
ISM	Industrial, Scientific and Medical
IV	Initial Vector
JBREWS	Joint Biological Remote Early Warning System
LCG	Linear Congruential Generator

LFSR	Linear Feedback Shift Register
LP	Linear Probability
LR-WPAN	Low Rate-Wireless Personal Area Network
MAC	Medium Access Channel
MCU	Microcontroller unit/multichip unit
MIC	message integrity check
MIPS	Million Instructions Per Seconds
MITM	Meet-In-The-Middle
MMAC	Multimedia Mobile Access Communication Systems
MSP	Mixed Signal Processor
NIST	National Institute of Standards and Technology
NPCR	Number of pixels change rate
NSA	National Security Agency
OFDM	Orthogonal Frequency Division Multiplexing
OSI	Open Systems Interconnection
P-Box	Permutation-Box
PDA	Personal Digital Assistant
PIB	Pan Information Base
PNG	Portable Network Graphics
PRBG	Pseudorandom Bit Generator
PRNG	Pseudo Random Number Generator
PWLCM	PieceWise Linear Chaotic Map
RAM	Random Access Memory
RC4	Rivest Cipher 4
RNG	Random Number Generator
ROM	Read Only Memory
RSA	Rivest, Shamir and Adleman

SAC	Strict Avalanche Criterion
S-Box	Substitution Box
SNR	Signal to Noise Ratio
SP	Substitution Permutation
Ti	Texas instruments
UACI	Unified average changing intensity
U-NII	Unlicensed National Information Infrastructure
WATS	Wide Area Tracking System
Wifi	Wireless Fidelity
WLAN	Wireless Local Area Network
WMAN	Wireless Metropolitan Area Network
WPAN	Wireless Personal Area Network
WSN	Wireless Sensor Network
XOR	Exclusive OR
XSL	eXtended Sparse Linearization

Introduction Générale

Dans sa quête éternelle de la vérité, l'homme n'a pas cessé un jour de développer et déployer les moyens de communication et de transmission de l'information. L'évolution croissante des technologies de l'information validée par la loi de Moore a conduit à l'apparition des réseaux sans fil, comme étant un moyen reflétant et obéissant à l'évolution des sociétés modernes. Le rôle que jouent les réseaux sans fil dans la vie quotidienne a dépassé les attentes et les prévisions, pour être un élément majeur dans les révolutions populaires du nouveau millénaire.

La guerre non déclarée entre les gouvernements et les chevaliers du net incarnait essentiellement l'un de ses pionniers : La sécurité des informations propagées sur le réseau, bien évidement leur intégrité, leur authenticité et leur confidentialité. Toutefois, ce besoin de protéger et camoufler les données remontait à bien plus longtemps, avant même l'apparition des calculatrices et des premiers ordinateurs personnels, donnant naissance à l'art d'écrire en langage codé : la cryptographie. Elle débutait, en fait, à l'antiquité avec les grecs ; d'où l'origine de son appellation « Kryptos-Graphein» ou « cacher-écrire» et avait évolué depuis, en basculant entre deux états : la robustesse des communications des données et la faiblesse traduite par la cryptanalyse. Elle permet d'assurer la sécurité des schémas de chiffrement pour éviter que l'adversaire ne réussisse à déchiffrer les données.

La cryptologie, par ces deux axes cryptographie et cryptanalyse, a pris son essor durant les deux guerres mondiales où les exploits des cryptologues ont orienté le chemin de l'histoire.

A la fin de la deuxième guerre mondiale, Shannon donna des preuves théoriques sur l'impossibilité du déchiffrement du chiffrement de Vernam ou « one time pad » mais avec des grandes difficultés de mise en pratique jusqu'à aujourd'hui. Il proposa aussi des caractéristiques à vérifier pour un bon cryptage. Dans un bon mélange de transformations, toute petite variation des variables des fonctions compliquées du système change les sorties d'une manière considérable. Un autre principe très important énoncé par Kerckhoffs indique que la sécurité d'un système de chiffrement ne doit pas résider dans le procédé de chiffrement mais plutôt dans la clé. Concevoir des systèmes cryptographiques sûrs devait satisfaire ces critères.

En se référant à la sensibilité aux variations des variables du système de chiffrement, un axe récent de la cryptographie moderne s'annonce très prometteur et défiant les algorithmes classiques existants sur la scène cryptographique : c'est le chiffrement par chaos. Les signaux chaotiques ont des caractéristiques ressemblant au bruit, ils sont généralement à large bande et il est difficile de prévoir leur évolution à long terme ; ils sont alors utiles pour les communications à étalement de spectre. En particulier, puisque les signaux chaotiques ressemblent au bruit et sont difficiles à être identifiés, il est possible de les utiliser dans des situations où on désire avoir une faible probabilité d'interception.

Les systèmes de communications sans fil sont exposés à différentes formes de distorsions et d'attaques pouvant perturber la transmission, la synchronisation des données et aussi la localisation des nœuds. Notre étude effectuée concerne des applications où la sécurité et la rapidité de transfert de données sont indispensables (payement par carte bancaire, commande de processus à distance, communication militaire, ..), ainsi qu'une limitation des ressources

exigées par la technologie. La conception d'un système de communication sécurisée par dynamique chaotique, ayant une robustesse contre les altérations du canal (offrant une bonne qualité de service) et économique en ressources allouées est le but de notre recherche.

La sécurité au niveau couche MAC des réseaux sans fil (WPAN, WLAN, WSN) à l'aide des signaux chaotiques est une approche novatrice qui peut bénéficier des caractéristiques des dynamiques chaotiques surtout que le développement technologique, nécessite que l'implémentation doit se faire sur des composants de petite taille (PDA, nœuds terminaux des réseaux sans fil, systèmes embarqués), avec des débits plus faibles, une faible consommation électrique et à bas prix. Toutefois, la mise en pratique des cryptosystèmes chaotiques reste un domaine peu exploré où les recherches se limitent la plupart du temps à la simulation. La validation pratique de l'ensemble de nos travaux sur une plateforme réelle de réseaux de capteurs étendus est l'un des défis que nous avons réussi à emporter.

Nous allons décortiquer dans ce qui suit le plan de la thèse où nous avons aborder la cryptographie par chaos en liaison avec la cryptographie classique et son niveau de sécurité ainsi que la possibilité d'implémentation sur un réseau de capteurs sans fil.

Plan de la thèse

Cette thèse s'articule autour de quatre chapitres. Les deux premiers chapitres représentent l'état de l'art des réseaux sans fil, la cryptographie traditionnelle et la théorie du chaos. Le troisième et le quatrième chapitres se concentrent sur nos contributions pour la sécurité par chaos et leur application sur un réseau de capteurs sans fil.

Le premier chapitre : Dans la première partie de ce chapitre, il était indispensable de présenter les réseaux locaux sans fil et en particulier les réseaux de capteurs sans fil. Les caractéristiques et les contraintes matérielles des nœuds capteurs sont mises en exergue puisqu'elles font partie des problématiques à prendre en compte dans la suite des travaux.
La deuxième partie recense les notions de base de la cryptologie et les algorithmes de chiffrement classique en liaison avec la sécurité des réseaux locaux sans fil. Une attention particulière est accordée aux algorithmes retenus par le NIST tel que l'algorithme AES.

Le deuxième chapitre : Ce chapitre introduit la théorie du chaos. Nous analysons, en effet, les caractéristiques d'un signal chaotique et nous présentons quelques exemples d'attracteurs chaotiques célèbres. Pour étudier l'aspect aléatoire des récurrences chaotiques, une série de tests statistiques, appelés tests du NIST, est appliquée. Nous démontrons que le choix de la représentation binaire des sorties des récurrences chaotiques influence leur uniformité, leur extensibilité et leur cohérence.

Le troisième chapitre : Nous nous concentrons dans ce chapitre sur les cryptosystèmes chaotiques tout en donnant un aperçu sur les cryptosystèmes chaotiques dédiés aux réseaux de capteurs sans fil. Nous présentons et analysons aussi nos contributions pour la construction des S-Box dynamiques chaotiques. Une comparaison des performances de nos algorithmes proposés par rapport à la littérature est réalisée. En effet, nous effectuons une analyse de sécurité en étudiant les critères essentiels de construction des S-Box chaotiques. Ils comprennent le critère d'avalanche stricte, la probabilité d'approximation linéaire et la distribution équiprobable des différentielles d'entrée/sortie. Ces critères permettent de vérifier les propriétés de diffusion et de confusion des algorithmes proposés et de mesurer leur résistance aux cryptanalyses linéaire et différentielle.

Le quatrième chapitre : Ce dernier chapitre se décompose en deux sections.

La première section concerne l'étude de différents algorithmes à base de récurrences chaotiques pour le chiffrement d'images et la proposition d'un nouveau cryptosystème chaotique dédié à la sécurité des réseaux de capteurs sans fil. Nous examinons la robustesse de notre cryptosystème à l'aide d'un ensemble de métrique d'évaluation du degré de cryptage et nous comparons ses performances par rapport à l'AES standard, l'AES modifié où nous intégrons nos algorithmes de construction de S-Box dynamique par chaos et aux différents algorithmes étudiés de la littérature.

La deuxième section correspond à l'implémentation des différentes contributions étudiées dans la section précédente. Nous détaillons l'implémentation de ces algorithmes sur un réseau de capteurs. La validation des codes est accomplie en deux phases : une phase de simulation et une phase d'implémentation pratique sur une plateforme réelle de réseaux de capteurs sans fil étendus. Le facteur inhérent à ces réseaux, l'énergie, est mesuré pour les différents algorithmes, par la simulation et par la pratique.

Conclusion Générale : Finalement, nous concluons la thèse en rappelant les contributions apportées et nous présentons les futurs travaux de recherches et les perspectives dégagées.

Annexe : Des détails sur des notions de cryptographie et sur les systèmes chaotiques et les systèmes numériques ont été mis en annexe.

Chapitre 1

Sécurité des réseaux sans fil : Etat de l'Art

1.1 Introduction

La sécurité de l'information demeure parmi les sciences les plus intéressantes durant l'histoire jusqu'à nos jours. Les méthodes d'enregistrer l'information n'ont pas trop changé, typiquement sur des supports en papier ou bien magnétiques envoyée par la suite via des systèmes de télécommunications parfois sans fil. Ce qui a par contre beaucoup évolué, c'est la copie et la modification illégale de ces données ; d'où la nécessité de les protéger surtout lorsqu'il s'agit de communication via un réseau local sans fil. Cette protection peut être assurée par la cryptographie. Plusieurs protocoles et algorithmes ont été développés dans ce sens.

Nous présentons dans ce chapitre les caractéristiques des réseaux locaux sans fil et des réseaux WSN. Nous exposons aussi des notions de base de la cryptographie classique et nous présentons des algorithmes de cryptage conventionnels répandus pour la sécurisation des réseaux sans fil : WLAN et en particulier les réseaux de capteurs.

1.2 Réseau local sans fil (WLAN)

Après l'avènement et la popularité de la norme IEEE 802.3 Ethernet, la nécessité pour les réseaux locaux sans fil a été ressentie dans le milieu des années 90 [1], où le but est de remplacer les réseaux filaires, dans des environnements publics ou privés, en offrant une mobilité aux clients avec un débit comparable. Un réseau local sans fil est un réseau qui relie entre plusieurs terminaux (console de jeux, téléphone mobile, ordinateur portable, assistant personnel, etc.) via des ondes électromagnétiques (radio, infrarouge ou laser), dans une zone de couverture de quelques centaines de mètres de diamètre. Les recherches des instituts ETSI, et IEEE et du groupe MMAC au sein de l'association Japonaise ARIB ont mené à trois familles de standards pour les WLAN [2]:

- 802.11 (IEEE)
- HiperLAN (ETSI)
- HisWAN (ARIB)

La famille de standards la plus répandue et commercialisée est IEEE 802.11. La maturité de cette technologie est marquée par la norme "IEEE 802.11a" divulguée en 1999.

Plusieurs standards et protocoles ont été développés dès lors pour cette norme et d'autres familles de réseaux ont émergé pour des applications spécialisées comme WPAN et WMAN (figure 1.1). Le réseau WPAN est un descendant du réseau local sans fil qui vise des portées limitées allant de quelques mètres à quelques dizaines de mètres seulement.

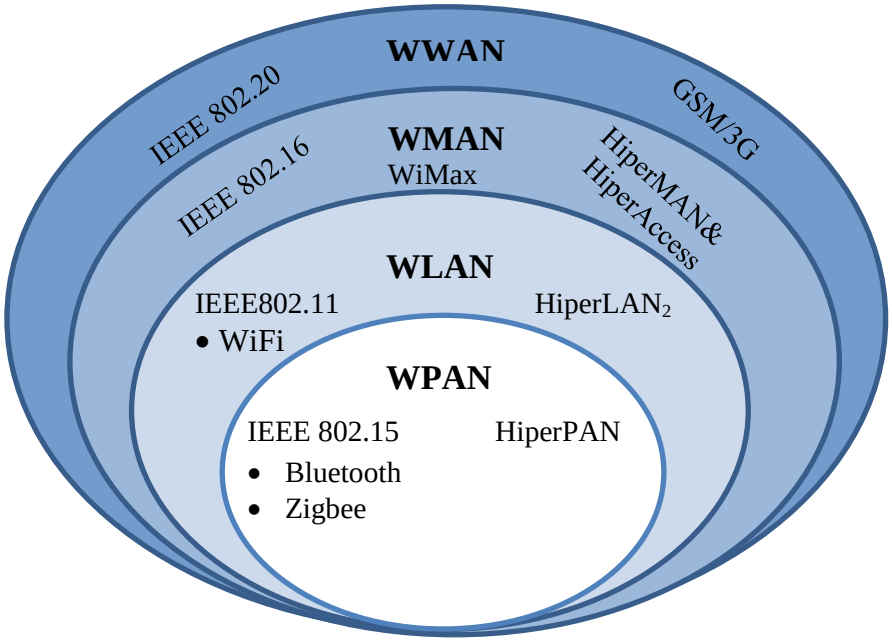


Figure 1. 1. Hiérarchie des réseaux sans fil.

1.2.1 Les modes d'opération de la norme IEEE 802.11

Il existe deux modes d'opération de la norme IEEE 802.11: mode infrastructure et mode ad hoc.

- Mode infrastructure : Dans le mode infrastructure, on effectue une pré-planification du réseau et on installe des points d'accès dans des emplacements fixes. Tous les hôtes ont à communiquer entre eux via les points d'accès en mode infrastructure reliés à leur tour au réseau filaire. Par conséquent, les hôtes mobiles ne peuvent pas se déplacer librement d'un endroit à un autre et sont limités dans une zone de couverture où il doit y avoir au moins un point d'accès.
- Mode ad hoc : Dans le mode ad hoc, un réseau est formé par des nœuds qui communiquent directement entre eux sans avoir besoin des équipements supplémentaires. Par contre, un protocole de routage est nécessaire pour acheminer les données.

1.2.2 Présentation des principaux standards IEEE 802.11

L'IEEE 802.11 est une famille de normes qui définit le contrôle d'accès au medium (MAC) et la couche physique d'un réseau local sans fil. Les améliorations apportées au premier standard depuis sa normalisation forment l'ensemble des normes IEEE802.11.

- **IEEE 802.11a :**

  Ce standard a été normalisé en 1999. Il applique une technique de multiplexage en fréquence appelé OFDM pour transférer les données avec un taux de transfert maximal 54 Mbit/s sur une bande de fréquence de 5.25 à 5.35 GHz (bande U-NII). Les dispositifs 802.11a ne sont pas compatibles avec les autres technologies 802.11 puisqu'ils opèrent

dans une partie différente du spectre radioélectrique et utilisent des techniques de modulation différentes.

- **IEEE 802.11b :**

C'est la version la plus connue et la plus utilisée des standards IEEE 802.11. Elle opère sur la bande de fréquence 2.4 GHz (bande ISM) avec un débit de 11 Mbit/s. Elle utilise CSMA/CA comme méthode d'accès au canal pour éviter les collisions et la technique d'étalement de spectre par séquence directe DSSS au niveau de la couche physique. Les produits conforment à ce standard sont nommés "WiFi" (Wireless Fidelity). L'application du protocole de sécurité WEP (Wired equivalent Privacy) a commencé avec ce standard afin d'apporter un niveau de sécurité équivalent au réseau filaire.

- **IEEE 802.11g :**

Paru en 2003, ce standard est une amélioration du standard IEEE 802.11a en le transposant de la bande U-NII à la bande ISM. Il est donc compatible à la norme IEEE 802.11b avec un débit maximal de 54 Mbit/s. Le protocole WEP est appliqué aussi pour sécuriser les données.

- **IEEE 802.11i :**

Il implique de nombreux changements, y compris l'ajout de l'algorithme AES pour chiffrer l'information. Il introduit le protocole WPA (Wi-Fi Protected Access) puis WPA2 pour remplacer WEP qui est devenu obsolète en raison de ses faiblesses.

1.3 Les réseaux personnels sans fil (WPAN)

Les réseaux personnels sans fil sont une classe spéciale des WLAN qui supporte des dispositifs électroniques très proches (environ 10 m). Cette technologie est la cible des réseaux locaux simples et réduits dans l'espace, et sert à réaliser des fonctionnalités avancées telles que la surveillance des paramètres physiques du corps humain et l'interconnexion des différents appareils sans fil dans l'environnement personnel [3, 4].

1.3.1 Classifications des réseaux personnels sans fil

Les réseaux personnels sans fil sont une variété qui diffère largement selon les exigences en termes de débit, de consommation d'énergie, de qualité de service, etc. Ils sont principalement classés en ces trois catégories [4], [5] :

- WPAN à haut débit : conçu pour les applications temps-réel et multimédia. Ces applications sont supportées par le standard IEEE 802.15.3 avec un débit maximal égal à 55 Mbit/s.
- WPAN à débit moyen : utilisé pour remplacer les câbles d'interconnexions entre les appareils personnels. C'était l'application originale des WPAN comme prévu par le standard IEEE 802.15.1 (Bluetooth) avec un débit de 1 à 3 Mbit/s.
- WPAN à bas débit (LR-WPAN) : destiné pour les réseaux de capteurs sans fil où la complexité est réduite, le débit est faible et la durée de vie des batteries est estimée longue (mois ou années). Les LR-WPAN sont définis par le standard IEEE 802.15.4 qui permet un très faible débit ne dépassant pas 250 kbit/s.

Dans ce qui suit, nous focalisons sur les réseaux de capteurs sans fil qui constituent une tendance innovatrice modelant les nouvelles générations des produits électroniques.

1.3.2 Les réseaux de capteurs sans fil

Le progrès récent des communications sans fil et de la microélectronique a entraîné le développement d'une nouvelle technologie de capteurs sans fil multifonctionnels à faible puissance et à faible coût. Ces capteurs sont équipés de processeurs embarqués qui leurs permettent d'effectuer des opérations simples et de transmettre des données partiellement traitées. Ces propriétés ont attiré plus d'attention à la nécessité d'avoir des réseaux de capteurs sans fil RCSF ou WSN (Wireless Sensor Network). Un réseau de capteurs sans fil se compose d'un grand nombre de nœuds capteurs qui se condensent dans le milieu à prospecter où la position des nœuds n'est pas fixée pour permettre une flexibilité de déploiement surtout dans un environnement abrupt et dangereux.

Les réseaux de capteurs ont un large spectre d'applications allant du domaine militaire à la vie quotidienne.

- *Applications militaires :* Les réseaux de capteurs sans fil se caractérisent par leur auto-organisation, leur déploiement rapide et par leur tolérance aux erreurs. Ils constituent alors une technologie prometteuse pour les applications militaires. Ces réseaux jouent, généralement, un rôle défensif. En effet, ils effectuent la surveillance des forces ennemies ou amies. Ils récoltent les informations concernant leurs états, leurs équipements et leurs géolocalisations.
  En temps de guerre, ils surveillent les champs de bataille, détectent et prévoient les attaques terroristes nucléaires (projet WATS [6] : repérer les dispositifs nucléaires par détections des neutrons et des rayons gamma), biologique (projet JBREWS [7]) ou chimique.
  Les données envoyées par les nœuds capteurs dans ce cas ont une grande importance sécuritaire et stratégique et doivent être camouflées et non repérables par l'ennemi.
- *Applications médicales :* Les réseaux de capteurs sans fil sont appliqués à la surveillance à distance ou locale des patients, l'enregistrement et parfois l'envoie de leurs signaux vitaux comme la température, le taux de glycémie, la pression artérielle ou le rythme cardiaque aux médecins traitants. Ils peuvent aider les patients à mobilité réduite en détectant leurs chutes, leurs mouvements ou encore programmer la prise des médicaments.
- *La domotique :* Elle se résume au concept de la maison intelligente où des capteurs sont embarqués sur des appareils électroménagers ou à l'intérieur de la maison. Ces capteurs peuvent détecter la présence du propriétaire et lancer automatiquement un programme de fonctionnement lui est approprié (luminosité, chaîne de télévision, volets, …). Ils permettent aussi de déclencher un système d'alarme lié au propriétaire ou à la police dans le cas d'un cambriolage.

1.3.2.1 Définition d'un capteur

Un capteur est un dispositif qui surveille une grandeur physique ambiante qui peut être : l'humidité, la lumière, la température, la pression, les sons, les mouvements, vitesse-accélération, capture d'image, etc. Il permet la transformation de ces données physiques

collectées, généralement, en un signal électrique traité à son tour pour qu'il soit compréhensible et manipulable par l'utilisateur.

La notion de capteur a évolué avec le développement technologique. On parle alors de la notion nœud capteur, capteur intelligent, capteur sans fil ou micro-capteur.

L'idée de départ du projet "Smartdust" était de concevoir des capteurs intelligents, autonomes et communicants avec seulement 1 mm^3 de volume. En effet, un capteur intelligent est un capteur de petite taille capable de mesurer des grandeurs physiques, les traiter et les envoyer à d'autres nœuds du même réseau.

1.3.2.2 Architecture d'un nœud capteur

A l'origine l'architecture matérielle d'un simple capteur intégrait seulement une unité de captage (mesure de données physiques) et une unité d'alimentation (pile, batterie, …). Actuellement un nœud capteur est formé par quatre unités de base (figure 1.2) [8]:

- Unité de captage : Elle est constituée généralement de deux sous unités : un capteur et un convertisseur analogique-numérique (ADC). Le signal analogique produit par le capteur, en se basant sur le phénomène observé, est converti en un signal numérique par le convertisseur ADC. Il est introduit ensuite à l'unité de traitement.
- Unité de traitement : Elle est généralement composée d'un processeur couplé à une petite unité de stockage. Elle est chargée de recueillir et de traiter les signaux capturés avant de les transmettre aux autres nœuds du réseau.
- Unité d'émission-réception : Elle permet de connecter le nœud au réseau sans fil, en envoyant et en recevant les trames de données mesurées, à travers des ondes radio ou optiques.
- Unité d'alimentation : Elle est l'unité la plus importante puisqu'elle définit la validité (durée de vie) du capteur dans son milieu hostile où il est généralement difficile de le réalimenter souvent. Elle comporte une batterie ou une pile et peut être dotée d'une unité d'énergie renouvelable.

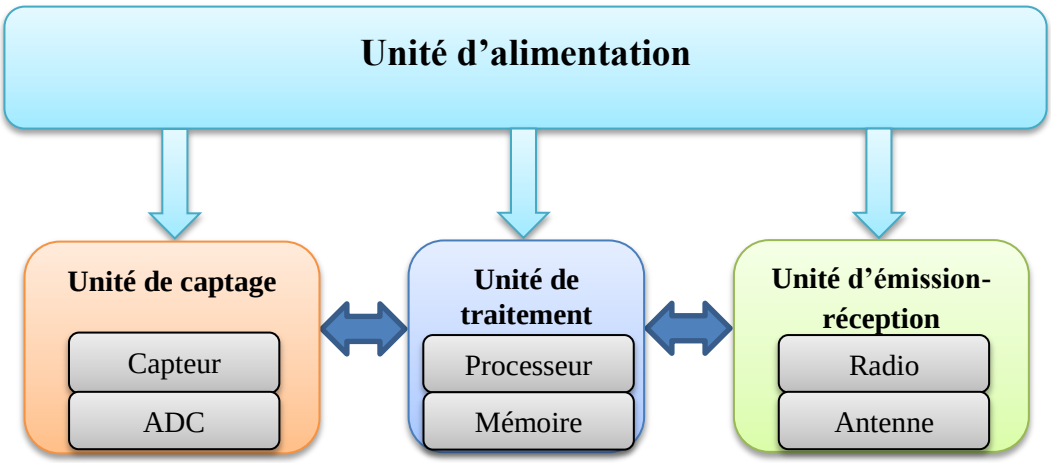


Figure 1. 2. Anatomie d'un nœud capteur.

Un capteur sans fil peut avoir des composants supplémentaires, dépendant de l'application souhaitée, comme un générateur d'énergie (cellule photovoltaïque), un mobilisateur (pour le déplacer en cas de besoin) ou un système de localisation (GPS).

1.3.2.3 Types et caractéristiques des nœuds capteurs

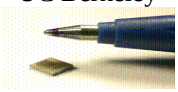
Il existe plusieurs types de nœuds capteurs sur le marché, ils diffèrent selon l'application pour laquelle ils sont conçus et selon les performances et les caractéristiques (type de MCU, radio, mémoire, …) voulues par l'utilisateur. Les deux familles de microcontrôleurs les plus connues sont ATMEGA de Atmel (architecture Harvard) et MSP de Texas Instruments (architecture Von Neumann). Elles sont choisies en raison de leur capacité d'entrer en mode en veille pour préserver l'énergie des batteries [9].

Nous avons essayé de regrouper dans le tableau 1.1 les nœuds capteurs les plus utilisés en détaillant quelques propriétés.

Les contraintes matérielles des réseaux de capteurs sans fil sont intelligibles d'après ce tableau.

- *Faible puissance de calcul :* La faible puissance de calcul est un encombre pour le développement de crypto-système propre aux réseaux de capteurs sans fil. En effet, la fréquence d'horloge des microcontrôleurs ATmel par exemple ne franchit pas le seuil de 20 MHz avec 20 MIPS [9] et les registres sont limités à 8 ou 16 bits.
- *Mémoire limitée :* Les capteurs sans fil ont des ressources en mémoire limitées. Les exemples du tableau 1.1 montrent que la RAM, la mémoire de stockage et la mémoire flash ne dépassent pas 10 KB, 1 MB et 128 KB respectivement. Le développeur doit être alors vigilant en termes de mémoire et chaque solution de sécurité proposée doit être restreinte au niveau de la taille du code.
- *Energie limitée :* Dans la plus part du temps, un capteur est déposé dans des milieux hostiles où l'accès et le chargement des batteries est délicat. L'énergie doit être préservée au maximum, ce qui ajoute un défi de plus pour le choix des crypto-systèmes les plus appropriés. Toutefois, la communication entre les nœuds est l'activité qui dissipe le plus d'énergie et doit être traitée avec précaution.
- *Faible débit et portée radio :* Le débit pour la plupart des nœuds capteurs est de 250 kbit/s et la portée est de quelques dizaines de mètre. Des algorithmes de routage s'avèrent généralement important pour acheminer les données du nœud capteur vers le nœud puits (sink : un nœud capteur qui assure l'interface avec le milieu extérieur et les opérations d'entretien et de réparation du réseau).

Tableau 1. 1. Caractéristiques des nœuds capteurs les plus utilisés.

Capteurs	MCU	RAM/ Stockage(Measurement (serial) flash)/Flash	Radio	Batterie	Dimension
Rene (1999) Berkeley 	ATMEL 90LS8535	512B/32KB/8K	TR1000 (916 MHz)	-	-
MICA2 (2002) UCBerkeley/Crossbow 	ATmega 128L	4KB/512KB/128K	Chipcon CC1000 (315, 433 ou 868/916MHz)	2×AA	58×32×7 mm^3
Spec(2003) UC Berkeley 	AVR RISC	3KB/0KB/0KB	RF	-	2×2.5 mm^2
TelosB (2004) UC Berkeley/ Crossbow 	TI MSP430	10KB/1MB/48KB	CC2420	2×AA	65×31×6 mm^3
MICAz (2004) Crossbow 	ATmega 128L	4KB/512KB/128KB	CC2420	2×AA	58×32×7 mm^3
Tmote Sky (2004) Motiv 	TI MSP430	10KB/1MB /48KB	CC2420	-	3.2×8×1.3 cm^3
Iris (2008) Crossbow 	ATmega 1281	8KB/512KB/128KB	CC2420	2×AA	58×32×7 mm^3

1.3.2.4 Spécificités des réseaux de capteurs sans fil

Dans un réseau de capteurs, des centaines ou des milliers de nœuds sont déployés sur le champ. La distance qui sépare un nœud d'un autre varie de quelques mètres à quelques dizaines de mètres avec une densité qui peut arriver à 20 nœuds/m^3 [8]. Ce nombre élevé de nœuds requière une manipulation prudente de la topologie et une attention particulière à la pile protocolaire. Pour les sept couches du modèle OSI (figure 1.3) nous avons détaillé la

couche physique et la sous-couche MAC de la couche liaison de données, qui nous intéressent dans la suite du travail.

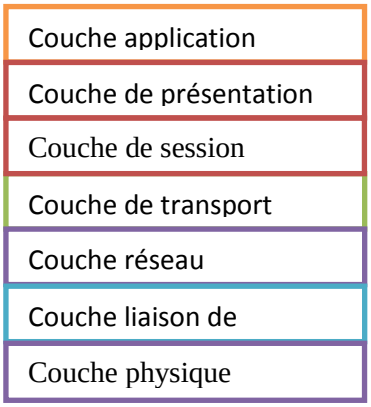


Figure 1. 3. Les sept couches du Modèle OSI.

- **Topologie :**

Les nœuds capteurs peuvent être placés en masse ou un par un dans l'environnement. Ils peuvent être mobiles ou fixes. Les topologies des réseaux de capteurs sont généralement aléatoires (topologie en toile ou mesh) tel que l'exemple de la figure 1.4, où les nœuds communiquent avec les autres nœuds dans leur zone de couverture. L'information destinée à des nœuds distants circulent d'un nœud à un autre jusqu'à livraison, c'est la transmission multisaut. La station de base (puits) assure le collecte, le stockage des données et interfaçage avec les réseaux extérieurs, elle peut être un PC portable ou un nœud capteur plus puissant. D'autres topologies peuvent exister comme la topologie en étoile ou la topologie hybride.

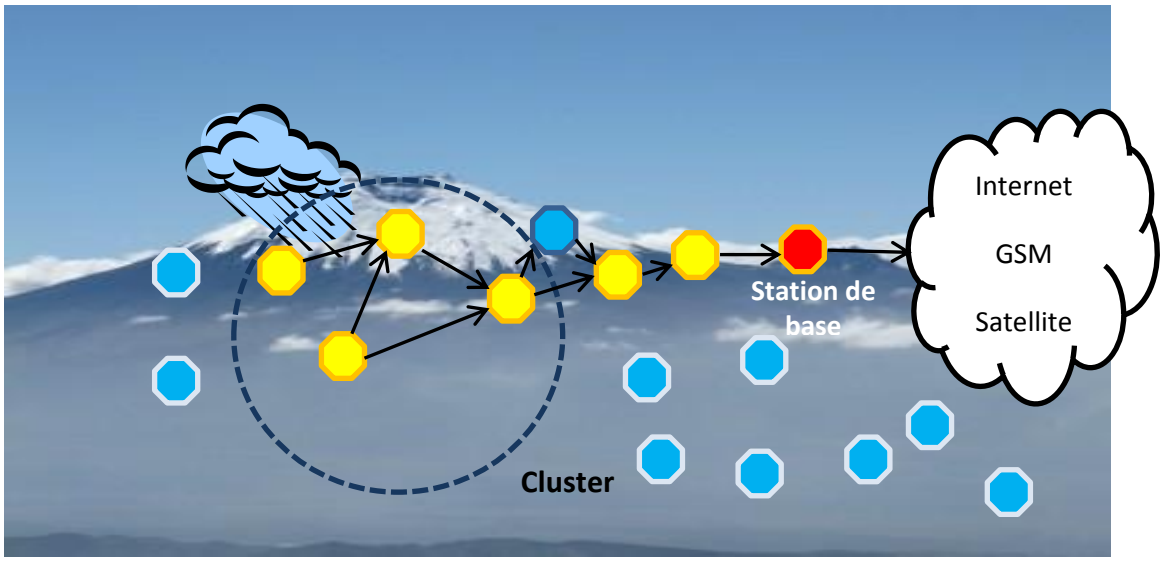


Figure 1. 4. Topologie aléatoire d'un réseau de capteurs sans fil.

- **Couche physique**

La couche physique porte sur le besoin d'avoir une modulation à la fois simple et robuste, avec des techniques d'émission-réception à moindre consommation. Elle est responsable aussi de la sélection de fréquence d'émission, la génération de fréquence porteuse, la détection et la conversion des signaux (du binaire vers électrique ou inversement).

Généralement, la bande ISM de fréquence 915 MHz est suggérée pour les réseaux de capteurs sans fil [8], mais actuellement, les nœuds capteurs sont dotés de l'émetteur-récepteur radio CC2420 conforme au standard IEEE 802.15.4 et qui fonctionne à la bande de fréquence 2.4 GHz.

- **Sous-couche MAC (Medium Access Control)**

Elle fait partie de la couche liaison de données du modèle OSI. Elle définit les conditions d'accès des systèmes aux données du réseau.

Le protocole MAC dans un réseau de capteurs sans fil, à multi-saut, auto-organisé ; il doit accomplir deux objectifs :

- Le premier est de créer l'infrastructure du réseau (établir les liens de communication pour transférer les données).
- Le deuxième est de partager les ressources de communication d'une manière efficace et équitable entre les nœuds.

Il doit faire face aussi au problème de dissipation d'énergie [10], due essentiellement à:

- *L'écoute abusive (over hearing) :* Elle se produit lorsqu'un nœud reçoit des paquets destinés à un autre nœud. Tous les nœuds se trouvant dans la zone de couverture d'un nœud émetteur sont la cible de cette écoute. La perte d'énergie peut augmenter si le trafic dans le réseau est important et si la densité des nœuds est élevée.
- *L'écoute à vide (idle listening) :* Lorsqu'un nœud est à l'écoute du canal radio pour recevoir des paquets bien qu'il n'existe pas des données qui circulent dans le réseau. Dans ce cas, le nœud reste longtemps en état inactif ; ce qui consomme énormément d'énergie. Toutefois, dans plusieurs protocoles MAC par exemple les protocoles IEEE 802.11 en mode ad hoc et CSMA, le nœud est à l'écoute du trafic à l'attente d'un paquet probable. Cette action consomme de 50 à 100 % de l'énergie requise pour recevoir le trafic de données [10].
- *Les collisions des trames :* Les collisions se produisent lorsque deux nœuds émettent leurs paquets en même temps. De ce fait, les paquets sont corrompus et doivent être retransmis. La retransmission augmentera la consommation d'énergie et le temps de latence.
- *Surdébit des paquets de contrôle :* Le nœud peut envoyer, recevoir ou même être à l'écoute d'information additionnelle telle que les paquets de contrôle. Ceci augmente le débit de transmission mais diminue la vitesse de transmission et le débit effectif utile.

Le reste de la couche liaison de données s'occupe du contrôle d'erreur et du contrôle de débit. Le contrôle d'erreur assure l'exactitude des données reçues et le contrôle du débit permet de régler le flux de l'information transmise pour protéger un récepteur lent d'être submergé par les données reçues.

Elle organise les bits physiques en groupes logiques. Sur cette couche, le message est appelé une trame.

- **La norme IEEE802.15.4/Zigbee**

La norme 802.15.4 a été divulguée en 2003, elle définit les couches basses de la pile protocolaire d'un réseau de capteurs sans fil à savoir la couche physique et la couche MAC. Les éléments clé de la norme IEEE 802.15.4 sont la faible consommation, la faible complexité, le faible débit, la faible portée et le bas prix.

La couche MAC de la norme IEEE 802.15.4 emploie deux modes de fonctionnement : un mode non-beacon (sans balise) utilisant la méthode d'accès CSM/CA et un mode beacon (avec balise). Le mode beacon se base sur la synchronisation des nœuds avec le nœud principal ou routeur.

Zigbee est un protocole ratifié en 2004 par un ensemble de grandes entreprises qui forment Zigbee Alliance. Il est l'étendu du protocole IEEE 802.15.4 pour les couches supérieures.

1.4 Notions de bases et primitives cryptographiques

La cryptologie est une discipline moderne ayant une relation étroite avec plusieurs domaines tels que la théorie de l'information, l'arithmétique modulaire ou même les codes correcteurs d'erreurs. Elle est une science mathématique qui englobe la cryptographie (la construction de cryptosystèmes) et la cryptanalyse (la recherche de failles dans les cryptosystèmes).

1.4.1 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 [11].

- La confidentialité permet de garantir que seul le destinataire ou le détenant de la clé puisse découvrir le message en clair.
- L'intégrité permet la non modification ou non altération des données pendant le stockage ou la transmission.
- La non répudiation empêche de nier la participation à un échange ou 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 [11], [12]. La sécurité d'un système de chiffrement est beaucoup plus sûre si sa cryptanalyse fait un temps qui n'est pas meilleure que celui de la recherche exhaustive de la clé. Cette évaluation est appelée la sécurité calculatoire.

Deux autres critères déjà annoncés à l'introduction sont essentiels pour construire un chiffrement sécurisé, ces deux techniques de bases utilisées pour obscurcir les redondances

dans un message sont, selon Shannon, la confusion et la diffusion. Un chiffrement qui vérifie ces deux propriétés s'avère difficile à être cassé [13], [14].

La **confusion** est la relation complexe entre clé, message clair et message crypté. La méthode la plus simple pour appliquer la confusion est la substitution. Les exemples de chiffrement par substitution sont : Vigenère, Xor, César, Enigma.

La **diffusion** consiste à répartir la redondance du message clair et de la clé sur la plus grande longueur possible du message crypté. On peut avoir la propriété de diffusion par simple transposition ou permutation.

Le principe de chiffrement et de déchiffrement est donné par la figure 1.5. Le texte clair est chiffré en utilisant la relation de chiffrement $C = E_{Ke}(M)$ et la restitution de ce message se fait par la fonction de déchiffrement $M = D_{Kd}(C)$ où K_e et K_d sont les clés utilisées. En outre, le chiffrement et le déchiffrement (respectivement la fonction E et D) reposent sur deux méthodes : La cryptographie symétrique et la cryptographie asymétrique.

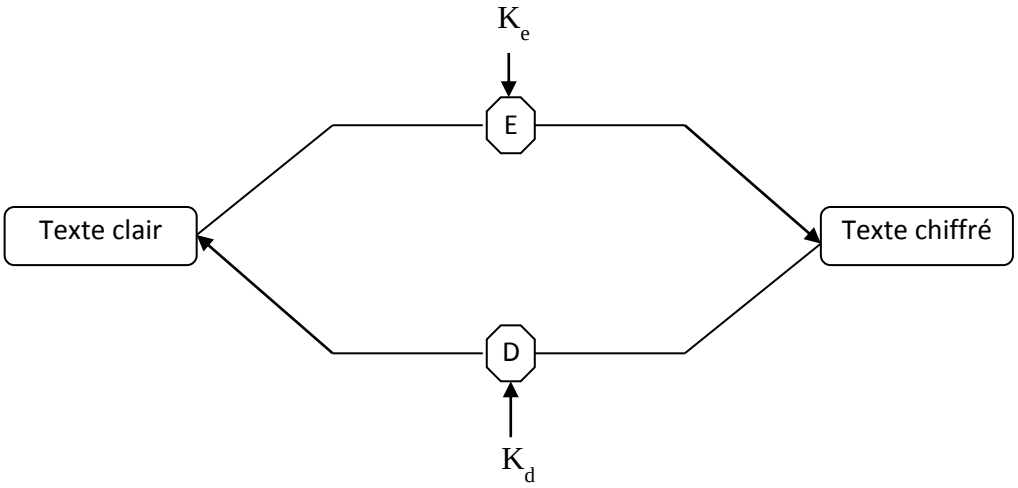


Figure 1. 5. Principe de chiffrement et déchiffrement.

1.4.1.1 Cryptographie symétrique

Dans le cas de la cryptographie symétrique (clé secrète), la relation entre les clés K_e et K_d est tels que soit $K_e = K_d$, soit la connaissance d'une des deux clés permet d'en déduire facilement l'autre.

L'émetteur et le récepteur utilisent la même clé qui doit être privée (figure 1.6) ou bien une clé peut être déduite de l'autre.

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

a) ***Chiffrement par flot (Continu) :***

Chaque bit est traité directement ; c'est-à-dire on opère sur un flot continu de données. Ce mode est adapté surtout pour la communication en temps réel et implémenté en général sur des supports hardwares. Il est plus facile à analyser.

b) *Chiffrement par bloc :*

Chaque message est divisé en blocs de tailles fixes. On peut ajouter des bits néants à la fin du message pour obtenir des blocs entiers (généralement de 64 bits). Ce mode est adéquat pour l'implémentation logicielle en général. Ce chiffrement est plus répandu vu les performances logicielles des algorithmes. La sécurité augmente lorsque les blocs ont une longueur plus grande, mais la durée du traitement augmente alors notablement.

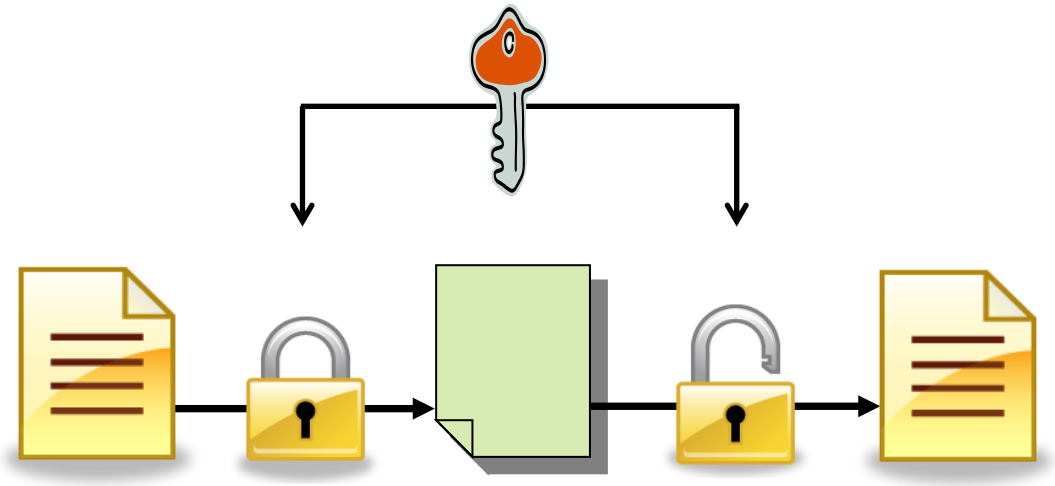


Figure 1. 6. Principe de cryptographie par clé Privée.

Les exemples de cryptage symétrique par blocs sont multiples, on peut citer le DES et l'AES qui est le plus récent.

- **L'algorithme DES**

La norme DES est adoptée par NSA en 1967. C'est un algorithme de chiffrement par bloc qui rassemble les deux techniques de base : la confusion (substitution) et la diffusion (permutation).

Cette norme fonctionne avec une clé de 64 bits, dont 56 sont utilisés pour le cryptage (les 8 bits restant sont utilisés comme bits de parité). L'algorithme est composé de trois étapes :

1) On applique une permutation P à un bloc x de 64 bits et on obtient une chaîne x_o tel que :

$$x_o=P(x)=L_o\ R_o \qquad (1.1)$$

L_o Contient les 32 premiers Bits de x_o et R_o les 32 restants.

2) 16 tours d'une certaine fonction sont effectués. On calcule L_i et R_i , $1 < i < 16$ tel que :

$$\begin{aligned} L_i &= R_{i-1} \\ R_i &= L_{i-1} \oplus f(R_{i-1},K_i) \end{aligned} \qquad (1.2)$$

f est une fonction à deux variables : une à 32 bits correspondant à R_{i-1}, et l'autre de 48 bits représentant K_i.
K_i est obtenu par diversification de la clé K de taille 56 bits.

3) La permutation inverse P^{-1} est appliquée à $R_{16}L_{16}$ pour obtenir un bloc chiffré y. La puissance réside dans les 16 itérations où le message source sera indétectable.

Longtemps inviolable, le DES n'a pas résisté à l'augmentation de la puissance des ordinateurs.

- **L'algorithme AES**

Rijndael est le chiffrement par bloc retenue comme nouveau standard américain de chiffrement (AES) choisi par le NIST en 2001 pour remplacer le DES. C'est un chiffrement par blocs encodant 128 bits avec des clés de 128, 192 ou 256 bits.

- *Principe de l'AES*

Les entrées - sorties d'AES sont constituées par des séquences de 128 bits interprétées comme des éléments du corps fini à 256 éléments IF_{256}. Tout flux d'octets est organisé sous forme matricielle (figure 1.7), cette matrice sera formée de 4 lignes et un nombre de colonnes en fonction de la taille du flux [15].

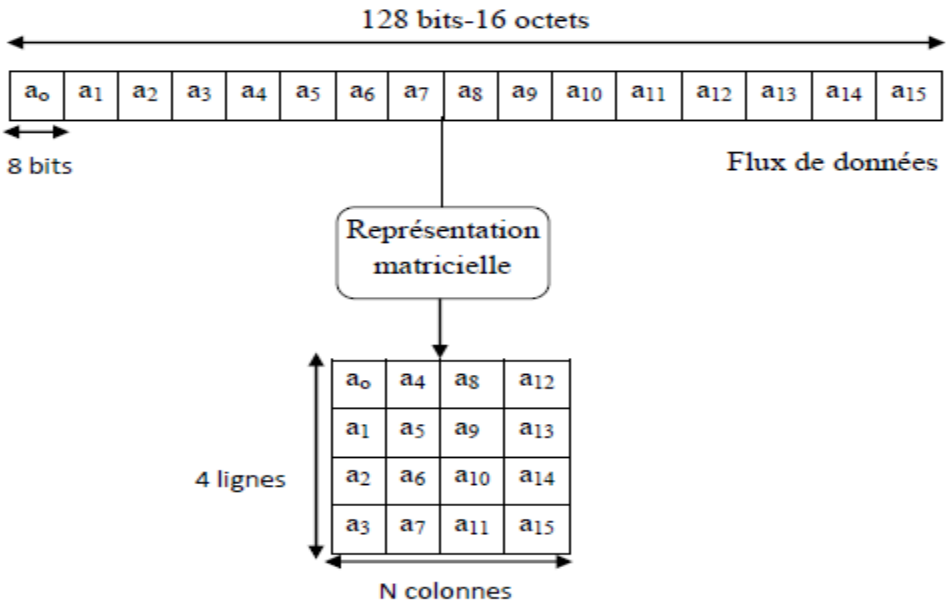


Figure 1. 7. Représentation matricielle d'un flux de seize octets.

Pour une clé de longueur 128 bits, le nombre d'itérations correspond à N_r=10.
On commence par l'addition de la clé secrète bit à bit au message, puis on itère une fonction N_r-1 fois.

Les étapes principales de cette fonction sont :

1) SubBytes : Il s'agit d'une substitution non linéaire où chaque octet est remplacé par un autre choisi dans une table particulière de substitution inversible (Boîte S-Box). La S-Box est la composition de deux transformations (une inversion multiplicative $t : a \rightarrow a^{-1}$ dans GF (2^8) et une transformation affine f) tel que :

$$SBox[a] = f(t(a)), \ \forall a \in F_{256} \tag{1.3}$$

La transformation de Sub-Byte est donnée par la figure 1.8.

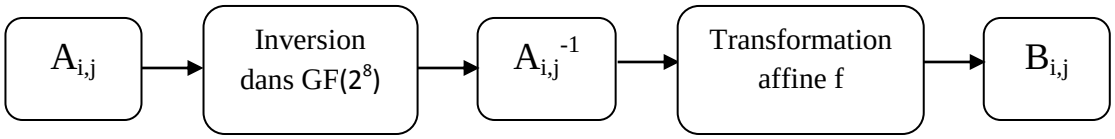


Figure 1. 8. Transformation Sub-Byte.

2) ShiftRows : Chaque élément de la matrice est décalé cycliquement à gauche d'un certain nombre de colonnes. Par exemple la ligne i est décalée de C_i éléments, si bien que l'élément en position j de la ligne i est décalé de ($j-C_i$) éléments mod N_b.

3) Mixcolumns : On opère sur les colonnes c de la matrice State en les traitant comme un polynôme a(x) de degré 3 à coefficients dans F_{256}. On effectue une multiplication par un polynôme c(x) suivie d'une réduction modulo le polynôme x^4+1 :

$$(3x^3+x^2+x+2) \times a(x) \bmod (x^4+1) \tag{1.4}$$

Matriciellement, cette opération s'écrit :

$$b(x) = c(x) \times a(x) \bmod (x^4+1) \Leftrightarrow \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \times \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} \tag{1.5}$$

4) AddRoundkey : L'addition terme à terme de la matrice avec une clé de ronde obtenue en diversifiant la clé (à partir de la clé K de 4 N_k), crée une clé étendue de 4 N_b (N_r+1) octets.
N_b représente le nombre de mots de 32 bits dans un bloc, il représente aussi le nombre de colonnes pour une représentation matricielle, N_r est le nombre de ronde et N_k est la longueur de la clé.
Enfin une dernière ronde correspond à refaire les étapes précédentes en omettant l'étape Mixcolumns.

c) *Avantages et inconvénients du cryptage symétrique*

Pour ce type de cryptage classique, deux problèmes le mettent en cause :

- ✓ Le secret ou la clé doit être échangé à toute communication.
- ✓ La distribution de N×(N-1)/2 clés pour N utilisateurs d'un seul réseau.

Le tableau 1.2 résume les avantages et les inconvénients du chiffrement symétrique.

Tableau 1. 2. Avantages et inconvénients du chiffrement symétrique.

Avantages	Inconvénients
• Assure la confidentialité des données. • Algorithme de cryptage performant. • Plus utilisé pour la transmission de long message (débit plus important). • Les clés sont relativement de faible taille. • Primitive de mécanismes cryptographiques.	• Problème de distribution de clés : trouver un canal parfaitement sûr pour transmettre la clé. • Problème de Gestion des clés.

1.4.1.2 Cryptographie asymétrique (à clé publique)

Dans les cryptosystèmes asymétriques (clé publique), la connaissance de la clé K_e (la clé de chiffrement) ne permet pas d'en déduire celle de K_d (la clé de déchiffrement).

La clé K_e est appelée aussi clé publique et la clé K_d est appelée clé privée (figure 1.9).

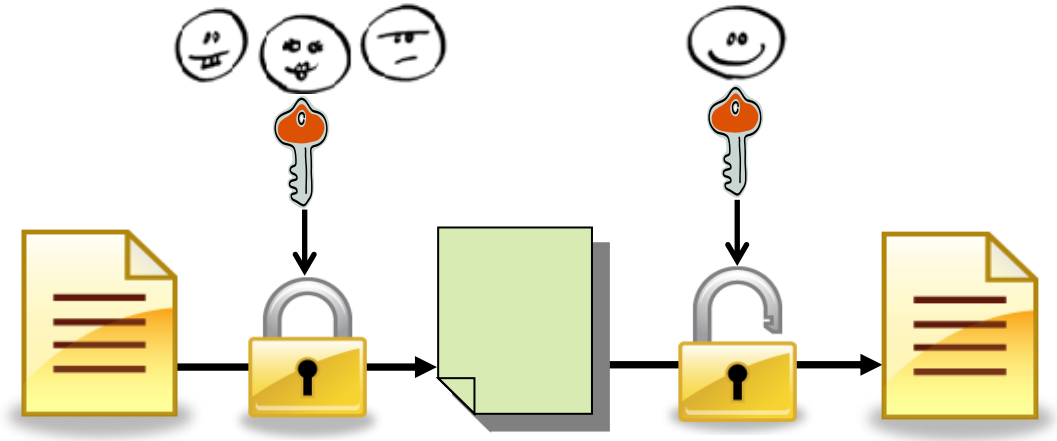


Figure 1. 9. Principe de cryptographie par clé Publique.

Dans la plus part des implémentations, le cryptage à clé publique est utilisé pour sécuriser et distribuer les clés, qui sont utilisées avec les algorithmes symétriques [13]. Les avantages et les inconvénients du chiffrement asymétrique sont regroupés dans le tableau 1.3.

Tableau 1. 3. Avantages et inconvénients du chiffrement asymétrique par rapport au symétrique.

Avantages	Inconvénients
• La distribution des clés est simplifiée: La clé privée n'est jamais révélée ou transmise et la clé publique est disponible à tous les utilisateurs. • Certification des clés publiques par la signature numérique [13]. • La paire de clé privée/publique reste inchangée pour une longue durée. • Le nombre des clés distribuées dans un large réseau est faible par rapport à celui d'une cryptographie à clé symétrique.	• Visiblement plus lent que les algorithmes symétriques [13]. • Garantir que la clé publique que l'on saisit est bien celle de la personne à qui l'on souhaite faire parvenir l'information cryptée : attaque d'usurpation d'identité. • La taille des clés est beaucoup plus importante que les clés symétriques [13].

- ***Exemple de cryptage à clé publique***

L'algorithme RSA a été inventé par Rivest Shamir et Adleman en 1977. Cet algorithme est un chiffrement à clé publique (ou chiffrement asymétrique). Les utilisateurs de cet algorithme doivent posséder deux clés privée et publique. Parmi les points forts du RSA est la difficulté de factoriser des grands nombres. Les clés publique et privée sont considérées comme des fonctions à grand nombre premier avec par exemple 100 et 200 chiffres.

- *Création des clés*

Pour former les clés, il faut choisir deux nombres premiers différents p et q, avec n est le module de chiffrement tel que $n = p \times q$.

Il faut calculer ensuite l'indicatrice d'euler suivante : $\varphi(n) = (p-1)(q-1)$, puis choisir e comme étant un entier premier avec $\varphi(n)$, appelé exposant de chiffrement tel que : $p,q < e < \varphi(n)$.

Déterminer d tel que $(e .d \bmod(\varphi(n)))=1$ et $p,q<d<\varphi(n)$.

On forme alors les deux clés : La clé privée (n, d) et la clé publique (n, e)

RSA affirme que si la taille des clés est supérieure à 2048, l'algorithme est considéré incassable.

- *Chiffrement*

Soit M un entier inférieur à n désignant le message à crypter, le message crypté C sera donc de la forme : $C = M^e \bmod(n)$.

- *Déchiffrement*

Le déchiffrement est possible en connaissant d, on retrouve le message clair en appliquant la formule suivante : $M = C^d \bmod(n)$.

1.4.1.3 Cryptographie Hybride

Les algorithmes à clé publique sont très lents : au moins mille fois plus lents que les algorithmes symétriques. En plus la cryptographie à clé publique est vulnérable à l'attaque à

texte chiffré choisi [13]. Un autre mode de chiffrement est le chiffrement hybride qui fusionne les caractéristiques des deux modes : symétrique et asymétrique. Il se résume dans les quatre étapes suivantes :

1) Bob envoie sa clé publique à Alice.
2) Alice génère une clé de session aléatoire, K, et le crypte à l'aide de la clé publique de Bob, et l'envoie tel que la nouvelle clé sera $E_B(K)$.
3) Bob décrypte le message d'Alice en utilisant sa clé privée pour restituer la clé de session $D_B(E_B(K))=K$.
4) Ils chiffrent par la suite leurs messages en utilisant cette clé de session.

L'utilisation du chiffrement à clé publique résout le problème de distribution de clé.

1.4.2 La cryptanalyse

La cryptanalyse est l'étude des informations cryptées dans le but de trouver des faiblesses (failles), d'en découvrir le secret et décrypter les textes chiffrés. Le décryptement est l'art de retrouver le texte en clair sans savoir la clé de cryptage.

La cryptanalyse traditionnelle associe l'application d'outils mathématiques à la recherche de motif, et à la résolution analytique. La patience, la détermination et la chance peuvent être parmi les ingrédients de réussite d'un cryptanalyste. Les cryptanalystes sont également appelés des pirates ou hackers.

Les techniques de cryptanalyse peuvent se résumer en cinq niveaux d'attaques liés aux données utilisées:

– *L'attaque par force brute* (Brute-force attack) : Le cryptanalyste teste toutes les combinaisons de clés possibles jusqu'à l'acquisition du texte clair.

– *L'attaque sur texte chiffré seul* (Ciphertext-only attack) : Le cryptanalyste ne connaît que le message chiffré par l'algorithme et essaye de déduire la clé ou le texte clair. Le manque d'information rend la cryptanalyse plus délicate.

– *L'attaque à texte clair connu* (Known-plaintext attack) : Le cryptanalyste possède le texte ou parties du texte en clair et leur correspondance chiffrée.

– *L'attaque à texte clair choisi* (Chosen-plaintext attack) : Le cryptanalyste peut choisir le texte en clair, et il peut produire la version chiffrée de ce texte (il a accès à la machine à crypter) avec l'algorithme considéré dès lors comme une boîte noire. Les techniques de cryptage asymétrique sont notamment sensibles à ce type d'attaque.

– *L'attaque à texte chiffré choisi* (Chosen-ciphertext attack) : Le cryptanalyste possède le texte chiffré et peut obtenir le texte en clair associé.

Pour vérifier la sécurité d'un cryptosystème récemment conçu, quelques éléments sont indispensables à analyser. Des algorithmes de cryptanalyse appartenant aux modes cités précédemment ont été conçus et développés en fonction de la robustesse et les caractéristiques des algorithmes de cryptage proposés.

Dans ce qui suit, nous allons détailler quelques notions essentielles pour mesurer le niveau de sécurité.

1.4.2.1 Cryptanalyse linéaire

Pour casser l'algorithme DES, Mitsuri Matsui a conçu en 1993 cette technique de cryptanalyse linéaire. Cet algorithme est une attaque à texte clair connu. Son principe tire profit des probabilités élevées des occurrences des expressions linéaires déduites du texte clair et du texte chiffré. Ces expressions linéaires sont construites à partir d'approximation linéaire de l'algorithme à crypter. Pour crypter l'algorithme DES, une bonne implémentation aura besoin à peu près de 2^{39} couples chiffrés par la même clé. La faille du DES réside à une certaine caractéristique linéaire de ses S-Box (table de substitution) qui doivent être normalement non linéaires. Chaque algorithme de cryptage doit résister à cette attaque. On en parlera avec plus de détails dans d'autres sections.

1.4.2.2 Cryptanalyse différentielle

La cryptanalyse différentielle est conçue par Biham et Shamir en 1993. C'est une attaque par texte clair choisi. L'attaquant doit être capable d'avoir des extraits de texte crypté à partir d'un texte clair à son choix. Elle peut casser l'algorithme DES après 247 couples de texte choisi. Elle étudie l'effet des différences entre les textes d'entrée sur les différences de leurs sorties où l'on cherche des différences constantes ou un biais. Les différences entre les paires de texte se font généralement par la fonction OU-exclusif. Cette attaque sera aussi détaillée dans des sections suivantes.

1.4.2.3 Cryptanalyse algébrique

La plupart des algorithmes de cryptage modernes sont conçus de sorte qu'ils résistent aux attaques classiques (linéaire et différentielle).

Courtois et Pieprzyk ont étudié la sécurité de ces algorithmes en évoquant une autre hypothèse : le système peut être écrit sous forme d'équations algébriques. Ils ont résolu que l'AES contient 23 équations quadratiques linéairement indépendantes qui peuvent être résolues à l'aide de leur nouveau algorithme de cryptanalyse "XSL" [16].

XSL est une méthode de cryptanalyse pour les chiffrements par blocs. Cette attaque s'avère plus rapide que la recherche exhaustive, mais sa mise en pratique est encore sujet de controverse vue la puissance de calcul qu'elle nécessite.

1.5 Sécurité de la couche MAC

La couche MAC gère la transmission et l'accès au canal. La sécurité des données transmises de la norme IEEE802.15.4/Zigbee s'effectue au niveau de la couche MAC à l'aide de l'algorithme AES.

La couche MAC fournit donc le processus de sécurité, mais les couches supérieures déterminent les clés de cryptage et le niveau de sécurité à appliquer.

La structure de la trame dans la couche MAC du réseau de capteur sans fil est illustrée par la figure 1.10.

Octets	2	1	4-20	0-122	2
	Tramede contrôle	N° de séquence de données	Adresse de l'information	Donnée utile (data payload)	FCS Séquence de contrôle de trame

Figure 1. 10. Structure de la trame de la couche MAC du réseau Zigbee

Où le champ contrôle de trame sert à spécifier la structure et le contenu du reste de la trame.

On détermine l'emploi de la sécurisation ou non grâce à un bit de ce champ qui est réservé à l'activation de la sécurité en le mettant à "1".
La trame est alors protégée par l'algorithme symétrique AES en utilisant une clé qui est stockée localement dans chaque module dans le MAC PIB.

En résumé, les objectifs des services de sécurité dans les réseaux de capteurs sans fil sont:

- Contrôle d'accès : Il fournit une liste (ACL) des équipements valides à partir de laquelle le dispositif peut recevoir des trames de données. Cette liste contient les adresses MAC de ces équipements autorisés. Ce mécanisme empêche les périphériques non autorisés de communiquer sur le réseau en activant le filtrage par adresse MAC. Ceci reste insuffisant pour l'intégralité et la confidentialité des données.
- Cryptage : Seulement les nœuds possédant la clé secrète peuvent décrypter l'information et communiquer.
- Intégrité : Protéger l'information contre les modifications provoquées par un pirate et assurer la réception des données non modifiées.
- Fraîcheur des données : C'est d'empêcher le message d'être accepté plusieurs fois par le récepteur, et à faire en sorte que la trame n'est pas une trame déjà reçue. Un compteur est mis en place pour rejeter les trames qui ont une valeur du compteur inférieure ou égale à la valeur précédente.

1.5.1 Taxonomie des attaques

Les réseaux de capteurs sans fil sont vulnérables à plusieurs types de menaces. Nous avons classé les attaques suivant la couche sur laquelle la menace est effectuée.

1.5.1.1 Attaque sur la couche physique

- *Jamming (Brouillage radio)* : C'est une attaque de type déni de service, utilisée pour tous les réseaux sans fil. Elle consiste à interférer les fréquences radio utilisées avec des signaux envoyés en continue ou sur des intervalles constants ou parfois aléatoires (lors de la détection d'une activité sur le réseau). Ceci empêche les nœuds de communiquer sur le réseau et le réseau devient hors service.

- *Tampering (Accès physique au nœud)* : Les nœuds sont généralement déployés à grand nombre sur une surface étendue, ils sont alors vulnérable au vol, à la destruction ou à la falsification. L'attaquant peut extraire des données critiques comme les clés secrètes de chiffrement.

1.5.1.2 Attaque sur la couche liaison de données

- *Attaque par collision :* Il suffit que l'adversaire provoque la collision sur un seul octet pour que tout le message soit perturbé. Le changement d'une partie des données provoquerait une erreur du code CRC. Un acquittement altéré pourrait causer des back-off avec un délai exponentiel dans certain protocole MAC.
- *Epuisement de batterie :* Si le protocole de la couche de liaison impose au capteur de retransmettre les messages jusqu'à recevoir un acquittement, alors une collision continue peut épuiser la batterie du capteur qui tenterait à retransmettre le message non acquitté.
- *Injustice (unfairness) :* C'est une forme de l'attaque par déni de service partielle. Elle consiste à appliquer les attaques par collision et par épuisement irrégulièrement. Cette attaque n'empêche pas l'accès au canal mais le limite. Le nœud capteur peut manquer, par exemple pour un protocole MAC à temps réel, une date limite de transmission.

1.5.1.3 Attaque sur la couche de routage

- *Attaque de trou noir (Blackhole) :* Cette attaque prend sa nomenclature par analogie avec le trou noir dans l'espace : tout ce qui rentre ne sort jamais. En effet, un nœud malicieux inséré dans le réseau annonce des chemins à coût nul et le réseau achemine tout le trafic vers ce nœud devenu un trou noir.
- *Attaque de trou de ver (Wormhole) :* Un trou de ver est un lien à faible latence établi entre deux nœuds malicieux. Ce lien peut être filaire ou radio (à longue portée). Cette attaque vise le protocole de routage en falsifiant les données des distances réelles et des chemins les plus courts pour envoyer les paquets. Ainsi, les deux nœuds malicieux peuvent contrôler tout le trafic des données et mettre en péril les services du réseau.
- *Altération des paquets :* Un nœud malicieux peut altérer les informations de routage échangées entre les nœuds.
- *Attaque de trou à la base (Sinkhole) :* Cette attaque consiste à proposer aux nœuds voisins des chemins plus courts et des puissances de transmission plus élevées. Ces nœuds vont choisir le nœud malicieux pour acheminer leurs prochaines communications vers la base. Cette attaque facilite le renvoi sélectif (selective forwarding) puisque le trafic d'une grande partie du réseau passera par ce nœud malicieux.
- *Attaque Sybil :* L'attaque Sybil est une attaque où un nœud malicieux présente plus d'une identité dans le réseau en usurpant l'identité d'autres nœuds légitimes. Les algorithmes et les protocoles qui sont facilement affectés incluent les systèmes tolérants aux pannes ou les systèmes de stockage distribués qui autorisent la redondance des données. Le nœud malicieux dans ce type d'attaque vise les services

de routage, de sécurité, d'agrégation de données ou de vote dans les élections du cluster head.

- *Hello flood :* De nombreux protocoles employant les paquets Hello appliquent l'hypothèse suivante : La réception d'un paquet Hello signifie que l'expéditeur est un voisin. Un attaquant peut utiliser un émetteur à haute puissance pour tromper un grand nombre de nœuds.

1.5.1.4 Attaque sur la couche application

- *Inondation (flooding) :* Un attaquant peut demander l'établissement de nouvelles connexions jusqu'à épuisement des ressources requises pour chacune. Les demandes des nœuds légitimes seront ignorées.
- *Attaque par désynchronisation :* Cette attaque se réfère à la rupture d'une connexion existante entre deux nœuds. Un attaquant envoie, par exemple, des messages erronés à deux nœuds légitimes pour qu'ils demandent la retransmission et rompent la synchronisation.

1.5.2 Modes d'opération de sécurité des réseaux de capteurs sans fil

Il existe trois modes d'opération : le mode CTR qui assure le cryptage des données, le mode CBC-MAC qui garantit l'intégrité et le mode CTR CBC-MAC (CCM) qui est la combinaison des deux assurant ainsi à la fois le cryptage et l'intégrité des informations [17].

1.5.2.1 Mode CTR

Dans le mode CTR (figure 1.11), les compteurs : T_1, T_2, … , T_n sont cryptés avec le chiffrement symétrique par bloc (AES) pour produire une séquence de blocs (O) à la sortie tel que :

$$O_j=CIPH_k(T_j). \quad \text{avec } j = 1, 2, \ldots, n. \tag{1.6}$$

Chaque bloc de sortie subit un OU exclusif (XOR) avec le texte clair (P) créant ainsi le texte chiffré tel que :

$$C_j=P_j \oplus O_j. \text{ avec } j=1, 2, \ldots, n-1. \tag{1.7}$$

Le dernier bloc C_n sera un bloc de u bits donné par :

$$C_n=P_n \oplus MSB_u(O_n) \tag{1.8}$$

Le décryptage est assuré par la formule suivante :

$$P_j = O_j \oplus C_j. \quad \text{avec } j=1, 2, \ldots, n-1. \tag{1.9}$$

Le dernier bloc décrypté est :

$$P_n = C_n \oplus MSB_u(O_n). \tag{1.10}$$

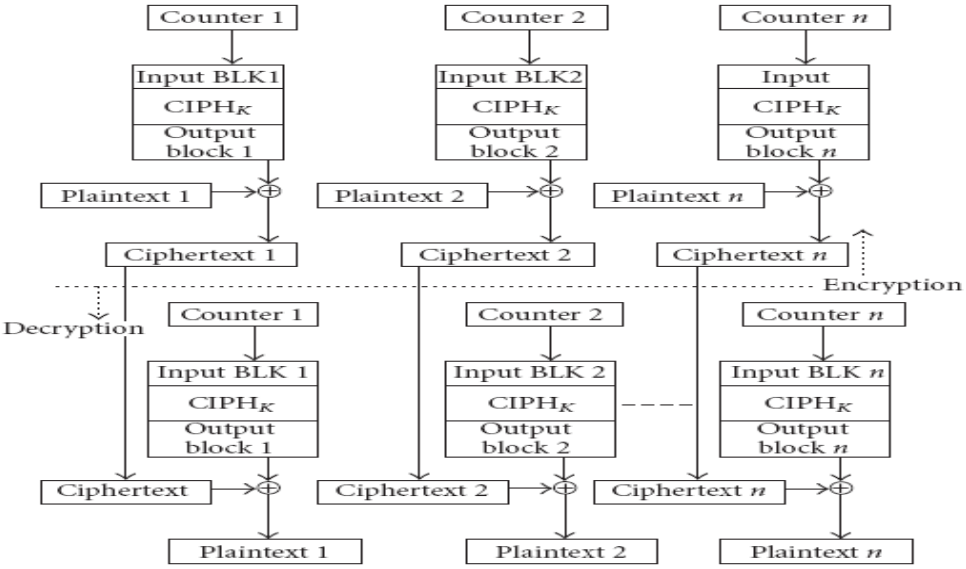


Figure 1. 11. Service de sécurité du mode CTR [17].

1.5.2.2 CBC-MAC

Le mode CBC-MAC utilise un chiffrement en bloc avec une clé K pour crypter les vecteurs d'entrée. On note D et O respectivement les vecteurs d'entrée et de sortie :

$$O_1 = E_K (D_1),\ O_2 = E_K (D_2 \oplus O_1),\ O_3 = E_K (D_3 \oplus O_2),\ \ldots,\ O_n = E_K (D_n \oplus O_{n-1}) \qquad (1.11)$$

Le MIC est les M bits d'extrême gauche de O_n, avec 32< (M = 8 h) <128 ; h est un entier.

Ce mode n'est pas utilisé pour crypter mais pour assurer l'intégrité des données. Pour vérifier cette intégrité du message d'information en comparant le MIC calculé par l'émetteur à celui du récepteur. A partir du MIC, on en déduit un nombre calculé à partir d'un message et envoyé avec celui-ci (équivalent d'un checksum)). Ce mode est expliqué par la figure 1.12.

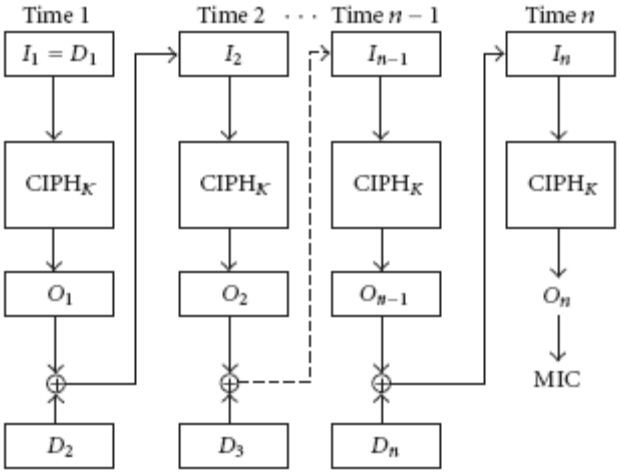


Figure 1. 12. Mode CBC [17].

1.5.2.3 CTR CBC-MAC (CCM)

C'est un mode de cryptage et d'intégrité utilisant un chiffrement par bloc de 128 bits par exemple (AES pour la norme 802.15.4). Il y a trois entrées pour le mode CCM :

- L'information (data payload) est cryptée et authentifiée.
- L'information associée (entête) est authentifiée seulement.
- Le vecteur initial (appelé IV ou nonce) assigné à l'information utile et l'entête.

Le nonce est un nombre aléatoire généré une seule fois dans les protocoles d'authentification pour s'assurer que les communications précédentes ne peuvent être réutilisées dans les attaques de répétition (replay attack).

Le CCM utilise la technique du CTR pour crypter et la technique du CBC-MAC pour l'intégrité. Le CCM est composé de deux méthodes :

1) génération-cryptage : elle nécessite la génération du MIC en premier lieu puis le cryptage.
2) Décryptage-vérification : elle nécessite le décryptage en premier lieu puis la vérification du MIC.

L'émetteur a besoin d'une entrée {K, N, m, a}, où K est la clé de AES, N est le nonce de 15 à L octets, m est le message constitué par une suite de l(m) octets où $0 < l(m) < 2^{8L}$, a est l'information additionnelle authentifiée constituée d'une suite de l(a) où $0 < l(a) < 2^{64}$. L'information additionnelle est authentifiée mais non cryptée, et n'est pas incluse dans la sortie de ce mode. En plus, elle peut être utilisée pour authentifier l'entête du texte clair, qui affecte l'interprétation du message.

Le champ d'authentification est calculé en utilisant CBC-MAC.

Soit $B_0, B_1, \ldots\ldots\ldots, B_n$ une séquence de blocs pour CBC-MAC.

Le CBC-MAC est calculé par :

$$X_i = E_k(B_0);\ X_{i+1} = E_k(X_i \oplus B_i)\ ;\ i = 1, \ldots, n. \qquad (1.12)$$

$$T = \text{premier-M-octets}(X_{n+1}) \qquad (1.13)$$

Le mode CTR est utilisé pour le cryptage, et la clé est définie comme suit :
$S_i = E_k(A_i)$ i = 0, 1, 2, . . ., avec A_i = {F, N, Compteur i}, et F = {bits-réservés (2 bits), 0 (3 bits), L − 1 (3 bits)}, N est le nonce avec 15 – L octets de longueur, et le compteur a de L octets de longueur.

Le message est crypté en utilisant Xor:

$$m \oplus S, \text{ avec } S = l(m) \text{ octets de } S_1 \| S_2 \| S_3, \ldots.., \qquad (1.14)$$

On note que S_0 n'est pas utilisé pour crypter le message (utilisé pour l'authenticité). La valeur de l'authentification (ou MIC) est obtenue comme suit :

$$U = T \oplus \text{first-M-octets}(S_0) \qquad (1.15)$$

Le texte chiffré est :

$$m \oplus S \| U \qquad (1.16)$$

Pour décrypter, le récepteur a besoin de la clé K, du nonce N, de l'information additionnelle a, et du message chiffré C. La clé est générée pour extraire le message m et la valeur de T.

Le message et l'information d'intégrité additionnelle sont utilisés pour calculer la valeur du CBC-MAC et vérifier T.

Si T n'est pas correct, alors le récepteur n'acceptera aucune donnée. En particulier, le récepteur n'acceptera pas le message décrypté, la valeur T ou n'importe quelle information. L'idée est de gérer l'espace des clés en utilisant des clés génériques.

1.6 Conclusion

Les recherches pour améliorer et développer les algorithmes de cryptanalyse sont en développement croissant, ce qui a engendré une rapidité remarquable de "casser" les cryptosystèmes. Une solution proposée est d'augmenter la taille des clés de chiffrement à chaque fois. Cette approche est supportée par l'essor technologique des calculateurs ce qui a ouvert l'horizon d'envisager d'autres alternatives comme les ordinateurs quantiques qui ont montré des capacités prometteuses mettant en cause les algorithmes de cryptage classique.

Cependant, un algorithme de cryptage symétrique par bloc conventionnel comme l'AES n'a pour l'instant pas été cassé et la force brute demeure la seule solution. Il a été conçu de telle manière à rendre des méthodes classiques comme la cryptanalyse linéaire ou différentielle difficiles. Théoriquement, la cryptanalyse algébrique semble casser cette légende pour laisser penser à des remplaçants. Le chaos n'est pas une science nouvelle mais il date des années 60-70, et comme tout axe de recherche le développement technologique permet d'envisager plus de domaines d'applications. L'utilisation du chaos semble une alternative pour résoudre quelques problèmes des crypto-systèmes classiques tels que la gestion des clés ou aussi pour améliorer leur performance du point de vue complexité algorithmique, temps d'exécution, débits ou aussi consommation d'énergie.

Les signaux chaotiques ont des caractéristiques ressemblant au bruit, ils sont généralement à large bande et il est difficile à prévoir leur évolution à long terme ; ils sont alors utiles pour les communications à étalement de spectre. En particulier, puisque les signaux chaotiques ressemblent au bruit et sont difficilement appréciables, il est possible de les utiliser dans des situations où on désire avoir une faible probabilité d'interception.

Les systèmes chaotiques sont utilisés alors dans le cryptage des signaux. Les problèmes majeurs de ces systèmes sont la démodulation et la synchronisation en présence du bruit.

Dans ce qui suit, nous allons aborder la théorie du chaos avec plus de détails.

Chapitre 2

Rappel sur la théorie du chaos et les tests statistiques du NIST

2.1 Introduction

La théorie du chaos fait partie des sciences les plus récentes et est devenue l'un des domaines les plus avancés dans la recherche contemporaine. Les origines de cette nouvelle théorie s'étendent aux mathématiques et physiques des débuts du 20ème siècle, mais elle a émergé dans les années 1960-70.

Durant des années, le chaos était considéré comme incontrôlable et même inutilisable, malgré la mise en équation de certains phénomènes et la démonstration du déterminisme dans des aspects d'apparence aléatoire.

La théorie du chaos est définie comme une étude des systèmes dynamiques non-linéaires complexes ou les systèmes complexes qui sont exprimés par des récurrences et des algorithmes mathématiques et qui sont dynamiques non constants et non périodiques. Elle inclut l'étude qualitative et quantitative d'un comportement instable non périodique et aléatoire des systèmes dynamiques non linéaires déterministes. Le chaos peut être vu aussi comme un système avec des propriétés stochastiques. Dans toutes les définitions qui peuvent exister pour le chaos, un phénomène fondamental est indispensable : la sensibilité aux conditions initiales.

En effet, en programmant son ordinateur et en changeant par 10^{-4} les conditions initiales des prévisions météo, Edward Lorenz a découvert que pour certaine équation ou système d'équations non linéaires les résultats montrent une grande sensibilité aux conditions initiales. On peut dire que cette anecdote est la base du chaos déterministe.

La théorie du chaos influence l'explication de plusieurs phénomènes et trouve son application dans plusieurs domaines tels que :

- Economie : Prévision des cycles économiques, des mouvements commerciaux et des marchés financiers.
- Météo : Prévisions météorologiques.
- Santé : Prévision des crises d'épilepsie.
- Sciences sociales : Comportement des systèmes sociaux.
- Cryptage de l'information.

2.2 Définition et propriétés du signal chaotique

Plusieurs définitions peuvent être données pour le chaos. Le chaos dans son sens linguistique est la confusion générale des éléments, de la matière, avant la création du monde. C'est le désordre. Une autre définition considère le chaos comme l'un des dispositifs intriguant de la dynamique non linéaire déterministe. Les systèmes dynamiques sont toute chose qui varie dans le temps. Les exemples sont diversifiés comme le pendule ou le système solaire. Il existe un espace d'état ou un espace de phase qui contient tout état possible du système et sa loi d'évolution qui décrit le futur lorsque le présent est donné [18].

Une définition proposée par Devaney pour les systèmes à temps discret est la suivante :

Soit (χ, δ) un espace métrique compact et $f : \chi \rightarrow \chi$, une fonction.

Le système dynamique à temps discret $x_{k+1} = f(x_k)$ est dit chaotique si les conditions suivantes sont vérifiées :

- Sensibilité aux conditions initiales.
- Transitivité topologique.
- Densité des orbites périodiques.
- Ergodicité.

2.2.1 Sensibilité aux conditions initiales

En faisant la troncature de quelques chiffres de conditions initiales de son système de prévision météorologique, Lorenz a mis en relief le caractère le plus important des systèmes chaotiques qui est la sensibilité aux conditions initiales. Mais en fait c'est avant cette anecdote, que ce phénomène a été découvert. Vers la fin du 19ème siècle, Poincaré montrait que les trois orbites de 3 corps en mouvement sous une force centrale due à la gravité changent radicalement avec une petite modification des conditions initiales.

Pour un système chaotique, une très petite erreur sur la connaissance de l'état initial x_0 dans l'espace des phases va se trouver (presque toujours) rapidement amplifiée.

Il existe un nombre réel $\beta > 0$, tel que pour tout $x_0 \in \chi$ et pour tout $\varepsilon > 0$, il existe un point $y_0 \in \chi$ et un entier $k > 0$ vérifiant :

$$\delta(x_o, y_o) < \varepsilon \Rightarrow \delta(x_k, y_k) > \beta \qquad (2.1)$$

2.2.2 Transitivité topologique

La fonction f est dite topologiquement transitive :

S'il existe $x \in \chi$ tel que l'orbite $\{f^k(x) \ / \ k \in \mathbb{N}\}$ est dense dans χ. f^k représente la $K^{ième}$ composition de la fonction f.

2.2.3 Densité des orbites périodiques

Soit F et X deux ensembles. F est dense dans X ; si F est inclus dans X et si pour tout point $x \in X$, chaque voisinage de x contient au moins un point de F.

L'ensemble des orbites périodiques $\{ x_o \in \chi; \exists k \rangle 0, x_k = x_o \}$ est dense dans χ.

2.2.4 Ergodicité

L'ergodicité est la propriété dans laquelle les trajectoires suivies par les points appartenant à l'espace des phases se déplacent à travers l'espace avec une distribution uniforme. La trajectoire d'un système chaotique satisfait cette propriété.

On peut dire autrement que l'ergodicité d'un système signifie qu'il parcourt tous les états possibles avec des probabilités égales.

2.2.5 Espace des phases

Les trajectoires dynamiques des systèmes chaotiques sont fréquemment situées dans un espace appelé espace de phase. Les régions de l'espace sans l'existence permanente des dynamiques chaotiques seront inutiles puisque les points dans ces zones tendent vers l'infini et ne contribuent pas à la continuité du processus chaotique. Les variables qui construisent cet espace doivent contenir toute information sur la dynamique du système. On forme alors des équations chaotiques fonctionnant avec ces coordonnées dans l'espace et chaque itération de ces équations signifie l'incrémentation au temps suivant.

2.2.6 Attracteurs étranges

On peut définir un attracteur en étant une limite asymptotique des solutions de toute condition initiale localisée dans un domaine de volume non nul ou bassin d'attraction.

Les trajectoires complexes dans l'espace de phase qui attirent les solutions du système chaotique sont alors des attracteurs. L'ensemble de points attirés vers l'attracteur constitue le bassin d'attraction. Autrement, l'attracteur est une figure géométrique de l'espace de phase indiquant le comportement d'un système chaotique. L'attracteur peut être étrange avec structure fractale ou bien point fixe ou bien cercle limite. Parmi les premiers exemples des attracteurs étranges mentionnés dans l'histoire du chaos, on cite l'attracteur de Lorenz. On donnera par la suite plusieurs exemples d'attracteurs étranges.

2.3 Génération du chaos

Il existe plusieurs systèmes chaotiques qui sont utilisés pour générer les signaux chaotiques. Dans ce paragraphe, nous présenterons deux classes : les systèmes chaotiques continus et les systèmes chaotiques à temps discret.

2.3.1 Systèmes chaotiques continus

- **Attracteur de Lorenz**

Cet exemple a été publié en 1963 dans un journal météorologique. L'attracteur de Lorenz est généré par le système d'équations suivant :

$$\begin{cases} \dot{x} = -\sigma x + \sigma y \\ \dot{y} = \rho x - y - xz \\ \dot{z} = -\beta z + xy \end{cases} \tag{2.2}$$

Les paramètres σ, β et ρ sont des réels strictement positifs.

Le chaos est obtenu pour les valeurs suivantes : $\sigma > \beta+1$; $\rho > 0$ et $\rho > \dfrac{\sigma(\sigma+\beta+3)}{\sigma-\beta-1}$.

La figure 2.1 illustre l'attracteur de Lorenz en 3 dimensions x(t), y(t) et z(t) tel que $\sigma = 10$, $\beta = 8/3$ et $\rho = 28$.

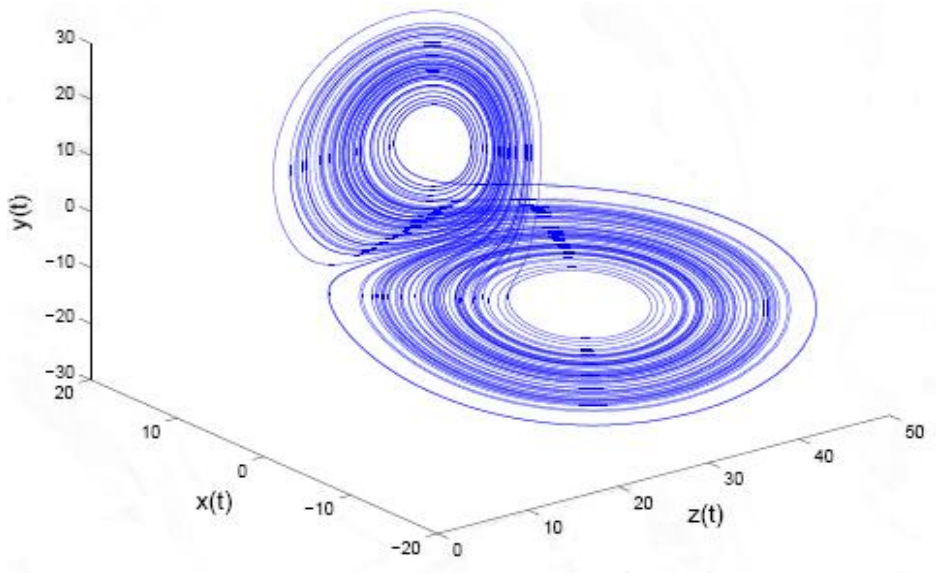


Figure 2. 1. Attracteur de Lorenz.

- **Système de Chen**

Il est donné par le système d'équations suivant :

$$\begin{cases} \dot{x} = a(y - x) \\ \dot{y} = (c - a)x - xz + cy \\ \dot{z} = xy - bz \end{cases} \tag{2.3}$$

La figure 2.2 montre l'attracteur de Chen en 3 dimensions x(t), y(t) et z(t) avec a=35, b=3 et c=28.

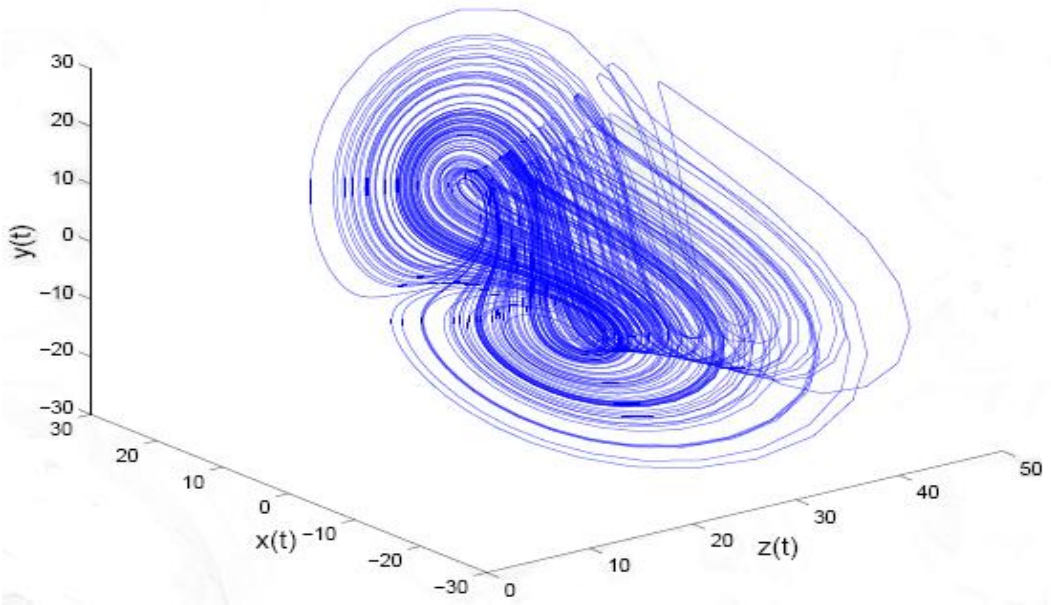


Figure 2. 2. Attracteur de Chen.

- **Système de Rössler**

Il a été inventé par Otto Rössler en 1976. Ce simple système chaotique est présenté comme suit :

$$\begin{cases} \dot{x} = -(y+z) \\ \dot{y} = x + ay \\ \dot{z} = (x-c)z + b \end{cases} \tag{2.4}$$

Avec a = 0.2, b = 0.2 et c = 5.7

La figure 2.3 montre l'attracteur de Rössler en 3 dimensions x(t), y(t) et z(t).

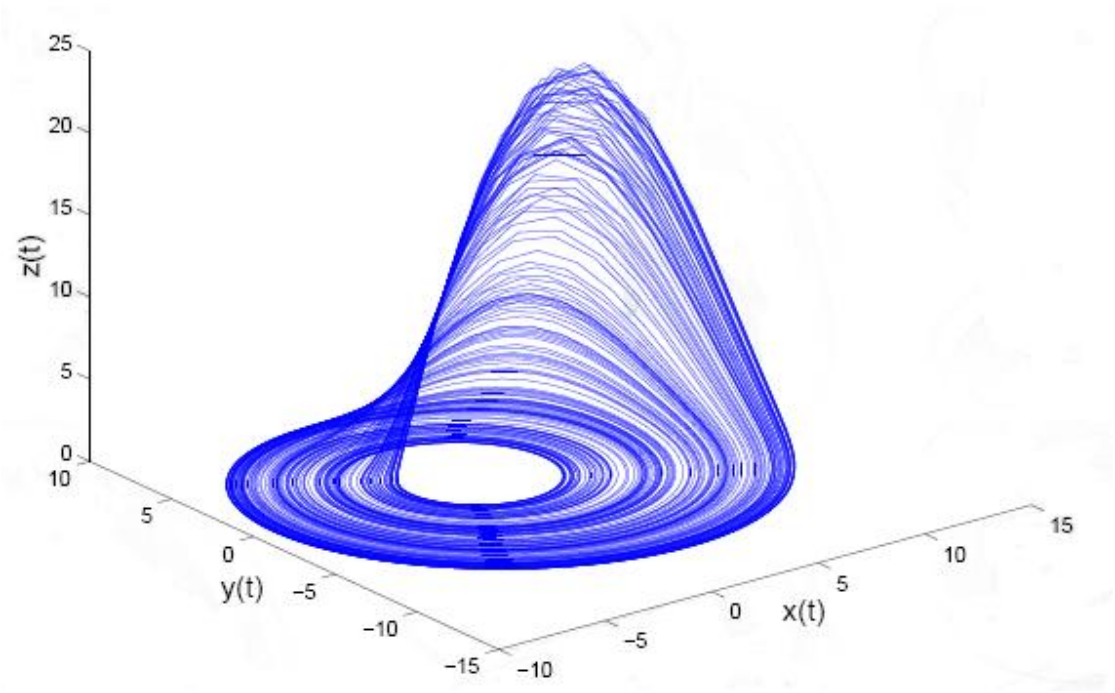


Figure 2. 3. Attracteur de Rössler.

- **Système de Chua**

Il est donné par le système d'équations suivant :

$$\begin{cases} \dot{x} = p(-x+y-f(x)) \\ \quad = p(-x+y-(m_o x + \dfrac{(m_1-m_o)(|x+1|-|x-1|)}{2}) \\ \dot{y} = x-y+z \\ \dot{z} = -qy \end{cases} \tag{2.5}$$

$f(x)$ est la caractéristique non linéaire de la diode du circuit de Chua, avec m_0 et m_1 des constantes négatives. On prend $p = 10$, $m_0 = -0.7$, $m_1 = -1.3$ et $q = 15$. La figure 2.4 représente l'attracteur de Chua.

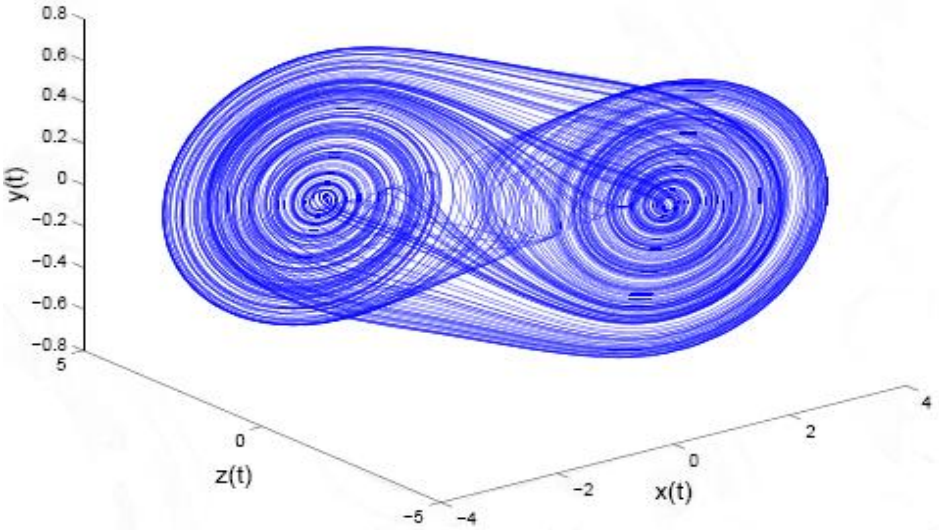


Figure 2. 4. Attracteur de Chua.

2.3.2 Suites chaotiques à temps discret

- **Suite logistique (Logistic Map)**

Cette fonction est donnée par l'équation suivante :

$$X_{n+1} = r\, X_n(1 - X_n) \qquad (2.6)$$

X_n est compris entre 0 et 1 et r est un nombre positif compris entre 1 et 4. Le comportement est chaotique à partir de r égal à 3.6.

La figure 2.5 illustre le diagramme de bifurcation (X_n en fonction de r).

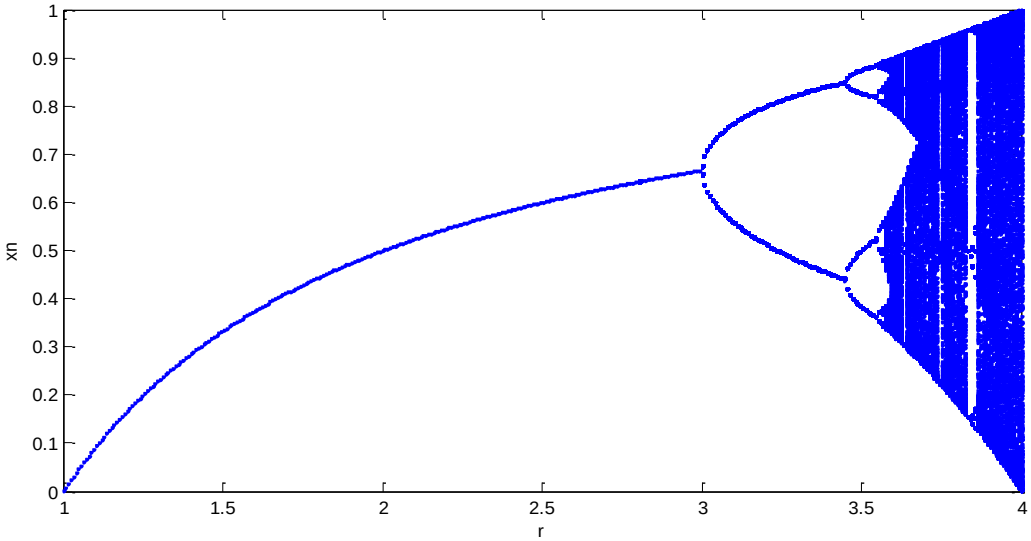


Figure 2. 5. Diagramme de bifurcation.

- **Les suites chaotiques linéaires par morceaux (PWLCM)**

Il existe plusieurs récurrences chaotiques linéaires par morceaux, dont on cite les suivantes :

- *Tent Map*

$$f(x) = \begin{cases} r\,x, & 0 \le x < 0.5 \\ r\,(1-x), & 0.5 \le x \le 1 \end{cases} \tag{2.7}$$

- *Skew tent map*

$$\mathrm{f}(\mathrm{x}) = \begin{cases} \mathrm{x}/\mathrm{r}, & 0 \le \mathrm{x} < \mathrm{r} \\ (1-\mathrm{x})/(1-\mathrm{r}), & \mathrm{r} \le \mathrm{x} < 1 \end{cases} \tag{2.8}$$

- **La récurrence de Hénon**

Elle constitue un système dynamique à temps discret introduit par l'astronome Michel Hénon en 1976. Il est présenté par l'équation suivante :

$$\begin{aligned} x_{n+1} &= y_n + 1 - a x_n^2 \\ y_{n+1} &= b x_n \end{aligned} \tag{2.9}$$

Tel que $(x_n, y_n) \in \mathbb{R}^2$
La figure suivante représente l'attracteur de Hénon avec a=1.4 et b=0.3.

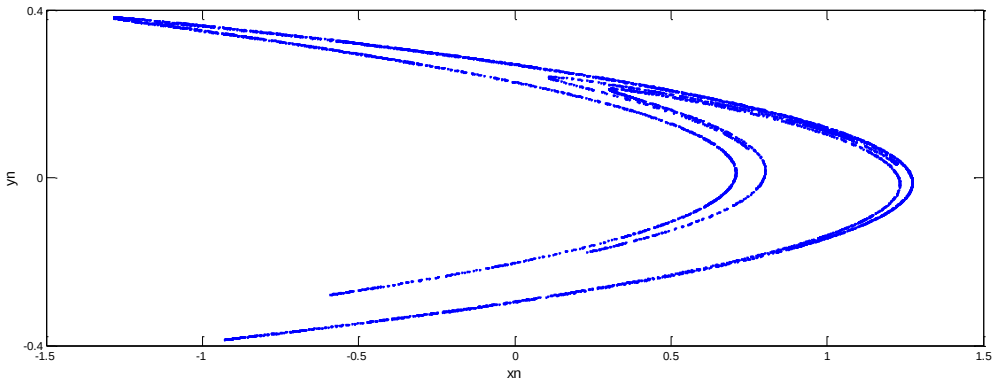


Figure 2. 6. Attracteur de Hénon.

2.4 Les tests du NIST

Les tests du NIST (National Institute of Standards and Technology) forment un paquetage statistique de tests qui sont conçus pour détecter l'aspect aléatoire des séquences binaires à la

sortie des générateurs de nombres aléatoires ou pseudo-aléatoires utilisés dans des applications nécessitant de la cryptographie [19], [20].

La sortie des générateurs de nombre pseudo-aléatoires doit être imprévisible en ignorant l'entrée.

Les tests du NIST se concentrent sur différents types d'aspects non-aléatoires que l'on peut trouver dans une séquence et les comparer avec une séquence aléatoire. Quelques tests sont décomposables en un ensemble de sous-tests.

L'ordre d'application des tests est arbitraire. Cependant, le test de fréquence doit être appliqué en premier lieu, puisqu'il fournit la preuve la plus évidente de l'aspect non aléatoire, qui est la non uniformité. Si le test ne réussit pas, la probabilité d'échec des tests suivants est élevée.

Le résultat de chaque test est donné par une P-Value qui représente la probabilité qu'un générateur de nombre aléatoire parfait produise une séquence moins aléatoire que la séquence déjà testée. Cette variable a une distribution uniforme sur l'intervalle [0 1].

P-Value = 1 : aspect aléatoire parfait.

P-Value = 0 : aspect non aléatoire.

Une constante α est fixée dans l'intervalle [0.001-0.01]. Elle est appelée "niveau de signification". Si les P-Value sont supérieures ou égales à α, alors la séquence réussit le test sinon elle échoue.

On présente par la suite les 15 tests du NIST.

2.4.1 Test statistique

Un test statistique est formulé pour tester une hypothèse nulle spécifique (H_0). L'hypothèse nulle est que la séquence testée est aléatoire. Une hypothèse alternative (H_a) est que la séquence n'est pas aléatoire.

Pour chaque test, une décision ou une conclusion est proposée d'accepter ou de rejeter l'hypothèse nulle. Une statistique appropriée d'aspect aléatoire est choisie et utilisée pour déterminer l'acceptation ou le rejet de l'hypothèse nulle.

Une statistique a une distribution de valeurs possibles. Une distribution théorique de référence est déterminée par des méthodes mathématiques. A partir de cette distribution de référence, une valeur critique est déterminée.

Durant le test, une valeur statistique du test est calculée (sur la séquence testée). Si la valeur dépasse la valeur critique, l'hypothèse nulle pour l'aspect aléatoire n'est alors rejetée. Sinon, l'hypothèse nulle est retenue et acceptée.

Dans la pratique, la raison pour laquelle l'hypothèse de test statistique fonctionne est que la distribution de référence et la valeur critique sont dépendantes et générées avec une supposition d'aspect aléatoire.

Si l'aspect aléatoire est vérifié, la valeur du test statistique calculée aura une faible probabilité (0.01 %) de dépasser la valeur critique.

La probabilité de conclure que la donnée est non aléatoire est fixée avant le test et notée α. Pour cela une séquence peut apparaître non aléatoire même si elle est produite par un bon générateur.

Une autre probabilité β, est la probabilité que le test conclu qu'une séquence est aléatoire alors qu'elle ne l'est pas.

Le test statistique est employé pour calculer une P-value. Chaque P-value est la probabilité qu'un générateur de nombre aléatoire parfait produise une séquence moins aléatoire que la séquence testée.

Une P-value égale à 1 signifie que la séquence est parfaitement aléatoire.

Une P-value égale à 0 signifie que la séquence est non-aléatoire.

Si la P-value ≥ α, alors l'hypothèse nulle est acceptée (i.e., la séquence apparaît aléatoire). Si P-value < α, alors l'hypothèse nulle est rejetée (i.e., la séquence apparaît non aléatoire).

Le niveau de signification α peut être choisi pour les tests. Il est choisi typiquement dans l'intervalle [0.001, 0.01].

- α égale à 0.001 indique qu'une séquence sur 1000 est rejetée par le test si la séquence n'est pas aléatoire. Pour une P-value ≥ 0.001, la séquence peut être considérée comme aléatoire. Pour une P-value < 0.001, une séquence peut être considérée comme non aléatoire.
- α égale à 0.01 indique qu'une séquence sur 100 est rejetée. Une P-value ≥ 0.01 montre que la séquence est aléatoire.

2.4.2 Propriétés d'une séquence aléatoire testée

Les hypothèses suivantes ont été mises en œuvre en ce qui concerne la séquence binaire aléatoire à tester :

1- Uniformité :

L'occurrence de 0 ou de 1 est probablement égale, i.e., la probabilité de chacun est exactement 1/2. Le nombre attendu de 0 ou de 1 est n/2, où n est la longueur de la séquence.

2- Extensibilité :

Tout test applicable à une séquence peut être aussi appliqué à une sous-séquence extraite aléatoirement. Si une séquence est aléatoire, alors toute sous-séquence extraite doit être aléatoire. Ainsi, toute sous-séquence doit passer tout test de l'aspect aléatoire.

3- Cohérence :

Le comportement d'un générateur doit être constant à travers les valeurs initiales (seeds).
Il est inadéquat de tester un PRNG basé sur une sortie d'un seul seed, ou un RNG basé sur une sortie produite d'une seule sortie physique.

2.4.3 Test de fréquence

a) But du test

Le but de ce test est de déterminer si le nombre de 0 et de 1 dans une séquence est approximativement le même comme il est prévu pour une séquence réellement aléatoire. Le test vérifie si la fraction des 1 est proche de 1/2.

b) Description du test

1) Conversion en ± 1 : les 0 et les 1 de la séquence ε sont convertis respectivement en 1 et -1. On aura $S_n = X_1+X_2+... + X_n$, tel que $X_i = 2^{\varepsilon_i} -1$.
2) Calculer la statistique du test $S_{obs} = \frac{|S_n|}{\sqrt{n}}$.
3) Calculer P-Value = erfc($\frac{S_{obs}}{\sqrt{2}}$), avec erfc est la fonction d'erreur complémentaire.
4) Si la P-Value < 0.01, alors la séquence est non-aléatoire. Sinon elle est aléatoire. Il est recommandé que la séquence testée soitd'une longueur minimale de 100 bits ($n \geq 100$). Avec n est la longueur de la chaîne de bits.
   ε est la séquence de bits du générateur RNG ou PRNG à tester tel que $\varepsilon = \varepsilon_1, \varepsilon_2, \varepsilon_3, ..., \varepsilon_n$ et S_{obs} est la valeur absolue de la somme des X_i, (avec $X_i = 2\,\varepsilon - 1 = \pm 1$)

2.4.4 Test de fréquence par bloc

Le but de ce test est de déterminer si la fréquence des 1 dans un bloc de M bits est approximativement 1/2. Pour un bloc de taille M = 1, on revient au test de fréquence.

1) Partager la séquence en N séquences, telle que $N=\left\lfloor \frac{n}{M} \right\rfloor$ et enlever les bits inutilisés.
2) Déterminer la proportion π_i des 1 dans chaque séquence de M-Bits en utilisant l'équation : $\pi_i = \frac{\sum_{j=1}^{M} \varepsilon_{(i-1)M+j}}{M}$ pour $1 \leq i \leq N$
3) Calculer la distribution χ^2, $\chi^2(\text{obs}) = 4M\sum_{i=1}^{N}(\pi_i - 1/2)^2$
4) Calculer P-value=igamc(N/2, χ^2(obs)/2), tel que "igamc" est la fonction de gamma incomplète.

Il est recommandé que chaque séquence à tester ait une longueur minimale égale à 100 bits ($n \geq 100$), tel que ($n \geq M.N$). $M \geq 20$, $M > 0.01\,n$ et $N < 100$.

2.4.5 Test de somme cumulative (inverse)

Le but de ce test est de déterminer si la somme cumulative dans une séquence est trop grande ou trop petite (somme de 1 et -1). Ceci indique la présence de nombre important de 0 ou de 1. La somme cumulative peut être considérée comme une marche au hasard (random walk) qui est un modèle mathématique d'un système possédant une dynamique discrète composée d'une succession de pas aléatoires, ou effectuée « au hasard ».

Pour une séquence aléatoire, les excursions du "random walk" doivent être proches de 0.

1) Former une séquence normalisée, ε est transformée en X_i, (avec $X_i=2\varepsilon-1=\pm 1$).

2) Calculer les sommes partielles des sous séquences S_k, de largeurs successives, tel que $S_k = S_{k-1}+X_k$ (mode 0) ; $S_k = S_{k-1}+X_{n-k+1}$(mode 1).
3) Calculer la statistique du test : $z = \max_{1\le k\le n} |S_k|$.
4) Calculer la P-Value.

$$K_1 = \sum_{k=\left(\frac{-n}{2}+1\right)/4}^{\frac{n}{2}-1} \left[\Phi\left(\frac{(4k+1)z}{\sqrt{n}}\right) - \Phi\left(\frac{(4k-1)z}{\sqrt{n}}\right)\right] ; \qquad (2.10)$$

P-value = 1- K_1 + K_2;

$$K_2 = \sum_{k=(\frac{-n}{2}-3)/4}^{(\frac{n}{2}-1)/4} \Phi\left(\frac{(4K+3)z}{\sqrt{n}}\right) - \Phi\left(\frac{(4k+1)z}{\sqrt{n}}\right) \qquad (2.11)$$

P-value = 1- K_1 + K_2;

Telle que Φ est la fonction de distribution cumulative (normal standard).

2.4.6 Test 4 (Test de série)

Le « Runs Test » permet de déceler des oscillations entre les 0 et les 1 trop rapides ou trop lentes. Pour cela, il faut :

1) Calculer π, tel que : $\pi = \frac{\sum_j \varepsilon_j}{n}$.
2) Calculer $V_n(obs) = \sum_{k=1}^{n-1} r(k)+1$, r(k) = 0 si $\varepsilon_k = \varepsilon_{k+1}$ sinon r(k) = 1.
3) Calculer P-Value=erfc($\frac{|V_n(obs) - 2n\pi(1-\pi)|}{2\sqrt{2n}\pi(1-\pi)}$).

2.4.7 Test 5 (Test de longues séries de 1)

Ce test consiste à déterminer si la distribution de longues séries de 1 est conforme avec les probabilités théoriques.

1) Diviser la séquence en M blocs. La longueur de chaque bloc doit être conforme au tableau 2.1.

Tableau 2. 1. Division de la séquence en M blocs.

M	**Minimum n**
8	128
128	6272
10^4	750.000

2) Classer la fréquence de la plus grande série de 1 dans chaque séquence dans des catégories selon le tableau 2.2.

Tableau 2. 2. Classement de la fréquence.

V_i	**M=8**		**M=128**		**M=10000**	
	Longueur block	π_i	**Longueur block**	π_i	**Longueur block**	π_i
V_0	≤ 1	0.2148	≤ 4	0.1174	≤ 10	0.0882
V_1	2	0.3672	5	0.2430	11	0.2092
V_2	3	0.2305	6	0.2493	12	0.2483
V_3	≥ 4	0.1875	7	0.1752	13	0.1933
V_4			8	0.1027	14	0.1208
V_5			≥ 9	0.1124	15	0.0675
V_6					≥ 16	0.0727

3) Calculer $X^2(obs)= \sum_{i=0}^{K} \frac{(v_i - N\pi_i)^2}{N\pi_i}$
4) Calculer P-value=igamc(K/2, X^2(obs)/2)

2.4.8 Test 6 (Test de rang)

Calculer le rang des sous matrices de la séquence et vérifier leur dépendance linéaire.

1) On divise la séquence en $N = \left\lfloor \frac{n}{M^2} \right\rfloor$ sous-séquences de blocs disjoints de longueur M^2 afin de construire la matrice carrée M*M notée.
2) Déterminer le rang R_l de chaque matrice, avec l = 1,.., N.
3) Calculer :

$$\chi^2(obs) = \frac{(F_M - 0.2888N)^2}{0.2888N} + \frac{(F_{M-1} - 0.5776N)^2}{0.5776N} + \frac{(N - F_M - F_{M-1} - 0.1336N)^2}{0.1336N} \quad (2.12)$$

Tel que F_k est le nombre de matrices de rang égal à k.

4) Calculer la P-Value = $e^{-X(obs)^2/2}$

2.4.9 Test 7 (Transformée de Fourier discrète)

Ce test tient compte des hauteurs des pics de la transformée de Fourier de la séquence pour détecter une périodicité.

L'intention est de détecter si le nombre de pics dépassant le seuil de 95 % est largement différent de 5 %. La P-Value sera égale à : $\operatorname{erfc}\left(\frac{|d|}{\sqrt{2}}\right)$

2.4.10 Test 8 (Non overlapping template Matching)

Ce test consiste à détecter des générateurs qui produisent trop d'occurrence d'un mot apériodique donné (template).
Une fenêtre de m-bits est utilisée. Si le mot n'est pas trouvé, la fenêtre est décalée d'un bit. Si le mot est trouvé, la fenêtre décale jusqu'au bit qui suit le mot trouvé.

1) Calculer les grandeurs suivantes :
   $\mu = \frac{M-m+1}{2^m}$ et $\sigma^2 = M\left(\frac{1}{2^m} - \frac{2m-1}{2^{2m}}\right)$
   Tel que M est la longueur en bits de la séquence à tester et m est la longueur en bits du template.
2) Calculer $\chi^2(\text{obs}) = \sum_{j=1}^{N} \frac{(W_j - \mu)^2}{\sigma^2}$
3) Calculer Pvalue = igamc(N/2, χ^2/2).

2.4.11 Test 9 (overlapping template Matching)

Le but de ce test est identique à celui du 8ème test, calculer le nombre d'occurrences de B dans chacun des N blocs. On crée une fenêtre de m bits qui traverse la séquence en comparant les bits de la fenêtre avec B. Un compteur s'incrémente quand il y a une égalité.

Après chaque test, la fenêtre est décalée de 1 bit.

Le nombre d'occurrences dans chaque bloc est enregistré en incrémentant un vecteur V_i.

V_0 est incrémenté quand il n'y a pas d'égalité.

V_1 est incrémenté pour une égalité dans le bloc.

1) Calculer $\chi^2(\text{obs}) = \sum_{j=0}^{K} \frac{(v_j - N\pi_j)^2}{N\pi_j}$ avec π_j est obtenue par la formule suivante :

$$\begin{cases} \pi_i = P(U=i) = \frac{e^{-n}}{2^i} \sum_{l=1}^{i} \binom{i-1}{l-1} \frac{n^l}{l!} & 0 \le i \le K-1; \\ \pi_k = 1 - \sum_{k=0}^{K-1} \pi_k & i = K; \end{cases} \tag{2.13}$$

2) Calculer P-Value= igamc (5/2, χ^2/2).

2.4.12 Test statistique universel : Test de Maurer

Le but de ce test est de déterminer si la séquence est compressible ou non sans perte d'information. Une séquence nettement compressible est considérée comme non aléatoire.

La séquence de bits est divisée en deux sous séquences : la première est un segment d'initialisation de $Q \times L$ bits non chevauchés. La deuxième est un segment de test de $K \times L$ bits non chevauchés (figure 2.7).

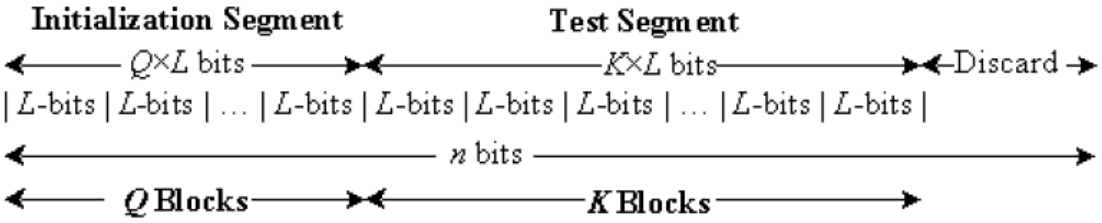


Figure 2. 7. Les sous séquences Q et K [19].

Calculer la P-Value : P-Value = erfc($\frac{f_n - \text{expectedValue}(L)}{\sqrt{2}\sigma}$).

Avec : $f_n = \frac{1}{K}\sum_{i=Q+1}^{Q+K} \log_2(i - T_j)$, tel que T_j est la représentation décimale du contenu du $i^{ème}$ L blocs. $T_{j=i}$ est la position du bloc L.

Avec $\sigma = \sqrt[c]{\frac{\text{variance}(L)}{K}}$ et $c = 0.7 - \frac{0.8}{L} + \left(4 + \frac{32}{L}\right)\frac{K^{\frac{-3}{L}}}{15}$

2.4.13 Test d'entropie approximative

On s'intéresse aux fréquences d'occurrences de toutes les sous-séquences possibles de longueur m fixée. Nous allons comparer les fréquences obtenues avec les longueurs m et m+1. L'entropie mesure le degré de désordre d'un système.
Pour une séquence de bits donnée, il faut ajouter les m-1 bits de la fin de la séquence à son début.

Exemple : 001101 et m=3, on obtient : 00110100.
Les blocs de bits chevauchés de longueur m sont testés : 001, 011, 110, 101, 010, 100.

1) La fréquence des blocs est comptée. $V^i_m 0 \le i \le 2^m$
V_{001}=1, V_{011}=1, V_{110}=1, V_{101}=1, V_{010}=1, V_{100}=1

2) Calculer $C^m{}_i = \frac{\#i}{n}$

3) Calculer $\varphi^{(m)} = \sum_{i=0}^{2^m-1} \pi_i \log \pi_i$, tel que $\pi_i = C^m{}_j$ et $j=\log_2(i)$

4) Répéter les étapes en remplaçant m par m+1

5) Calculer P-Value = igamc (2m-1, $\chi^2/2$), tel que $\chi^2 = 2n[\log 2 - ApEn(m)]$ et $ApEn(m) = \varphi^{(m)} - \varphi^{(m+1)}$

2.4.14 Random excursion

Un cycle d'une marche aléatoire (excursion) est une séquence de pas aléatoires qui commence et finit à son origine (figure 2.8).

On a recours à déterminer si le nombre de visites à un état particulier d'un cycle dévie de ce qui est attendu.

Ce test est une série de 8 tests et conclusions. Un test est une conclusion pour chaque état : -4, -3, -2, -1 et +1, +2, +3, +4 ;

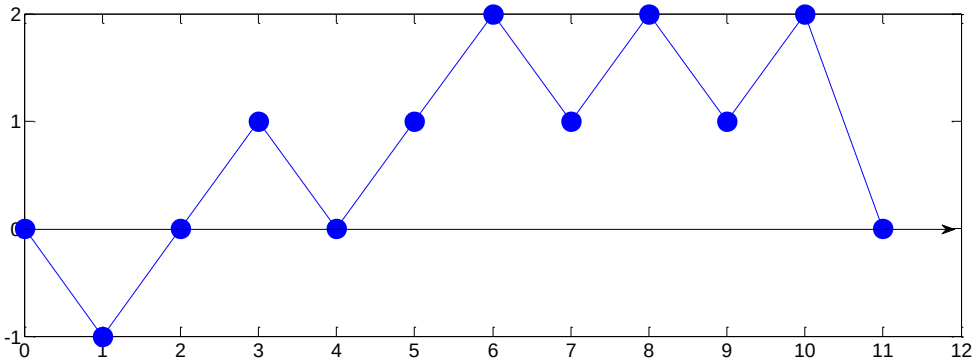


Figure 2. 8. Exemple de marche aléatoire.

1) Calculer P-Value:P-Value=igamc (5/2, $\chi^2/2$), tel que $\chi^2 = \sum_{k=0}^{5} \frac{(v_k(x) - J\pi_k(x))^2}{J\pi_k(x)}$

2.4.15 Random excursion variant

Le but de ce test est de calculer le nombre de fois où un état particulier est visité, et de détecter la déviation par rapport au nombre de visites attendu à différents états de la marche aléatoire.

Ce test est actuellement une série de tests et de conclusion, un test et une conclusion pour chaque état : -9,-8,…,-1. Et +1, +2,…, +9.

On doit convertir les 0 et les 1 de la séquence ε en -1 et 1. La nouvelle séquence sera :
$X = X_1, X_2,..., X_n$, tel que $X_i = 2\varepsilon_i - 1$.

1) Calculer la somme partielle S_i, tel que $S_1 = X_1$, $S_2 = X_1 + X_2$, $S_3 = X_1 + X_2 + X_3$, $S_n = X_1 + X_2 + \ldots + X_k + \ldots + X_n$.
   On obtient $S=\{S_i\}$.
2) On forme ensuite une nouvelle séquence S', en ajoutant un zéro au début et à la fin de l'ensemble S.

3) Calculer pour les 18 états, $\xi(x)$ qui est le nombre total des fois où m'états x est vérifié tout au long des cycles J.
4) Calculer la P-Value tel que :

$$P-\text{value} = \text{erfc}\left(\frac{|\xi(x)-J|}{\sqrt{(2J(4*|x|-2)}}\right) \tag{2.14}$$

2.4.16 Serial Test

Ce test est basé sur la fréquence de tous les m-bits de chevauchement tout au long de la séquence. Le but de ce test est de déterminer si le nombre d'occurrences des 2^m des modèles de chevauchement des m bits est identique à celui d'une séquence aléatoire (m est le nombre de bits dans chaque bloc).

Une séquence est aléatoire tel que chaque modèle de m-bits a la même chance d'apparence que d'autre m-bits. Pour m = 1, le test de série est équivalent au test de fréquence.

1) On commence par ajouter les m-1 bits au début de la séquence. Déterminer la fréquence de tout bloc de longueur m-bits, m-1 bits, et m-2 bits.
2) Calculer $\psi^2{}_m$, $\psi^2{}_{m-1}, \psi^2{}_{m-2}$ tel que :

$$\begin{cases} \psi^2{}_m = \frac{2^m}{n}\sum_{i_1..i_m}(v_{i_1...i_m} - \frac{n}{2^m})^2 = \frac{2^m}{n}\sum_{i_1...i_m} v_{i_1...i_m} - n \\ \psi^2{}_{m-1} = \frac{2^{m-1}}{n}\sum_{i_1..i_{m-1}}(v_{i_1...i_{m-1}} - \frac{n}{2^{m-1}})^2 = \frac{2^{m-1}}{n}\sum_{i_1...i_{m-1}} v_{i_1...i_{m-1}} - n \\ \psi^2{}_{m-2} = \frac{2^{m-2}}{n}\sum_{i_1..i_{m-2}}(v_{i_1...i_{m-2}} - \frac{n}{2^{m-2}})^2 = \frac{2^{m-2}}{n}\sum_{i_1...i_{m-2}} v_{i_1...i_{m-2}} - n \end{cases} \tag{2.15}$$

3) Calculer

$$\begin{cases} \nabla\psi^2{}_m = \psi^2{}_m - \psi^2{}_{m-1} \\ \nabla^2\psi^2{}_m = \psi^2{}_m - 2\psi^2{}_{m-1} + \psi^2{}_{m-2} \end{cases} \tag{2.16}$$

4) Calculer P-Value :

$$\begin{cases} \text{P-value1=igamc}(2^{m-2}, \nabla\psi^2{}_m) \\ \text{P-value2=igamc}(2^{m-3}, \nabla^2\psi^2{}_m) \end{cases} \tag{2.17}$$

2.4.17 Linear complexity

Ce test est basé sur la longueur d'un registre à décalage à rétroaction linéaire. Le but de ce test est de déterminer si la séquence est assez complexe pour être considérée comme aléatoire.
Les séquences aléatoires sont caractérisées par de long LFSR. Un LFSR trop court implique l'aspect non aléatoire.

1) Partitionner la séquence en N blocs indépendants de M bits chacun, tel que n = MN.
2) En utilisant l'algorithme de Berlekamp Massey, on détermine la complexité L_i des N blocs (i = 1…N). L_i désigne la longueur de la plus courte séquence du LFSR qui génère tous les bits du bloc i.
3) Calculer : $\chi^2(\text{obs}) = \sum_{i=0}^{K} \frac{(v_i - N\pi_i)^2}{N\pi_i}$
4) La P-value est donnée par la formule suivante : P-value=igamc($\frac{K}{2}, \frac{\chi^2(\text{obs})}{2}$)

Remarque : le test qui consomme le plus de temps est le test de complexité linéaire.
Le test numéro 15: Lempel zip compexity est éliminé et n'est plus utilisé dans le paquet des tests.

2.4.18 Interprétations des résultats

Afin de tester les suites chaotiques, la première étape consiste à convertir la sortie réelle de chaque suite en une séquence binaire.

Dans ce qui suit, on utilise 2 représentations binaires :

1) Une représentation issue du standard IEEE 754 simple précision (32 bits) :
   1 bit de signe, 8 bits d'exposant (-126 à 127), 23 bits de mantisse.
2) Arrondir chaque réel à l'entier le plus proche : r < 0.5 le réel prend la valeur 0, r > 0.5 le réel prend la valeur 1.

Deuxièmement, générer m séquences de longueur 1 million de bits chacune.

La 3ème étape consiste à appliquer les tests de NIST sur chaque séquence ; et finalement examiner les P-Value générées.

Nous avons choisi d'appliquer les tests du NIST sur trois suites chaotiques. Nous les avons classées suivant leurs dimensions.

a) Suite 1D

Dans le cas des suites chaotiques à une seule dimension, notre choix est la PWLCM qui est donnée par la formule suivante :

$$X(t+1) = F_p(X(t)) = \begin{cases} X(t)/p & 0 \le X(t) < p \\ (X(t)-p)/(0.5-p) & p \le X(t) < 0.5 \\ (1-X(t)-p)/(0.5-p) & 0.5 \le X(t) < 1-p \\ (1-X(t))/p & 1-p \le X(t) \le 1 \end{cases} \quad (2.18)$$

Le graphe de la figure 2.9 indique le diagramme de phase de la suite PWLCM, avec p = 0.15.

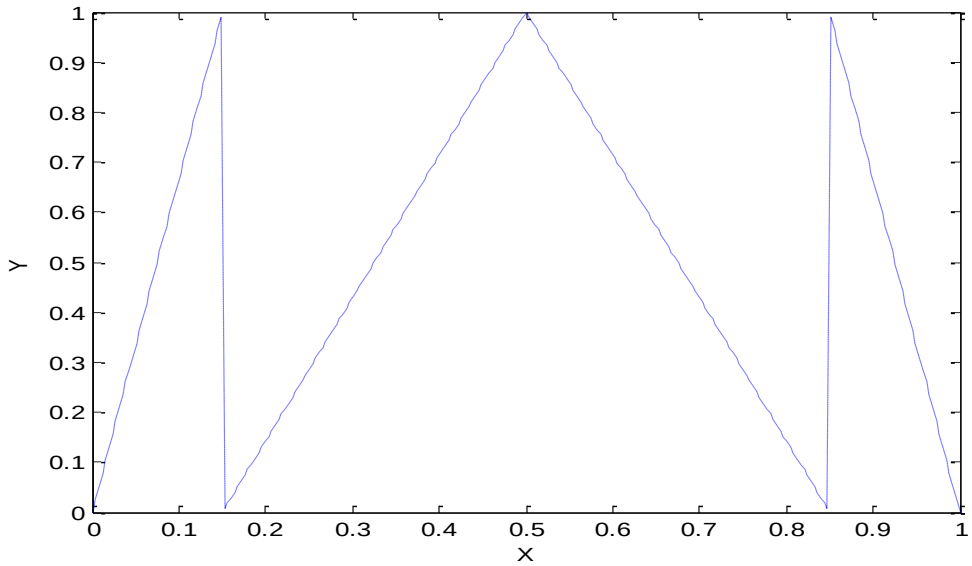


Figure 2. 9. Diagramme de phase de la suite 1D.

- *Conversion avec round*

15 tests ont été appliqués sur 100 séquences binaires de longueur 10^6 bits chacune. La génération de ces séquences était faite en variant la condition initiale (100 conditions initiales). La conversion en binaire est assurée par la fonction round.

La figure 2.10 correspond à la proportion des séquences ayant passées les tests, pour la suite 1D sur un intervalle [0.5 1] de conditions initiales et pour une séquence de longueur minimale 10^6 bits (50 séquences). En moyenne, la proportion des séquences qui ont réussi les tests est 0.94 ; ce qui correspond à 47 séquences parmi 50 testées.

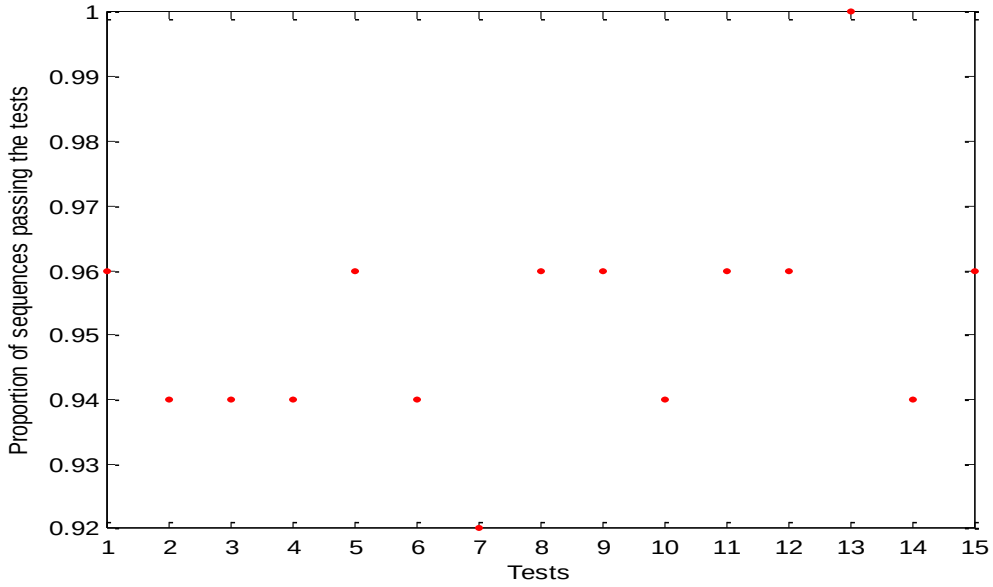


Figure 2. 10. Proportion des séquences ayant passées les tests en fonction des 15 Tests.

La figure 2.11 montre la variation des P-Values pour le premier test en fonction des conditions initiales de la suite 1D pour une valeur p = 0.15.
La ligne rouge est le seuil minimal de réussite alpha = 0.01. Six valeurs parmi 100 échouent le test de fréquence donc elles ne doivent pas être utilisées x1= {0.13, 0.15, 0.17, 0.5, 0.87, 1}.

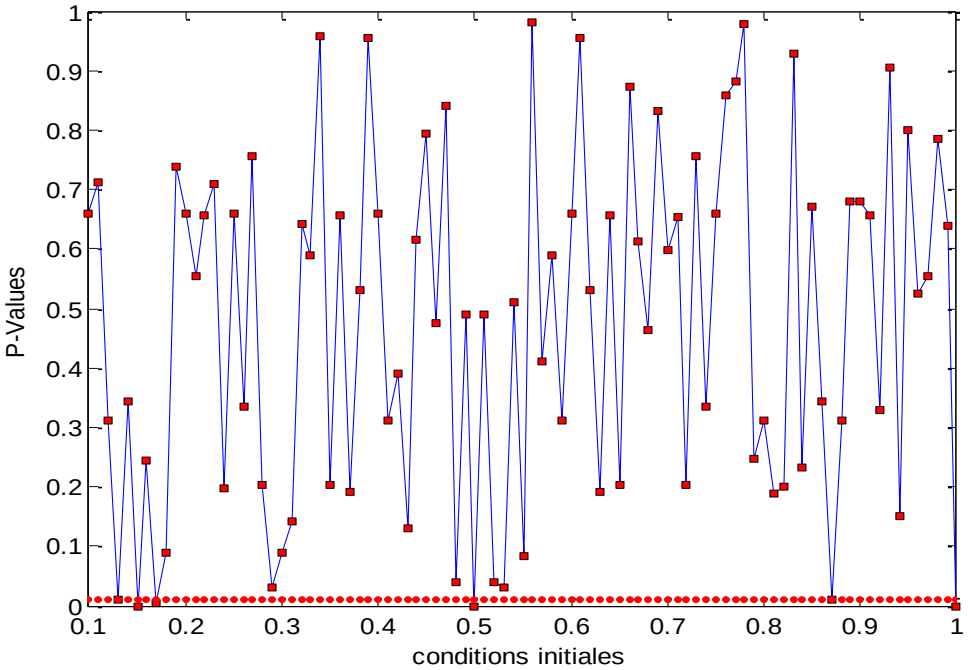


Figure 2. 11. P-Value du test de fréquence en fonction des conditions initiales.

En évitant les conditions initiales citées en dessus, on obtiendra un générateur de nombre pseudo-aléatoire idéal pour la construction des S-Box. En effet cette suite pourra être utilisée comme générateur de conditions initiales parfaitement aléatoire pour la suite 3D.

- *Représentation IEEE 754 simple précision (32 bits)*

Plusieurs essais ont été effectués pour choisir le nombre de bits à garder pour ne pas perdre l'aspect aléatoire de la suite. L'élimination du bit de signe avec les bits d'exposant était évidente. Les tests sont réalisés sur 160 séquences de 1 million de bits chacune. Les réels sont convertis en binaire en gardant seulement 16 bits. Nous avons fait varier les conditions initiales de [0.01 :0.01 :1]. La figure 2.12 montre la variation de la proportion des séquences ayant passé le test. Pour les conditions initiales citées, seulement le test 14 échoue. Pour le reste des tests, les proportions varient entre 0.8875 et 1.

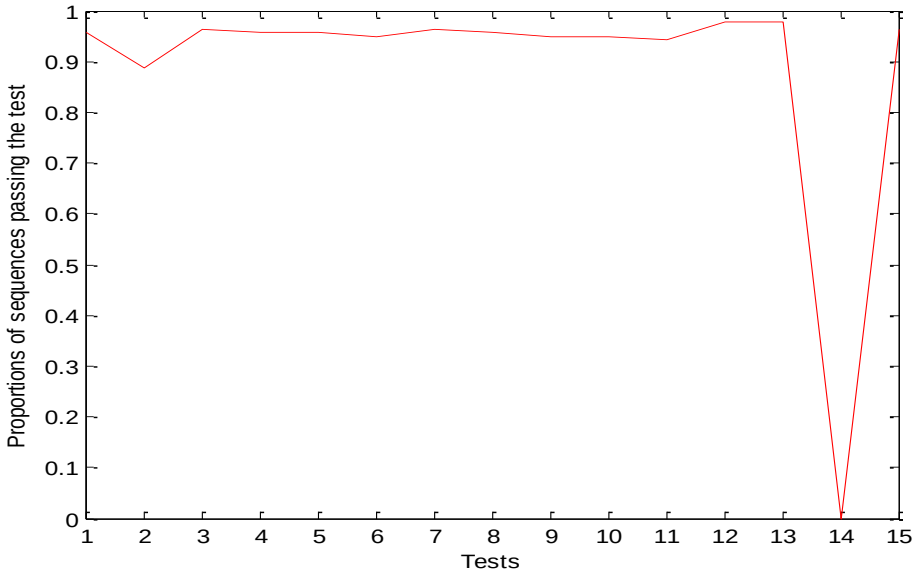


Figure 2. 12. Variation des proportions des séquences qui ont réussi en fonction des tests.

Une conclusion sur l'aspect aléatoire de la récurrence chaotique ne peut pas être confirmée. Nous passons alors à la deuxième méthode qui est l'étude de l'uniformité des séquences. La figure 2.13 représente les P-Values calculées à partir de l'application Chi-Square pour chaque test. Cette application a pour but d'évaluer l'uniformité des P-value au $i^{ème}$ test statistique.

Les séquences n'ont pas une distribution uniforme. Pour cette représentation binaire, l'aspect aléatoire et uniforme des séquences binaires est perdu. On garde alors la première représentation.

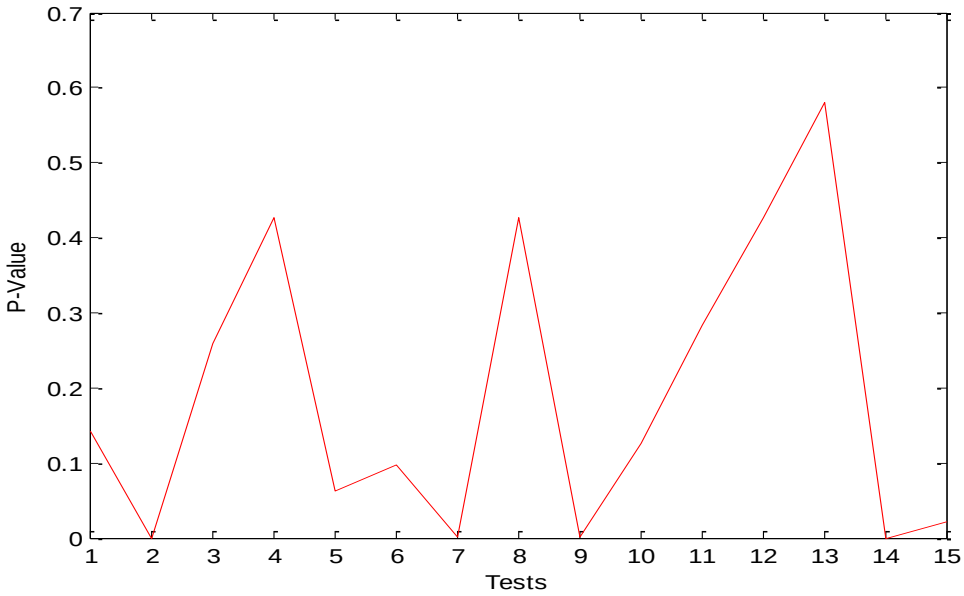


Figure 2. 13. P-Value calculée par Chi-Square en fonction des tests.

b) Suite 2D

La suite à deux dimensions est donnée par l'équation suivante [21]:

$$G_{\alpha,\beta}(x,z) := (|1-|\alpha x+(3-\alpha)y-1||, |1-|\beta y+(3-\beta)y-1||) \qquad (2.19)$$

Tel que $G_{\alpha,\beta} \in [0,1] \times [0,1]$ et $(\alpha, \beta) \in [0,3] \times [0,3]$

- *Conversion avec Round*

La figure 2.14 illustre les proportions des séquences ayant réussi les tests qui ont été effectués sur un ensemble de 210 séquences de 1 million de bits chacune. Les conditions initiales varient selon les intervalles suivant : x_1= [0.1 :0.01 :0.79] et y_1= [0.29 :0.01 :0.99]

Les résultats montrent que ce type de représentation binaire n'est pas adéquat pour cette suite chaotique. L'aspect aléatoire est perdu pour les tests (5, 6, 7, 8, 9, 10, 11, 12, 14 et 15).

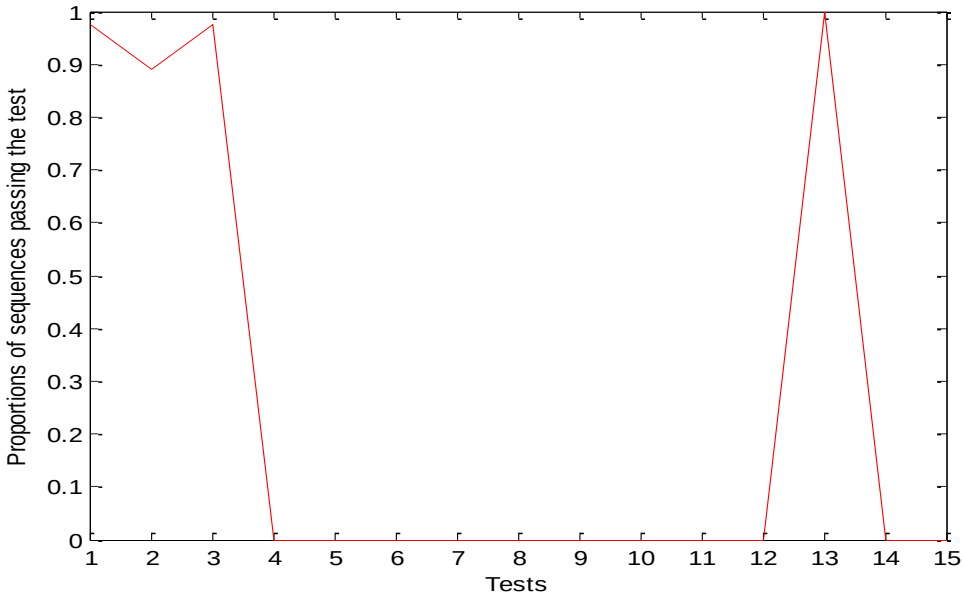


Figure 2. 14. Variation des proportions des séquences qui ont réussi en fonction des tests.

c) Suite 3D

Pour les suites chaotiques en 3 dimensions, nous avons choisi de tester une suite définie sur le cube unité [0; 1] × [0; 1] × [0; 1]. Cette suite est de la forme suivante [22] :

$$\begin{cases} x(n) = |1-|(2.x(n-1)+y(n-1)+z(n-1)/2-1|| \\ y(n) = |1-|(2.y(n-1)+x(n-1)+z(n-1)/2-1|| \\ z(n) = |1-|(2.z(n-1)+y(n-1)+x(n-1)/2-1|| \end{cases} \qquad (2.20)$$

Avec x_0, y_0, $z_0 \in]0,1[$ sont les 3 conditions initiales de la suite.

L'attracteur 3 dimensions est représenté par la figure 2.15.

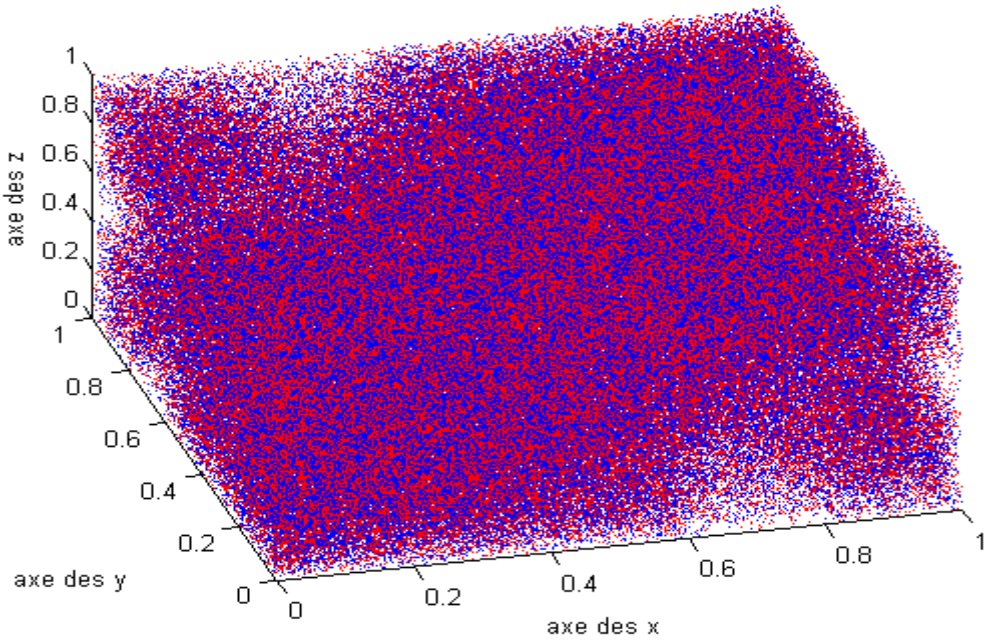


Figure 2. 15. Attracteur 3D.

Ces simulations (tests du NIST) ont été faites pour des séquences de longueur totale égale à 1 million de bits.

Afin de tester l'aspect aléatoire de cette suite, nous avons effectué la conversion du réel au binaire en utilisant la troncature et la représentation IEEE 754 simple précision (32 bits).

- *Conversion avec simple précision*

Les tests sont réalisés sur 120 séquences de 1 million de bits chacune. Les réels sont convertis en binaire en gardant seulement 16 bits y compris le bit de signe et les bits d'exposant pour ne pas perdre l'aspect aléatoire de la suite. Les conditions initiales varient selon les intervalles suivants : x_1= [0.1 : 0.01 : 0.49], y_1 = [0.3 : 0.01 :0.69] et z_1 = [0.45 : 0.01 : 0.84].

La figure 2.16 montre la variation de la proportion des séquences ayant passé le test. Les proportions varient de 0.95 pour le Serial test jusqu'à 1 pour le Universal test.

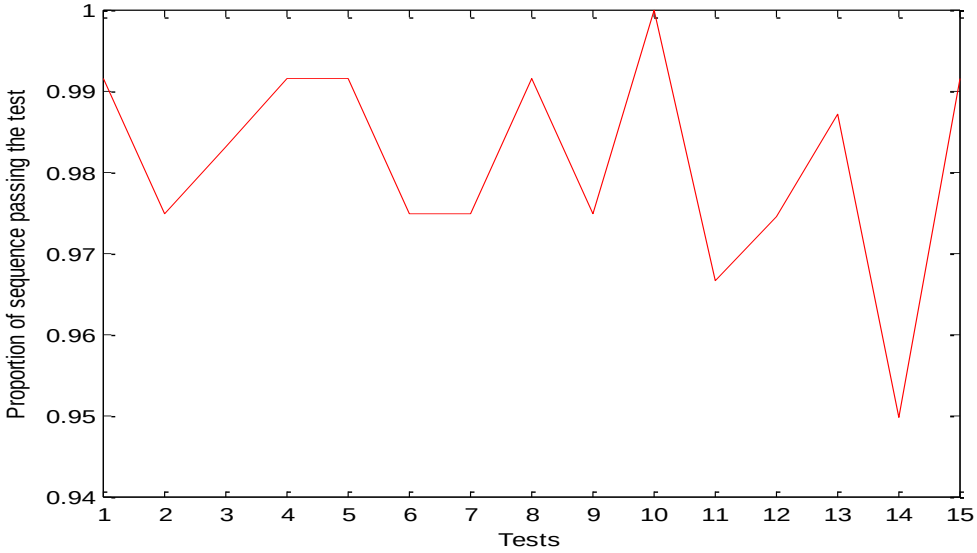


Figure 2. 16. Proportions des séquences passant les tests en fonction des tests.

La Figure 2.17 représente les P-Values calculées à partir de l'application Chi-Square pour chaque test. Ces valeurs varient de 0.001017 pour le test FFT jusqu'à 0.922036 pour le test de rang.

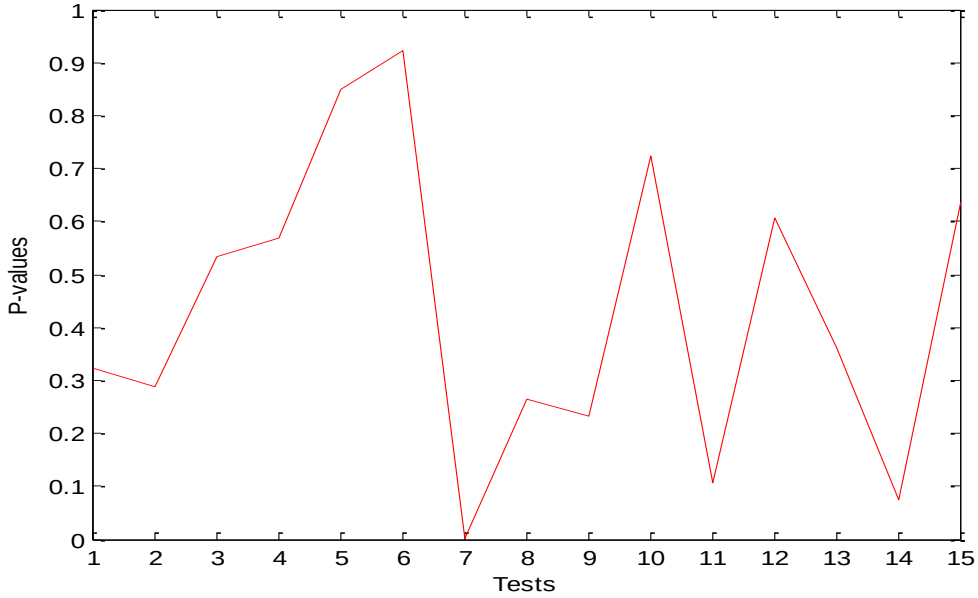


Figure 2. 17. P-Values calculées par Chi-Square en fonction des tests.

La Figure 2.18 représente les P-Values calculées pour chacune des 120 séquences testées et ceci pour le premier test de fréquence. Ces valeurs varient entre 0.0018 et 0.9936. Une seule séquence parmi les 120 échoue ce premier test.

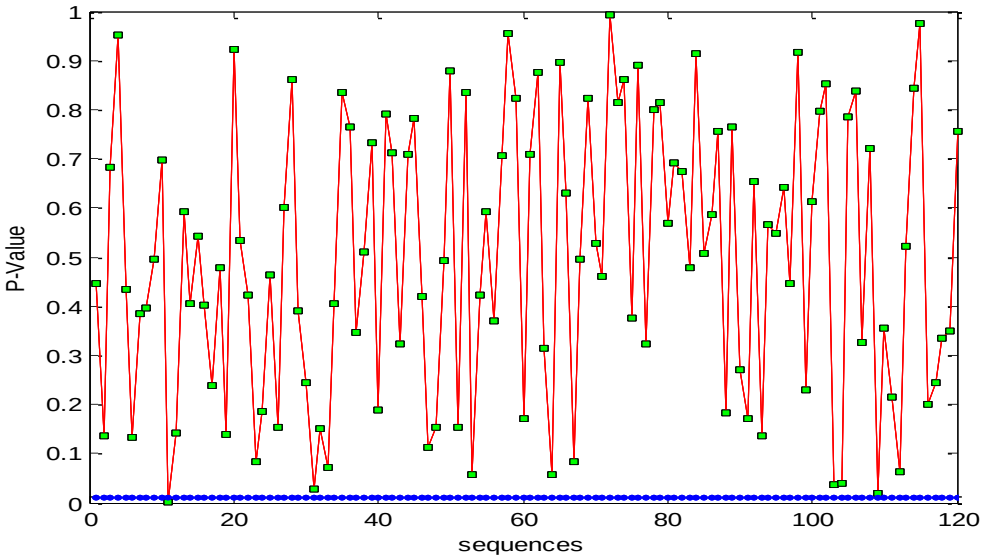


Figure 2. 18. P-Values du test de fréquence en fonction des séquences testées.

La proportion des séquences, ainsi que les valeurs des P-Value théoriques calculées à partir de la fonction Chi-square, montre que pour cette représentation binaire de la suite 3D, il n'y a pas une déviation par rapport à l'aspect aléatoire.

- *Conversion avec Round*

Les P-values de la figure 2.19 varient entre 0.1822 et 0.9713. Toutes les séquences testées réussissent alors le premier test. Avec cette fonction, toutes les séquences (40 séquences de 1 millions de bits chacune) ont échoué 8 tests. La proportion des séquences, qui ont passé le test, est donnée par la figure 2.20. La distribution des séquences n'a pas donné un résultat contradictoire. Les P-values sont uniformément distribuées pour le test de fréquence et le test de complexité linéaire seulement.

Cette représentation binaire de la suite semble non parfaite. On remarque qu'il n'y a aucune sous séquence de la séquence binaire générée ayant passée les tests 5, 7, 8, 9, 10, 11, 14. La méthode de troncature est rejetée pour cette suite à 3 dimensions.

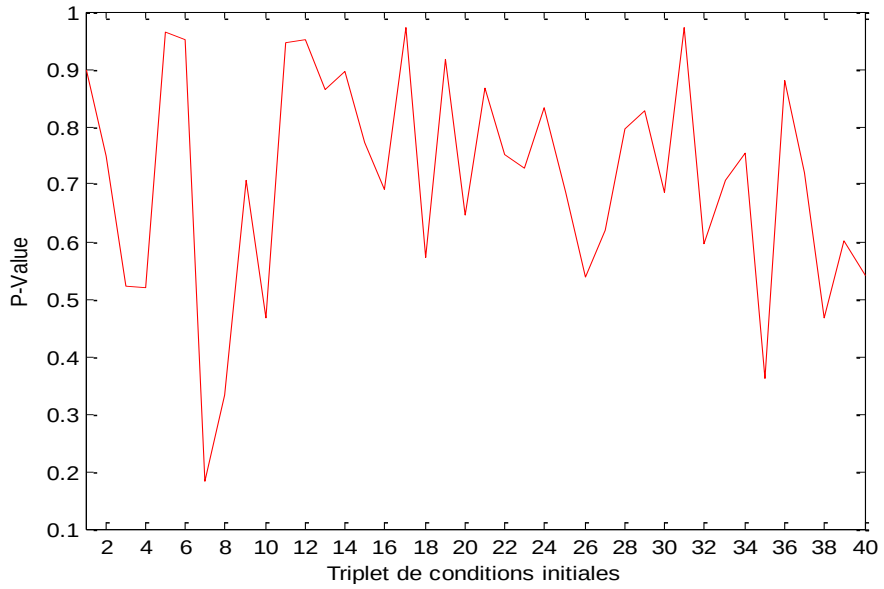


Figure 2. 19. Test de fréquence avec 40 triplets de conditions initiales ; les intervalles sont les suivant : x_1 = [0.1 : 0.01 : 0.5], x_2 = [0.3 : 0.01 : 0.7], z_1 = [0.45 : 0.01 : 0.85].

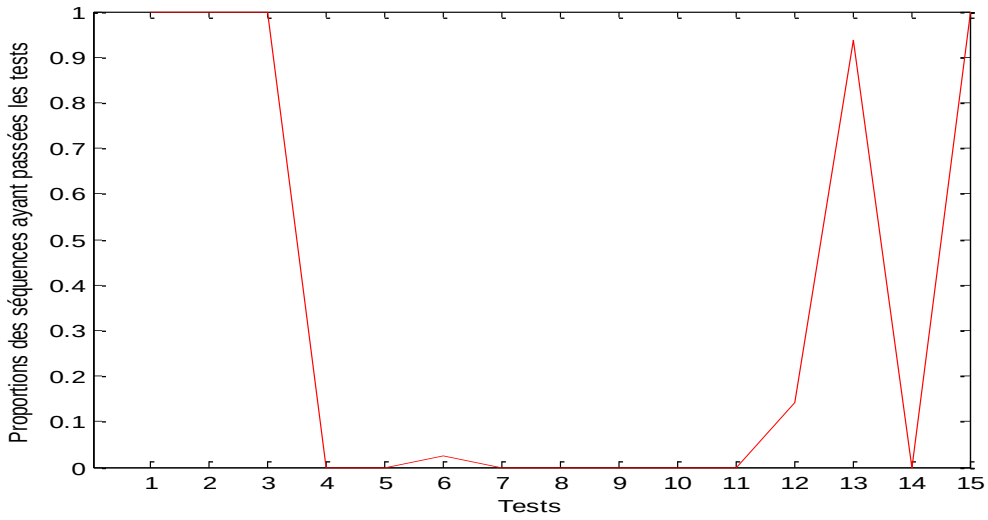


Figure 2. 20. Proportions des séquences passant les tests en fonction des tests.

2.5 Conclusion

Les récurrences chaotiques que ce soit à temps continu ou à temps discret, se caractérisent par leur sensibilité aux conditions initiales, et en plus d'un aspect aléatoire que nous avons validé par la série des tests de NIST. Les tests de NIST sont appliqués pour la validation des fonctions cryptographiques, ils identifient les séquences à mauvaises propriétés statistiques des générateurs de nombres aléatoires et aident à la conception de nouveaux générateurs. D'autres tests sont aussi complets que les tests du NIST comme les tests DIEHARD utilisés par la RSA mais qui nécessitent une taille de fichiers supérieure à 15 Mo pour s'exécuter. Le

choix de la méthode de représentation binaire des outputs de ces récurrences est déterminant. Dans ce chapitre nous avons utilisé deux méthodes de transformation binaire : la troncature et la représentation simple précision du standard IEEE 754. Ce passage du réel au binaire peut garder au maximum l'aspect aléatoire de la suite chaotique (la troncature avec la fonction round pour la suite 1D) comme il peut le dégrader énormément (la représentation IEEE 754 des séquences de la suite 1D). Toutefois le passage d'une grandeur physique théoriquement aléatoire vers une grandeur numérique, ne peut qu'altérer cet aspect.

Chapitre 3

Sécurité par Chaos dans les WSN: état de l'art et contributions

3.1. Introduction

La théorie du chaos décrit le comportement d'un système dynamique non linéaire déterministe, qui dépend énormément des conditions initiales. Un petit changement des conditions initiales conduit à de grands changements dans l'évolution du système. Les systèmes chaotiques révèlent un aspect aléatoire: ils sont déterministes, mais leur sortie ressemble à un comportement aléatoire à qui son paramétrage est inconnu [23].

Dans les systèmes chaotiques, la séquence chaotique est générée avec une condition initiale. La séquence est utilisée comme clé. La même séquence est utilisée pour le décryptage. L'émetteur et le récepteur ont besoin donc d'avoir exactement la même séquence. La synchronisation est primordiale. Les cryptosystèmes basés sur le chaos ont prédominé les cryptosystèmes classiques.

Il existe deux approches dans la modélisation d'un cryptosystème par chaos : analogique et numérique [24].
La première est basée sur la synchronisation de deux systèmes chaotiques conçus pour la sécurité d'information dans un milieu bruité [25]. La deuxième est indépendante de la synchronisation.

3.2. Cryptosystèmes chaotiques analogiques

Ils sont basés sur la synchronisation de deux systèmes chaotiques conçus pour la sécurité d'information dans un milieu bruité [25]. Deux systèmes chaotiques peuvent se synchroniser par couplage ou bien l'un peut conduire l'autre. Un signal scalaire ou plus est envoyé d'un système à l'autre et dans un autre cas d'une troisième source extérieure.

Il existe plusieurs types de synchronisation dus aux définitions mathématiques différentes de la synchronisation [26] : synchronisation complète, synchronisation de phase, synchronisation impulsive, synchronisation projective, synchronisation généralisée, synchronisation de retard, synchronisation induite par le bruit (signal scalaire envoyé via une 3ème source), …

Les cryptosystèmes analogiques sont classés selon [27] en quatre générations.

3.2.1. Les 4 générations des cryptosystèmes chaotiques analogiques

3.2.1.1. La première génération

La première génération des systèmes chaotiques analogiques est apparue en 1993, elle comprend le masquage additif (additive chaos masking) et chaotic shift keying.

a) Le masquage additif

Son principe consiste à réaliser une simple addition entre le signal de sortie de l'émetteur (générateur de chaos) et le signal d'information m_k afin de le masquer (figure 3.1). Le principal inconvénient est que la synchronisation n'est pas exacte et que le bruit additif du canal ne peut pas être extrait de l'information utile. Le masquage additif est donc très sensible au bruit du canal et la différence entre les paramètres du système chaotique d'émission et celui de réception. La transmission de l'information à l'aide de cette méthode est alors non sécurisée [27], [28].

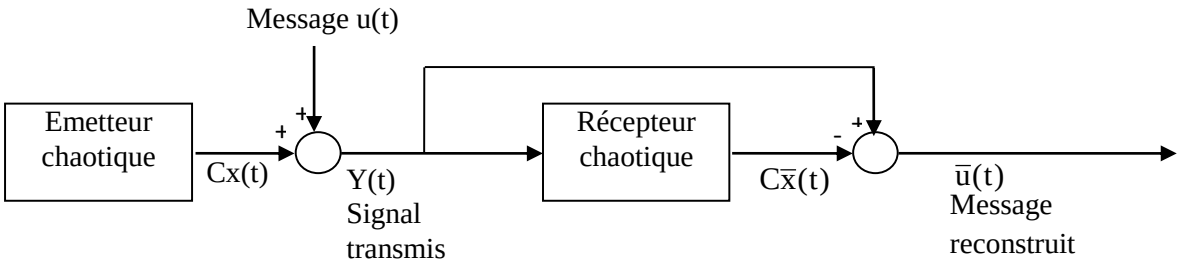


Figure 3. 1. Principe de cryptage par masquage additif.

b) CSK (Chaotic Shift Keying)

Les systèmes de communication modernes sont essentiellement numériques et les systèmes analogiques basés sur le chaos comme le masquage chaotique et la modulation chaotique paramétrée sont abandonnés en faveur des systèmes chaotiques basés sur les méthodes de Shift Keying.

L'un des premiers systèmes chaotiques de communication est la CSK qui est une modulation numérique basée sur la synchronisation au niveau récepteur [29].
Un émetteur CSK commute entre deux générateurs de chaos représentant les bits 0 et 1 [30]. Seulement les signaux binaires peuvent être cryptés par cette méthode [31]. Le récepteur décide, à travers la corrélation entre le signal reçu et un signal de référence synchrone, quel générateur a été utilisé et ainsi le message émis (figure 3.2). La complexité du CSK réside dans la performance réduite du BER. Le chaotic shift keying est robuste contre le bruit et la variation des paramètres du système d'émission - réception.

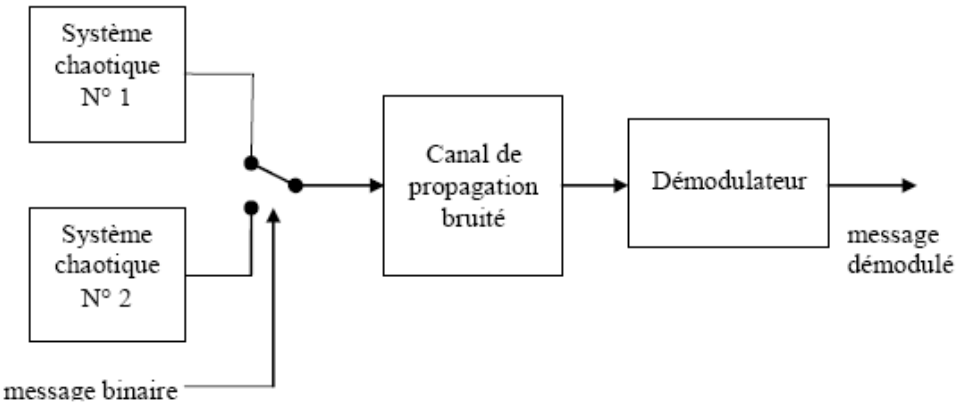


Figure 3. 2. Architecture d'un système de transmission CSK.

c) Amélioration du CSK: DCSK (Differential CSK)

DCSK c'est la version différentielle développée du CSK.
Le problème des récepteurs des systèmes chaotiques, est la difficulté de synchroniser le signal de référence. Les méthodes différentielles sont alors utilisées pour surmonter ces problèmes. Les mêmes conditions initiales et les mêmes paramètres de contrôle sont utilisés dans l'émetteur et le récepteur.

Le DCSK est un cas particulier des DNSK (Differential Noise shift keying) avec la simple particularité que les signaux chaotiques sont utilisés dans les transmetteurs (figure 3.3).
Dans DCSK, une référence chaotique est transmise durant la première demi période du symbole. Si le bit transmis est 1, le signal de référence est retransmis dans la deuxième demi-période, et si le bit est 0 l'inverse de référence est transmis. Ceci se traduit par la fonction suivante :

$$X = \begin{cases} x(t); & 0 \le t < \frac{T_b}{2} \\ \pm x(t - \frac{T_b}{2}); & \frac{T_b}{2} \le t < T_b \end{cases} \quad (3.1)$$

Cette méthode n'est pas sécurisée contre les attaques de corrélation et l'attaque « return map » [32].

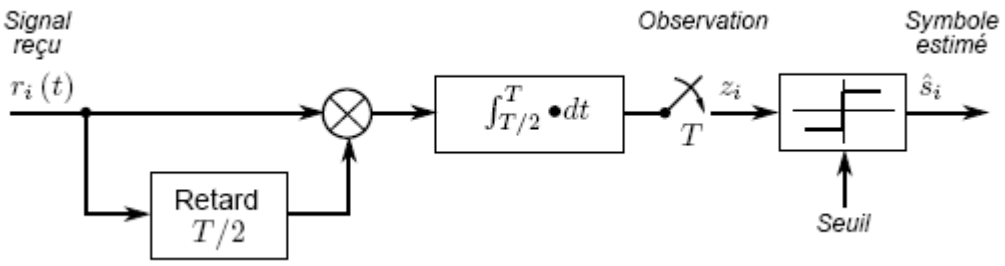


Figure 3. 3. Récepteur non cohérent DCSK.

3.2.1.2. La deuxième génération [27]

La deuxième génération est proposée entre 1993 et 1995, et connue comme la modulation chaotique. Il existe deux méthodes pour moduler le signal, la première est la modulation de paramètres chaotiques, la deuxième est la modulation chaotique non autonome.

La modulation des paramètres chaotiques, donnée par la figure 3.4, consiste à moduler un ou plusieurs paramètres du générateur chaotique par le message d'information, tel que ses trajectoires changent dans différents attracteurs.

Le message m(t) est utilisé pour moduler des paramètres du système chaotique. Par cette méthode, on peut envoyer plusieurs signaux tels que chacun module un paramètre de l'émetteur chaotique.

Au niveau récepteur, la récupération des paramètres modulés peut se baser sur l'estimation simultanée état / paramètre via un contrôleur adaptatif. Pour la modulation chaotique non autonome, donnée par la figure 3.5, l'émetteur commute entre des trajectoires différentes du même attracteur chaotique. Le signal d'information est injecté dans un générateur chaotique jouant le rôle d'émetteur, afin de perturber l'attracteur dans l'espace de phase.

On parle aussi de modulation chaotique lorsque les techniques d'étalement de spectre sont utilisées, en multipliant le message émis par la porteuse chaotique.

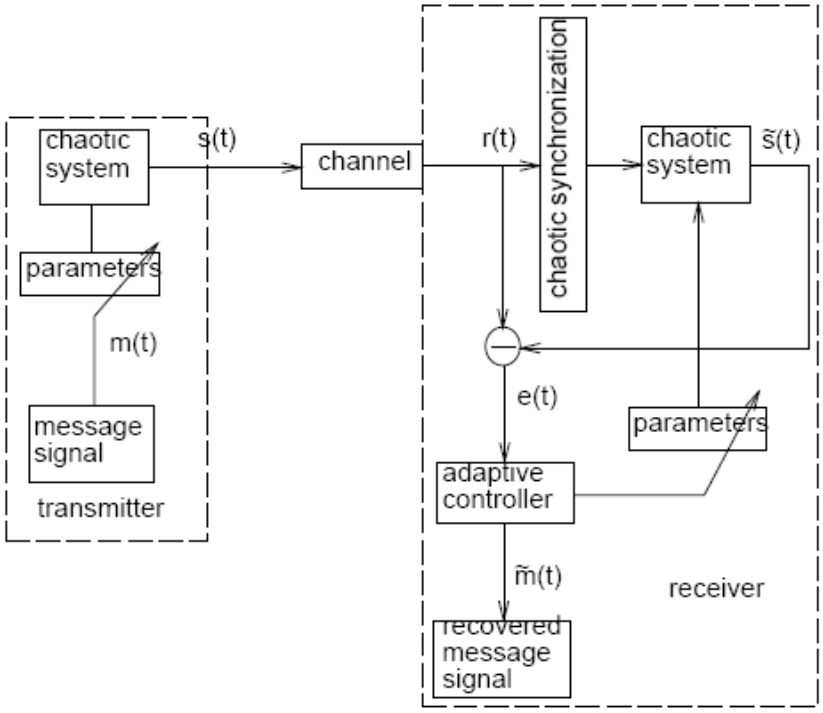


Figure 3. 4. Modulation des paramètres chaotiques [27].

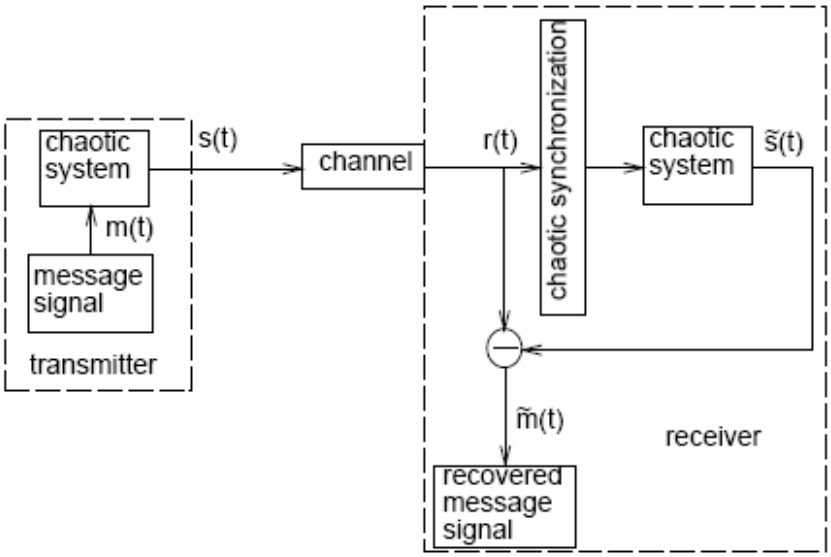


Figure 3. 5. Modulation chaotique non autonome [27].

3.2.1.3. La troisième génération [27]

La troisième génération est proposée en 1997 afin d'augmenter le niveau de sécurité des deux premières générations. Elle est connue sous le nom de cryptosystème chaotique. On a recours donc à la combinaison de la cryptographie classique et à la synchronisation chaotique. Ce système de sécurité n'est pas encore cassé. La clé de cryptage est générée par le système chaotique et utilisée pour crypter l'information, le message y(t) généré est réintroduit dans le système chaotique d'émission, la dynamique chaotique change continuellement. Ensuite, un signal s(t) fonction des variables d'état de l'émetteur est transmis au récepteur à travers le canal public. Un espion qui ne connaît pas la clé secrète ne peut pas extraire le message p(t) du signal transmis. Au niveau récepteur, le signal reçu est utilisé pour synchroniser entre le récepteur et l'émetteur. La synchronisation adaptative est utilisée.
La clé est alors reconstruite par le récepteur, qui peut finalement décoder le message. Le principe général de cette méthode est illustré à la figure 3.6. Il semble que la fonction de cryptage ne devrait pas être simple.

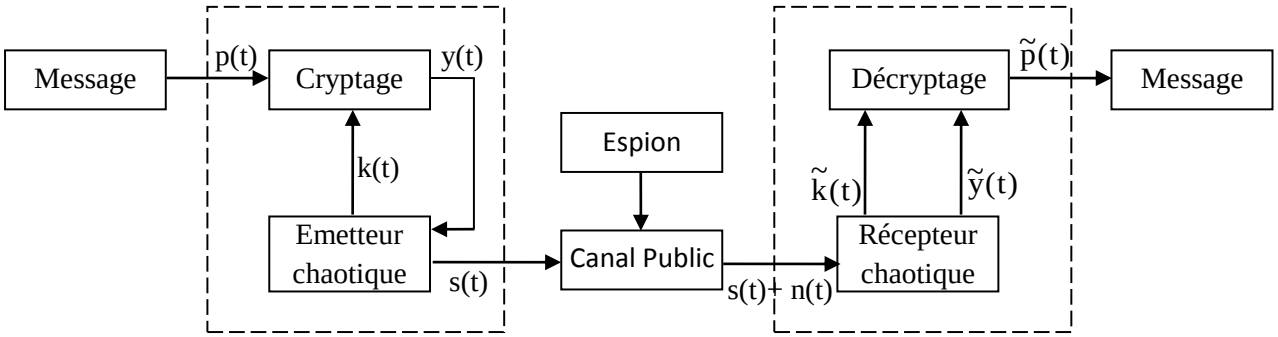


Figure 3. 6. Modulation chaotique non autonome.

3.2.1.4. La quatrième génération [27]

La synchronisation dans les 3 premières générations est continue. Dans cette synchronisation, la bande passante du signal de synchronisation est comparable au signal d'information ; ce qui diminue l'efficacité de l'utilisation de la bande passante. Pour la quatrième génération, une nouvelle technologie, appelée synchronisation impulsive, est utilisée. Elle est basée sur l'introduction d'un opérateur de Dirac. Le problème devient alors un problème de stabilité d'un système impulsionnel.

Cette méthode est beaucoup plus robuste au décalage de paramètre que les trois autres générations.

3.2.2. Avantages et inconvénients du chaos analogique

L'avantage de ce chaos analogique est de crypter et étaler le spectre du signal en même temps et donc de minimiser les outils utilisés pour réaliser ces fonctions.

En outre, il est difficile d'appliquer la technologie de brouillage sur un signal étalé [33]. Cependant, pour implémenter des systèmes de communications pratiques sécurisées par chaos, le chaos libre (modulation chaotique, chaotic switching (shift keying csk), etc..) souffre de plusieurs problèmes [33], [34] :

- Difficulté de déterminer le temps de synchronisation.
- Problème de non linéarité du canal.
- Difficulté de différencier entre petits signaux et bruit de transmission. Les bruits de synchronisation doivent être plus faibles que le signal. Si le rapport signal / bruit est inférieur au bruit du canal, les signaux récupérés n'auront pas de sens.
- Pour établir une communication, au moins deux systèmes chaotiques analogiques appariés sont nécessaires à deux endroits distants. Des systèmes pareils sont difficiles à avoir vue la divergence technologique des systèmes analogiques et l'influence de la variation de la température du milieu. Les erreurs inévitables des valeurs des composants du circuit hardware influencent la synchronisation et la rendent difficile. Une solution est envisagée dans ce cas, elle consiste à utiliser la synchronisation adaptative.

En se basant sur les failles déjà citées, des cryptanalyses de certains cryptosystèmes analogiques ont été présentées dans [35], [36], [37].
Les erreurs inéluctables des valeurs des composants ainsi que la redondance du signal réduisent l'espace des clés et facilitent les attaques à texte clair [23], [33]. Dans notre application, ces systèmes de cryptage analogique sont à éliminer étant donné qu'on vise une implémentation software. En plus, ces systèmes appartiennent plus au domaine de la stéganographie (voir Annexe) qu'à la cryptographie [33].

3.3. Cryptosystèmes chaotique numériques

Les cryptosystèmes chaotiques numériques sont conçus pour les calculateurs numériques où une ou plusieurs suites chaotiques sont implémentées [24]. On peut les classer en deux catégories principales : chiffrement chaotique par flot et chiffrement chaotique par bloc [34].

3.3.1. Chiffrement chaotique par flot

Dans le cas général, le message est crypté bit par bit, à l'aide d'un XOR appliqué à la sortie d'un générateur de nombre pseudoaléatoire basé sur une suite chaotique [24].
L'utilisation de ces systèmes chaotiques dynamiques discrets remonte à l'année 1989 où Matthews présentait une suite à une seule dimension à comportement chaotique. Cette suite est la source des séquences de nombres pseudo-aléatoires qui servent à un masque jetable (one time pad) pour crypter l'information [38], [33].

a) Chiffrement par flot basé sur les GNPA chaotiques (Générateurs de nombre pseudo aléatoire)

Comparé aux générateurs de nombres pseudoaléatoires ordinaires, le chaos est plus simple et moins cher à être intégré sur des systèmes embarqués [39].
La sortie des GNPA chaotique est la clé en flot qui est utilisée pour masquer le message (en appliquant XOR). Deux algorithmes principaux pour générer des nombres pseudo-aléatoires par chaos [40] :

1) Les bits sont extraits partiellement ou totalement à partir des orbites chaotiques.
Un exemple appelé CCS-PRBG (Coupled Chaotic Systems Based PRBG) là où deux systèmes chaotiques sont utilisés.

L'idée de base est de générer un des bits pseudo-aléatoire en comparant deux orbites chaotiques différents et asymptotiquement indépendantes.

Soient deux récurrences chaotiques différentes unidimensionnelles :

$F_1(x_1,p_1)$ et $F_2(x_2,p_2)$ tel que ; $x_1(i+1)=F_1(x_1(i),p_1)$, $x_2(i+1)=F_2(x_2(i),p_2)$, p_1 et p_2 sont les paramètres de contrôle, $x_1(0)$, $x_2(0)$ sont les conditions initiales et $\{x_1(i)\}$, $\{x_2(i)\}$ forment les deux orbites chaotiques.

On définit une séquence de bits pseudo aléatoires comme suit:

$$k(i)= g(x_1(i),x_2(i))=\begin{cases}1, & x_1(i)> x_2(i)\\ \phi, & x_1(i)= x_2(i)\\ 0, & x_1(i)< x_2(i)\end{cases} \tag{3.2}$$

2) L'espace de phase est divisé en $m = 2^n$ parties (sous espaces). On marque chaque partie avec un nombre entre 0 et m-1 et on génère les nombres pseudo-aléatoires $i \in \{0,\ldots,m-1\}$ lorsque l'orbite chaotique correspond à l'$i^{ème}$ sous espace.

b) Chiffrement par flot par l'intermédiaire de l'approche système inverse :

Ce type de chiffrement est proposé par U. Feldmann [41], mais il s'est avéré peu sécurisé contre les attaques connues. Zhou et all ont proposé un modèle amélioré basé sur une suite chaotique linéaire par morceaux PWLCM calculée en précision finie donné par la formule suivante [41] :

$$T(x(t),p)=\begin{cases}x(t)/p & 0 \le x(t) \le p\\ (x(t)-p)/(\frac{1}{2}-p) & p \le x(t) < \frac{1}{2}\\ T(1-x(t),p) & \frac{1}{2} \le x(t) \le 1\end{cases} \tag{3.3}$$

Où p est le paramètre de contrôle $1 < p < 1/2$

Le chiffrement est basé sur la rétroaction de texte chiffré précédent :

Chiffrement : $y(t) = [u(t)+T^m(y(t-1), p)] \pmod 1$,

Déchiffrement : $u(t) = [y(t)-T^m(y(t-1), p)] \pmod 1$.

U(t) est le texte clair, y(t) est le texte chiffré et p est la clé secrète.

L'inconvénient de ce chiffrement est le calcul avec précision finie qui peut dégrader le système. Cette dégradation concerne l'uniformité de la distribution du key-stream générée à partir des itérations de la PWLCM, et conduit à la génération de faibles clés ; ce qui cause une perte de l'information [42].

3.3.2. Chiffrement chaotique par bloc

Les données sont cryptées par bloc de longueur qui différent selon l'algorithme employé. Les propriétés de la transformation chaotique et la façon d'implémenter déterminent le niveau de sécurité d'un tel algorithme [34].

Plusieurs algorithmes de chiffrement chaotique par bloc proposés dans la bibliographie sont basés sur la structure de Feistel (voir annexe pour définition) [43], [44], [45], [46].

Les suites chaotiques sont employées alors pour générer les clés (keystreams) ou bien les tables de substitutions (S-Box). Toutefois, on peut donner un autre classement à ce chiffrement par bloc en se référant à la méthode d'application des suites chaotiques. Deux axes ont été définis dans l'article [24] : Chiffrements par bloc basés sur des systèmes chaotiques inverses et sur les fonctions chaotiques arrondies (ou S-Box).

3.3.2.1. Chiffrement par bloc basé sur des systèmes chaotiques inverses

Ce type de chiffrement proposé par T. Habutsu et al utilise la suite en accent circonflexe déviée (skew Tent map). Elle est présentée par l'équation suivante :

$$F_{\alpha}(x) = \begin{cases} \dfrac{x}{\alpha}, & 0 \le x \le \alpha, \\ \dfrac{(1-x)}{(1-\alpha)}, & \alpha < x \le 1 \end{cases} \tag{3.4}$$

La fonction inverse est donnée par :

$$F_{p^{-1}}(x) = \begin{cases} px, & b = 0, \\ 1-(1-p)x, & b = 1 \end{cases} \tag{3.5}$$

Où b est un bit aléatoire distribué uniformément sur {0,1}, α est la clé secrète.

On choisit la condition initiale x_o égale à la valeur du texte en clair $P \in [0,1]$, tel que $P \neq \alpha$. Puis on calcule dans l'étape suivante le texte crypté $c=F^{-n}(P)$, et on effectue le décryptage juste par l'inverse de la fonction de cryptage tel que le texte en clair soit : $P = F^{n}(c)$, où n est un bit aléatoire.

Biham a cryptanalisé cet algorithme en utilisant l'attaque à texte clair [34], [43], [47]. Pour augmenter la robustesse de ce chiffrement contre l'attaque à texte clair choisi, on peut utiliser d'autres suites non linéaires au lieu de la Tent map.

Masuda et al [34] ont repris cette équation en amplifiant les intervalles de x et de α pour utiliser des entiers au lieu de réels. On va reprendre ce principe avec plus de détails, dans la partie de construction des S-Box.

3.3.2.2. Chiffrement par bloc basé sur les fonctions chaotiques arrondies ou S-Box

Ce type de chiffrement permet de crypter les données à l'aide des S-Box chaotiques. La table de substitution non linéaire ou S-Box constitue la partie essentielle de l'architecture Feistel. Les méthodes de construction des S-box ne se limitent pas à ce classement qu'on a établi mais les concepteurs se sont inspirés aussi des générateurs chaotiques pseudoaléatoires.

Avant d'entamer cette partie, on ne peut que présenter quelques algorithmes de cryptage chaotique par bloc dédiés aux réseaux de capteurs.

3.3.2.3. Chiffrement chaotique par Bloc pour Réseaux de Capteurs

Les cryptosystèmes chaotiques orientés pour les réseaux de capteurs ne sont pas très nombreux. Un exemple de ces cryptosystèmes a été proposé par Chen [48], il est basé essentiellement sur une fonction f donnée par la figure 3.7. La suite de 8 bits que les auteurs prétendent chaotique est la suivante:

$$y = (x<<2) - ((x^2) >> 6) - 1 ; \qquad (3.6)$$

x et y sont des entiers à 8 bits non signés.
Les opérateurs "<<" et ">>" constituent respectivement le décalage binaire à droite et à gauche.

Ses caractéristiques chaotiques n'ont pas été vérifiées dans l'article d'origine. Les 4 bits à droite (R_i) du bloc de 8 bits sont considérés comme un demi octet, il est étendu pour former un octet (cette procédure n'a pas été expliqué dans l'article d'origine). L'opération XOR est appliquée sur l'octet résultant et la clé secrète. La suite supposée chaotique est alors itérée avec vecteurs d'entrées (input) le résultat du Xor précédent. La structure de Feistel est adoptée pour 4 tours seulement ; ce qui ne respecte pas les lois d'application de la structure de Feistel et rend l'algorithme accessible aux attaques. Cet algorithme a été cryptanalysé par [49] à travers l'attaque différentielle.

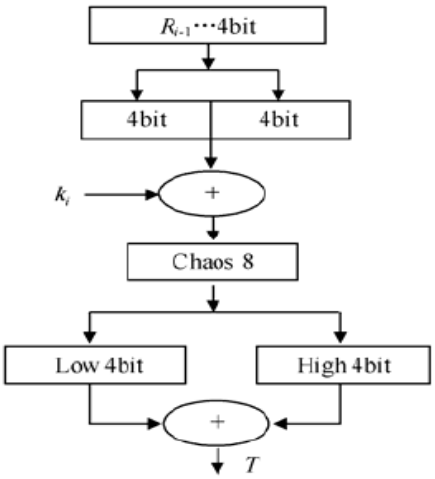


Figure 3. 7. Fonction f de l'algorithme de Chen [48].

Un autre algorithme basé sur des suites chaotiques est conçu par Fang et al dans [50].

La suite logistique originale est une fonction continue sur [0,1], alors que le réseau de capteurs est un système de communication numérique. Pour n'importe quel système numérique utilisé, la fonction ne serait pas continue, les inputs et les paramètres de contrôle auront des points limités sur l'intervalle de variation ; ce qui engendre une suite qui n'est pas réellement chaotique étant donné que l'ergodicité ne sera pas maintenu sur tout l'intervalle [0,1]. Les auteurs dans ce sens, proposent d'élargir l'intervalle de variation de la suite ; la suite logistique deviendra donc la logistique modifiée donnée par cette équation :

$$x_{n+1} = \mu x_n (N - x_n / m) / N; \tag{3.7}$$

Avec $x \in [0, mN]$, $N=2^K$ et $m=2^k$; K et k sont tous les deux des entiers.
Puisque m et N sont des puissances de 2, le processus dans le microcontrôleur va se baser sur des opérations binaires, la sortie est binaire aussi.
Cette équation requiert seulement des opérations de multiplication, addition, de décalage mais aussi de division.
Lorsque m.N tend vers l'infini, la séquence est ergodique, et plus le nombre m.N est grand plus la séquence devient chaotique.

Dans cet article, les auteurs proposent un système chaotique formé par l'association de deux suites : la logistique et la « Tent map » nommée N-Logistic-Tent. Comparée aux suites d'origine, ces suites possèdent une plage de données plus étendue.

Un autre avantage est la taille plus grande de la clé qui est élargie de (x_i, λ) à (x_i, y_i, μ, β, m, N).
L'équation (3.8) représente la suite logistique modifiée et l'équation (3.9) est la N-tent map.

$$x_{n+1} = \mu x_n (N - x_n / m) / N - y_n / 2; \tag{3.8}$$

avec $x \in [0, mN]$ et $\mu \in [0,4]$;

$$y_{n+1} = \beta(N - |N - y_n|) \tag{3.9}$$

avec $y \in [0, 2N]$, $\beta \in \{1,2\}$.

La clé de diversification ou « seed key » pour cet algorithme est l'ensemble (x_i, y_i, μ, β, m, N). Un préambule est utilisé pour synchroniser entre émetteur et récepteur. Le cryptage est réalisé à l'aide de l'opération XOR, de même pour le décryptage.

Nous avons utilisé cette suite pour construire la S-Box, mais en tenant compte des restrictions dans un réseau de capteurs. Les variables seront représentées sur deux octets (x et y) et on a fixé les autres paramètres tels que $m = 4$, $\mu = 4$, $N = 2^{14}$, $\beta = 2$. Le nombre d'itérations minimal dépend de la condition initiale à donner au système.

La représentation sur deux octets de "y" a mené à une convergence vers un point fixe. Pour remédier à ce problème, on a ajouté une perturbation à l'aide d'un registre à décalage.

Dans Silva [51], les auteurs utilisent un LFSR de période maximale 128. Un m-LFSR assure un cycle de (2^{128}-1) valeurs différentes. Son polynôme primitif est : $x^{128}+x^7+x^2+x+1$.

Les auteurs dans [51] ont présenté un algorithme de cryptage à base du système de Lorenz discrétisé donné par les équations suivantes :

$$\begin{cases} x_{k+1} = x_k + \sigma(y_k - x_k)\Delta t \\ y_{k+1} = y_k + [x_k(\rho - z_k) - y_k]\Delta t \\ z_{k+1} = z_k + [x_k y_k - \beta z_k]\Delta t \end{cases} \quad (3.10)$$

Δt constitue le pas d'intégration et les paramètres ρ et σ sont les mêmes que ceux du système d'origine cité dans la section 2.3.1 du deuxième chapitre ; par contre β est choisi égal à 3.

Toutefois, ce nouveau système doit maintenir l'aspect dynamique comme la sensibilité aux conditions initiales et la non convergence vers une seule valeur ou la génération de cycle court. Dans ce but, les auteurs ont choisi de changer le pas d'intégration Δt selon une fonction de perturbation chaotique.

Le système de perturbation est le suivant [51]:

$$\begin{cases} y_{k+1} = y_{k+1} \oplus LFSR_i \\ z_{k+1} = z_{k+1} \oplus LFSR_{i+1} \\ \Delta t = \Delta t \oplus x_{k+1} \end{cases} \quad (3.11)$$

La perturbation commence à t = 0 et la prochaine perturbation sera après Δ itérations. Le Δ doit être inférieure à la longueur du cycle sans perturbation.

L'algorithme est initialisé à travers x et le 128 m-séquence LFSR.

Un autre LFSR auxiliaire est itéré et sa nouvelle valeur est donnée à y. LFSR auxiliaire est itéré encore une fois et la valeur est donnée à z. La valeur d'une autre itération est donnée à Δt.

Les valeurs de y, z et Δt changent selon 3 fonctions F_1, F_2, F_3 qui constituent respectivement la rotation de 32 bits à droite, à gauche et de 64 bits à droite.

Ces 3 fonctions sont appliquées aux LFSR comme le montre le système d'équations suivant [51]:

$$\begin{cases} LFSR=seed;\ x_0=LFSR_0; \\ y_0=F_1(LFSR_1);\ z_0=F_2(LFSR_2); \\ \Delta t_0=F_3(LFSR_3); \end{cases} \quad (3.12)$$

3.4. Construction de S-box à base de suite chaotique : Etat de l'art et méthodes proposées

La première vague d'algorithmes de construction des S-Box dynamique à l'aide des suites chaotiques consiste à itérer les suites (à une dimension ou plus) et à discrétiser les valeurs réelles obtenues par plusieurs moyens pour créer une matrice d'entiers aléatoire entre 0 et 255. Parfois dans le but d'augmenter la sécurité contre certaines attaques et diminuer la dépendance des S-Box des suites chaotiques, les auteurs ajoutent des rotations et des permutations à la table obtenue.

Une deuxième alternative repose sur la discrétisation de la suite chaotique et l'accroissement de l'intervalle de variation des conditions initiales et des paramètres de contrôle passant des réels entre 0 et 1 à des entiers. Dans cette section, nous étudions quelques méthodes existantes de construction des S-Box et nous analysons leurs performances.

3.4.1. S-Box à base de suite chaotique à sortie réelle

3.4.1.1. Attribution d'entiers aux sous intervalles de l'espace de phase

La première méthode présentée par Jakimoski dans l'article [43] propose une procédure de discrétisation pour la Logistic map, dont l'algorithme est le suivant:

1) Diviser l'espace de phase en n+1 régions égales, assigner les nombres 0, …, n aux régions.

   Si un point tombe dans une région i, alors son amplitude est i.
2) Choisir aléatoirement un nombre de départ de chaque région. Déterminer son image après N itérations de la suite.
3) Trouver l'ensemble S de points de départ qui ont une image unique.

   Choisir un sous ensemble A qui contient 256 éléments de S et déterminer l'ensemble B de leurs images.
4) Assigner une nouvelle amplitude entre 0 et 255 aux éléments de A suivant leurs anciennes amplitudes.

Refaire l'étape 4 pour l'ensemble B.

Le nombre de régions est choisi tel que le nombre moyen de points de départ ayant une seule image est légèrement supérieur à 256, exemple le nombre d'éléments de l'ensemble S est card(S) = 259.
On choisit le nombre des régions de l'espace de phase n largement supérieur à 256 (exemple n = 767).

L'article [52] reprend presque les mêmes étapes, en changeant juste l'intervalle de r (paramètre de la suite Logistique), r ∈ [3.59, 4] et n = 767 par 537.

- ***S-Box fabriquée à partir de la suite PWLCM :***

On peut rester dans la même idée qui est l'allocation d'intervalle réel, mais cette fois en utilisant une autre suite qui est la PWLCM. Le choix de cette suite revient à la non discontinuité des intervalles du choix des paramètres de la suite. Toutes les valeurs contenues dans l'intervalle garantissent l'aspect chaotique de la suite, contrairement à la suite logistique ou bien la récurrence de Hénon dont la définition des limites des intervalles n'est pas simple étant donné l'existence de fenêtres périodiques même au sein de l'intervalle où on prétend avoir l'existence du chaos. Cette suite a une densité uniforme invariante et une fonction d'autocorrélation exponentiellement décroissante ; ce qui la rend adéquate pour les cryptosystèmes chaotiques [24].

Une méthode similaire pour construire des S-Box à base de suites PWLCM est présentée dans l'article [53] et dont nous donnons son fonctionnement par l'organigramme de la figure 3.8.

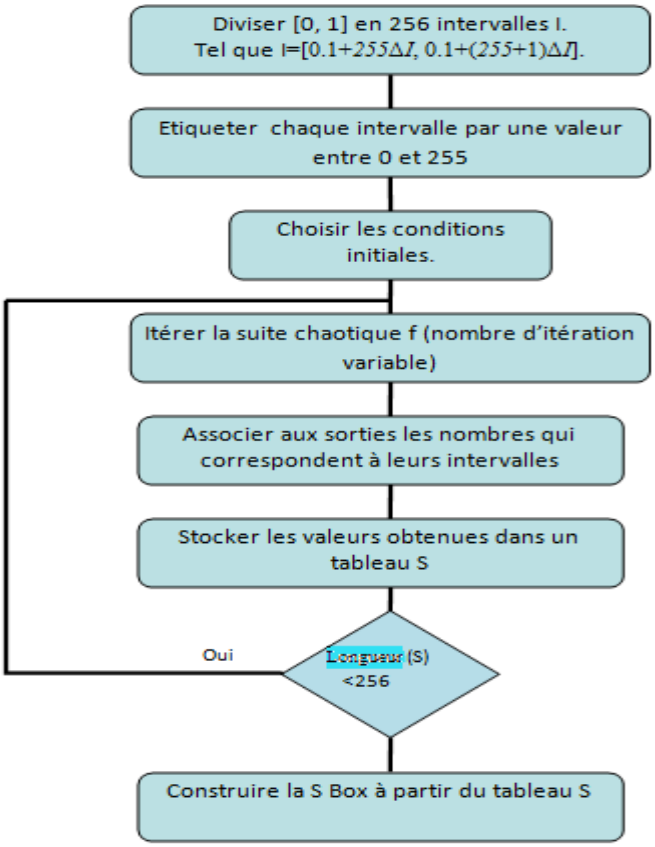


Figure 3. 8. Organigramme de construction de la S-Box par la PWLCM.

Les tableaux 3.1 et 3.2 représentent une S-Box 16x16 et son inverse réalisées à l'aide de la PWLCM pour un paramètre p = 0.15 et une condition initiale $x_0 = 0.76$.

Tableau 3. 1. S-Box construite à partir de PWLCM.

50	241	45	224	133	222	86	152	49	205	108	78	235	226	220	14
65	51	52	255	71	18	165	217	213	109	232	184	100	74	120	29
110	70	40	186	126	15	180	32	44	118	67	127	242	56	179	7
238	122	31	121	4	175	138	163	63	147	247	5	123	22	197	168
134	66	11	84	209	151	182	187	216	203	239	240	132	76	89	23
95	112	169	162	55	219	68	171	35	125	16	200	20	139	236	178
195	243	170	188	80	26	117	164	24	158	0	79	111	115	212	82
94	106	167	114	153	172	173	119	128	83	72	149	46	251	193	189
192	225	174	248	36	160	159	124	41	196	129	214	87	54	48	231
102	9	155	233	27	194	198	28	37	93	116	12	103	161	92	253
215	244	210	88	177	96	85	3	30	227	254	107	6	104	190	57
38	146	53	156	145	199	166	62	8	2	208	229	221	150	60	218
33	234	75	207	148	230	176	99	237	185	58	183	42	223	97	142
17	73	137	61	228	157	181	206	25	39	81	130	47	191	201	143
113	131	141	98	1	204	136	101	140	34	154	19	59	10	77	245
246	43	249	202	105	69	90	211	252	21	64	13	91	250	144	135

Tableau 3. 2. S-Box inverse.

106	228	185	167	52	59	172	47	184	145	237	66	155	251	15	37
90	208	21	235	92	249	61	79	104	216	101	148	151	31	168	50
39	192	233	88	132	152	176	217	34	136	204	241	40	2	124	220
142	8	0	17	18	178	141	84	45	175	202	236	190	211	183	56
250	16	65	42	86	245	33	20	122	209	29	194	77	238	11	107
100	218	111	121	67	166	6	140	163	78	246	252	158	153	112	80
165	206	227	199	28	231	144	156	173	244	113	171	10	25	32	108
81	224	115	109	154	102	41	119	30	51	49	60	135	89	36	43
120	138	219	225	76	4	64	255	230	210	54	93	232	226	207	223
254	180	177	57	196	123	189	69	7	116	234	146	179	213	105	134
133	157	83	55	103	22	182	114	63	82	98	87	117	118	130	53
198	164	95	46	38	214	70	203	27	201	35	71	99	127	174	221
128	126	149	96	137	62	150	181	91	222	243	73	229	9	215	195
186	68	162	247	110	24	139	160	72	23	191	85	14	188	5	205
3	129	13	169	212	187	197	143	26	147	193	12	94	200	48	74
75	1	44	97	161	239	240	58	131	242	253	125	248	159	170	19

Encore dans le même principe qui est l'allocation d'intervalle ou d'orbites de la suite chaotique à des entiers entre 0 et 255, Wang et all [54] ont conçu une fonction appelée OPS(i, j, K) qui génère la S-Box dynamique, où i est le nombre d'itérations de la suite chaotique, j est le nombre de sous intervalles et K est l'une des permutations arbitraires de la séquence d'entiers $\{0, 1, 2, \ldots, 2^n-1\}$. La fonction OPS(i, j, K) inclut les opérations suivantes :

1) Diviser l'intervalle en j sous intervalles $\left[0,\frac{1}{j}\right), \left[\frac{1}{j},\frac{2}{j}\right) ;\ldots ;\left[\frac{j-1}{j},1\right]$;
2) Itérer la suite i fois.
3) On suppose que la sortie de la suite est située dans le $m^{ième}$ sous intervalle $\left[\frac{m-1}{j},\frac{m}{j}\right]$ échanger le $m^{ième}$ entier k_m avec le $j^{ième}$ entier k_j dans la séquence K.
4) Changer le paramètre p de contrôle tel que $b = 0.9+0.1*k_m/(2^n-1)$.
   Le nombre d'itérations choisi dans ce cas est i = 32. La fonction OPS(32, j, K) est répétée de $j = 2^n$ à 2.

Cette méthode n'est pas efficace puisqu'une seule valeur est utilisée pour chaque 32 itérations. Les états $x_{i+p*32}(0, 1,2,\ldots, 2^n-2)$ sont utilisés pour générer une seule Sbox de $n*n$.
Pour augmenter l'efficacité, toutes les valeurs $x_{i+p*32+g}(0,1,2,\ldots, 2^n-2)$ doivent être donc utilisées, g peut être n'importe quel élément de l'ensemble {0, 1, 2, .., 31}. Ils peuvent créer 32 Sbox en même temps.

Pour cela les auteurs proposent 3 étapes pour construire 32 S-Box simultanément :

1ère étape : on définit 32 séquences d'entiers identiques $K_1, K_2, K_3, \ldots, K_{32}$, qui sont une des permutations arbitraire de la séquence d'entiers $\{0,1, 2, \ldots,2^n-1\}$.

2ème étape : Répéter l'exécution des fonctions :OPS(1, j, K_1) OPS(1, j, K_2) OPS(1, j, K_{32}) de j=2^n à 2. Le pseudo-code suivant est proposé :

```
For(j=2^n ;j>1 ;j--)
{
for(i=1;i<32;i++) OPS(1,j,K_i)
}
```

3ème étape: traduire K1, K2, …, K32 en 32 n×n S-Box.

3.4.1.2. Binairisation des réels

Pour avoir des entiers codés sur 8 bits à partir d'un nombre réels, l'idée la plus simple est de binairiser la sortie réelle pour enfin passer à des entiers qui varient en fonction de la précision binaire établie. De ce fait, Tang dans son article [55] utilise une fonction booléenne qui est une fonction qui renvoie les valeurs 0 ou 1. Généralement, une séquence de bits à l'entrée produit un bit à la sortie.

On désigne un nombre flottant x par :

$$x = 0.b_1(x)b_2(x)b_3(x)\ldots b_i(x) \; ; \; x \in [0,1], \; b_i(x) \in [0,1] \; ; \tag{3.13}$$

Le $i^{ième}$ bit est exprimé par :

$$b_i(x) = \sum_{r=1}^{2^i-1} (-1)^{r-1} \theta(x)_{(r/2^i)} \tag{3.14}$$

$\theta_t(x)$ est la fonction seuil définie par :

$$\theta_t(x) = \begin{cases} 0 & x < t \\ 1 & x \geq t \end{cases} \tag{3.15}$$

Une séquence binaire $B_i^n = f(x) = \{b_i(\tau^n(x))\}_{n=0}^{\infty}$ tel que n est la longueur de la séquence binaire et $\tau^n(x)$ est la $n^{ième}$ itération de la suite. La suite f peut être considérée comme une fonction booléenne, le réel x à l'entrée donne une valeur logique 0 ou 1.

- ***Génération de la SBox***

La première étape est de générer une séquence binaire de 8 bits (selon l'équation 3.14) et de la transformer en un entier. Si cet entier existe déjà, on itère la suite. La deuxième étape est une permutation qui dépend de la clé utilisée pour mélanger le tableau non linéairement en appliquant la suite du boulanger (Baker map) à plusieurs reprises.

La suite du boulanger (Baker map) est donnée par les équations suivantes :

$$\begin{cases} B(x,y) = (2x, y/2) & 0 \leq x \leq \frac{1}{2} \\ B(x,y) = (2x-1, y/2 + \frac{1}{2} & \frac{1}{2} \leq x \leq 1 \end{cases} \tag{3.16}$$

L'article [56] permet de corriger des erreurs qui sont survenues dans [55] comme la répétition d'un entier dans la S-Box, et aussi propose une amélioration en augmentant la dimension de la suite du boulanger (de 2-D vers 3-D), et en utilisant la suite de Chebyshev au lieu de la Logistique pour générer la séquence binaire aléatoire.

3.4.1.3. Chaos spatiotemporel

Un exemple des systèmes chaotiques spatiotemporels les plus utilisés est la suite à treillis couplée [57]:

$$x_{n+1}(i) = (1-\varepsilon)f(x_n(i)) + \frac{\varepsilon}{2}\left[f(x_n(i+1)) + f(x_n(i-1))\right] \quad (3.17)$$

Avec n = 1, 2, … est l'indice du temps ou l'indice d'état ; i = 0, 2,…, N-1 est l'indice de treillis ;
f est la suite et $\varepsilon \in (0,1)$ est une constante de couplage.
La condition de limite périodique, $x_n(N+i) = x_n(i)$ est utilisée dans ce système.

Dans cette méthode proposée par Yuan et all [57], la suite logistique malgré ses inconvénients est utilisée comme suite locale :

$$x_{n+1} = \mu x_n(1-x_n) \quad (3.18)$$

Avec $\mu \in (3.57, 4)$, $\varepsilon = 0.05$ et $N = 2^L$;

Yuan et all proposent les étapes suivantes pour obtenir la S-box chaotique :

1) Définir les valeurs initiales du treillis dans le système spatiotemporel.
2) Itérer le système spatiotemporel d fois tel que d > 2N.
3) Définir une séquence d'entiers S, qui est vide.
4) Ajouter les indices du treillis à S selon l'ordre décroissant de la valeur du treillis max à la valeur du treillis min.

   Si la valeur des 2 treillis est la même, choisir la valeur du treillis (Lattices) la plus grande.
5) Si la séquence d'entiers S contient 2^L entiers former ; alors une table de $2^{n/2} \times 2^{n/2}$ éléments.

Répéter les étapes de 2 à 4 pour générer d'autres SBox.

3.4.1.4. Méthodes proposées

a. S-Box fabriquée à partir de la combinaison de deux suites:

Pour améliorer la non-linéarité et l'immunité contre les attaques différentielles, mettre des structures chaotiques multiples en cascade s'avère intéressante [23].

Nous avons pensé donc à combiner deux suites chaotiques 1D [58] et 3D pour construire les tables de substitution (S-Box) 16×16. Nous proposons alors deux méthodes [59].

- La 1ère Méthode proposée repose sur les instructions suivantes:
  - Choisir arbitrairement les conditions initiales de la PWLCM.

- Itérer la suite n fois. Chaque 3 sorties seront rassemblées pour former les conditions initiales de la suite 3D.
- Itérer la suite 3D N fois, pour chaque ensemble (x_0, y_0, z_0) et construire un vecteur v qui contient la $N^{ième}$ itération de (x_i, y_i, z_i), avec $0 < i < N$.
- Choisir aléatoirement 256 valeurs du vecteur v.
- Remplacer chaque valeur avec son rang [45], en procédant par un tri par exemple.
- Organiser le tableau résultant en une matrice 16×16.

➢ La 2ème Méthode se base sur les étapes citées ci-dessous. La S-Box et son inverse sont présentés par les tableaux 3.3 et 3.4 :

- Choisir arbitrairement la condition initiale de la PWLCM.
- Itérer la suite chaotique n fois. Chaque trois entrée sont rassemblées pour former les conditions initiales de la suite 3D.
- Itérer la suite 3D N fois, pour chaque ensemble (x_0, y_0, z_0) et construire un vecteur v qui contient la Nième itération de (x_i, y_i, z_i), avec $0 < i < N$.
- Ecrire chaque élément de v en héxadécimal et convertir les résultats en entier de 0 à 255.
- Répéter la troisième et la quatrième étape jusqu'à avoir 256 éléments différents.
- Organiser le tableau en une matrice 16×16.

On peut remarquer que la seule différence entre les deux méthodes est la conversion de la sortie réelle en entier comprise entre 0 et 255.

En se basant sur le principe de conversion des deux méthodes, on peut construire des S-Box à partir d'une seule suite chaotique en passant directement de la première étape à la troisième. Pour les analyses de sécurité qui suivent, on fixe n à 258 et N à 550.

D'autres auteurs se sont inspirés du même principe du tri des outputs de la suite chaotique, comme dans [60] où les auteurs ont opté pour une suite continue (Lorenz) ; leur proposition est résumée dans les 6 étapes suivantes :

1ère étape: les trajectoires sont obtenues en résolvant le système de Lorenz avec les conditions initiales sélectionnées.

2ème, 3ème et 4ème étapes : La trajectoire sélectionnée est échantillonnée (chaque échantillon lui correspond un nombre de 0 à 255 selon un ordre croissant de leur apparition).
Le nombre qui correspond à la plus petite valeur est mis le premier dans le tableau S. Une Sbox n×n est générée.

5ème étape : Les lignes sont décalées à gauche par l'indice de la ligne sauf la première cellule (indice 0).

6ème étape : Une rotation des colonnes est effectuée.

Pour générer la Sbox, 100000 points sont générés et les 10000 premières valeurs sont négligées considérées comme transitoires. Özkaynak et all [60] considèrent que le point le plus fort de leur proposition est la non dépendance totale au système chaotique puisqu'il ajoute à la dernière phase une étape de décalage de lignes et de colonnes.

Tableau 3. 3. S-Box à l'aide de la combinaison de deux suites 1D et 3D (2ème méthode).

63	52	127	249	213	57	236	48	46	105	117	186	169	113	20	146
224	42	83	49	33	137	8	37	208	97	69	247	88	13	73	84
75	158	217	16	136	243	214	176	139	0	96	32	98	68	163	51
120	228	156	138	44	9	235	191	115	161	221	14	193	147	40	11
167	35	123	19	34	78	86	76	184	17	95	153	143	155	178	54
227	27	230	112	216	134	248	200	65	125	30	58	199	1	244	103
47	238	203	5	108	10	232	36	85	149	222	25	255	162	180	70
121	177	104	196	187	64	166	56	145	28	174	171	79	189	114	124
234	66	12	205	126	245	22	183	210	106	170	18	250	182	119	190
77	72	194	130	168	89	185	173	172	82	150	43	142	231	99	144
165	135	118	206	220	23	80	204	62	53	229	225	2	151	116	159
61	55	179	100	246	71	253	198	202	188	15	133	81	24	109	39
110	212	92	128	181	233	129	239	209	4	90	107	251	3	7	41
218	197	215	237	94	254	67	207	192	91	164	157	141	87	122	131
242	211	26	223	140	111	201	241	132	160	219	59	29	45	240	101
226	148	175	50	152	6	60	102	195	74	31	38	21	252	93	154

Tableau 3. 4. Inverse de la S-Box.

41	93	172	205	201	99	245	206	22	53	101	63	130	29	59	186
35	73	139	67	14	252	134	165	189	107	226	81	121	236	90	250
43	20	68	65	103	23	251	191	62	207	17	155	52	237	8	96
7	19	243	47	1	169	79	177	119	5	91	235	246	176	168	0
117	88	129	214	45	26	111	181	145	30	249	32	71	144	69	124
166	188	153	18	31	104	70	221	28	149	202	217	194	254	212	74
42	25	44	158	179	239	247	95	114	9	137	203	100	190	192	229
83	13	126	56	174	10	162	142	48	112	222	66	127	89	132	2
195	198	147	223	232	187	85	161	36	21	51	40	228	220	156	76
159	120	15	61	241	105	154	173	244	75	255	77	50	219	33	175
233	57	109	46	218	160	118	64	148	12	138	123	152	151	122	242
39	113	78	178	110	196	141	135	72	150	11	116	185	125	143	55
216	60	146	248	115	209	183	92	87	230	184	98	167	131	163	215
24	200	136	225	193	4	38	210	84	34	208	234	164	58	106	227
16	171	240	80	49	170	82	157	102	197	128	54	6	211	97	199
238	231	224	37	94	133	180	27	86	3	140	204	253	182	213	108

3.4.2. Discrétisation des suites chaotiques (du réel vers Entiers)

3.4.2.1. Etude de l'algorithme de Tang

Inspiré par la méthode proposée par Masuda [34] pour discrétiser la tent map, Tang [61] propose une adaptation de cet algorithme pour construire des S-Box dynamiques. Un système dynamique continu est le couple (x,f) ; avec x est l'espace d'état et f est la transformation de

X à X. La trajectoire d'une condition initiale x_0 est l'ensemble d'éléments de x, qui est obtenu par l'itération de x_t, tel que $x_{t+1}=f_a(x_t)$; t=0,1,2,…

La fonction "skew tent map" est obtenue en choisissant son point critique a différent de 0.5. f_a est définie par :

$$f_a(x) = \begin{cases} \dfrac{x}{a}, & 0 < x \le a \\ \dfrac{x-1}{a-1}, & a < x \le 1 \end{cases} \tag{3.19}$$

L'inverse de cette fonction n'existe pas, puisque c'est une fonction (two to one).
Formellement :

$$f_a^{-1}(x)=ax \text{ or } 1+(a-1)x. \tag{3.20}$$

L'exposant de Lyapunov (voir annexe) est:

$$\lambda = -a\log a - (1-a)\log(1-a) \tag{3.21}$$

- *Discrétiser la suite chaotique*

Pour un entier M^2, on écrit : P = C = {X ; X=1,2,…M} et K = {A ; A = 1,2,….M}.
Avec P, C et K l'espace du texte en clair défini, l'espace du texte crypté et la clé. La permutation chaotique scalaire F_A est: $X_{t+1} = F_A(X_t)$, t = 0, 1, 2,…

$$F_A = \begin{cases} \left(\dfrac{M}{A}X\right), & t < X \le A \\ \left\lfloor \dfrac{M}{M-A}(M-X) \right\rfloor + 1, & A < X \le M; \end{cases} \tag{3.22}$$

Un nombre k d'itérations est choisi, si $k \approx 2{,}39 \log_2 M + 15$, la séquence satisfait les propriétés : sensibilité aux conditions initiales, sensibilité au point critique a, sensibilité au nombre d'itérations, propriétés statistiques aléatoires, décroissance exponentielle de l'information, indépendance des opérations bits à bits.

- *Générer la SBox*

1ère étape : une séquence d'entiers $K = X_0 = \{1, 2,..., 2^n\}$ obtenue de façon arbitraire est considérée comme clé.
2ème étape : Pour un $M=2^n$ et A on itère la suite plus que k fois avec la valeur intiale X_0, une séquence {X} est obtenue finalement.

Les variables dans [61] sont choisies de la manière suivante :
n=8, $M=2^8=256$, A=123, le nombre d'itérations est 39, et $X_0 = \{1, 2,..., 256\}$.

Remarque : La génération de la séquence aléatoire doit être suivie d'un test de doublons. Le code devient alors plus complexe.

3.4.2.2. Méthodes Proposées

Nous avons utilisé l'algorithme de Fang cité dans la section 3.3.2.3 pour construire la S-Box à partir de la suite N-Logistic-Tent (tableau 3.5), mais en tenant compte des restrictions dans un réseau de capteurs. Nous avons choisi les variables représentées sur deux octets (x et y) et nous avons fixé les autres paramètres tels que $m = 4$, $\mu = 4$, $N = 2^{14}$, $\beta = 2$. Le nombre d'itérations minimal dépend de la condition initiale à donner au système.

La représentation sur deux octets de " y " a mené à une convergence vers un point fixe. Pour remédier à ce problème, nous avons recours à ajouter une perturbation à l'aide d'un registre à décalage.

Pour chaque itération, on binairise la sortie de ce système. Chaque octet représentera un élément de la S-Box. En cas de répétition, la suite N-Logistic-Map est itérée encore une fois et les étapes précédentes sont exécutées.

Nous avons essayé de construire une autre S-Box basée cette fois sur la suite de Lorenz discrétisée proposée dans la même section 3.3.2.3. La taille des variables sera limitée à 2 octets ; ce qui mène à une sortie avec un cycle court. Pour remédier à ce problème, nous avons recours à une perturbation qui consiste à XORer les sorties de la suite avec un registre de décalage ainsi que le pas d'intégration. Chaque octet des 3 sorties de la suite représente un entier entre 0 et 255 de la S-Box. Nous parlerons avec plus de détails de cette méthode dans la partie implémentation.

Tableau 3. 5. S-Box construite à partir de nommé N-Logistic-Tent map pour des conditions initiales y=30000 x=1200.

105	226	133	195	40	216	22	196	97	154	201	162	231	249	92	203
137	32	155	247	74	199	207	234	205	20	102	59	31	0	174	119
70	246	65	45	89	150	130	173	75	179	221	85	252	141	198	165
129	134	78	245	191	37	233	254	94	68	55	46	3	140	126	139
138	145	109	192	210	63	51	30	96	16	223	26	87	237	182	213
176	218	18	80	112	33	238	187	93	253	4	217	104	90	73	106
219	248	66	82	186	197	7	121	50	43	28	143	14	214	108	211
114	107	62	157	208	222	48	180	88	239	13	8	171	131	230	35
57	9	116	2	146	200	123	1	34	212	24	193	241	12	49	77
127	120	236	229	27	42	100	56	64	160	250	69	152	242	71	53
29	122	47	185	235	183	136	72	167	206	181	240	243	6	149	225
220	101	188	172	228	54	17	5	103	224	156	244	169	25	23	204
164	135	60	232	227	79	255	19	190	153	52	166	215	39	115	111
142	151	11	125	184	124	21	178	76	113	98	132	86	159	175	144
110	177	209	38	194	15	83	251	189	163	202	81	41	128	84	161
36	147	10	61	148	117	118	158	67	99	170	91	58	44	168	95

Le tableau 3.6 représente une S-Box construite pour les conditions initiales ($x = 0$, $y = 220$, $z = 62100$).

Tableau 3. 6. S-Box construite à partir de l'équation discrétisé de Lorenz.

56	40	103	65	92	225	69	136	21	33	177	97	100	146	17	198
144	36	137	124	246	60	254	37	91	41	201	178	16	9	188	135
158	4	153	223	98	205	53	10	109	230	67	151	74	15	229	142
131	191	75	224	31	62	128	85	219	86	57	202	166	241	68	167
1	248	249	207	217	240	79	179	64	213	169	28	44	95	80	245
115	2	93	132	139	238	206	192	197	70	159	47	212	125	183	181
173	48	87	101	14	111	78	25	152	156	145	185	232	247	148	162
164	43	18	129	234	243	50	236	233	218	133	134	59	253	45	244
20	123	140	11	73	237	182	203	143	187	29	141	176	138	121	216
84	155	7	49	30	126	175	239	204	66	89	114	42	251	190	32
171	5	117	0	63	235	112	168	122	199	149	189	77	35	214	227
186	194	208	165	200	196	242	250	52	6	107	174	3	172	209	120
118	231	130	193	106	150	34	76	180	24	116	46	27	154	55	83
163	13	12	104	228	222	110	26	220	102	108	215	105	88	71	221
38	54	96	8	157	147	58	90	211	210	161	119	82	184	195	39
255	19	23	94	51	61	170	99	81	127	226	22	72	252	113	160

3.4.3. S-Box inverse

Pour les processus de décryptage, on utilise les S-Box inverses dans la fonction de substitution inverse. L'inverse des S-Box chaotique proposées se calcule de la même façon que la S-Box statique de l'AES standard. Etant donné que la fonction de substitution est bijective, les S-Box inverses chaotiques peuvent se calculer à l'aide du pseudo code suivant :

```
%%Création de S-Box inverse%%

 for i=1:256

      inv_sbox1(sbox1(i)+1)=i-1

 end

inverse_sbox1=reshape(inv_sbox1, 16, 16)

disp_hex ('    inv_sbox1 : ', inverse_sbox1)
```

Figure 3. 9. Pseudocode Matlab de la construction des S-Box chaotiques inverses.

3.5. Analyse de Sécurité

Dans plusieurs articles, certaines propriétés sont considérées comme des critères de conception des S-Box nécessaires pour vérifier l'aspect cryptographique. Après que Biham et Shamir [62] ont introduit la cryptanalyse différentielle pour les cryptosystèmes comme DES, Dawson et Tavares ont élargi les critères de conception des S-Box basées sur la théorie de l'information et ont révélé comment les S-Box peuvent avoir une immunité contre les attaques différentielles [56].

3.5.1. Critère d'avalanche Stricte (SAC)

L'effet avalanche est une propriété cryptographique nécessaire pour les algorithmes de chiffrement par bloc qui permet de s'assurer qu'un cryptosystème n'est pas susceptible aux attaques statistiques et qu'une petite différence à son entrée engendre un changement significatif à la sortie. La progression des données à crypter dans la structure du cryptosystème accroit ce changement afin d'obtenir des entrées/sorties décorrélées.

Le critère d'avalanche stricte a été introduit par A. F. Webster et S. E. Tavares en 1985 : « If a cryptographic function is to satisfy the strict avalanche criterion, then each output bit should change with a probability of one half whenever a single input bit is complemented ».
Il concerne les fonctions non linéaires appliquées à la cryptographie. Toute fonction qui satisfait ce critère doit avoir une probabilité de changement de un demi des bits de sortie, si un seul bit à l'entrée change. La sortie sera alors uniforme et aucune prédiction statistique ne peut avoir lieu [63], [64], [45], [65].

On peut définir ce critère de la manière suivante :
Soit x et c_i deux vecteurs de n bits, tels que x et c_i diffèrent seulement en un bit i.
La fonction $f(x)$ est une fonction booléenne, δ est la somme des bits changés tel que :

$$\delta = \sum_{x \in Z_2^n} f(x) \oplus f(x \oplus c_i) \tag{3.23}$$

Une fonction booléenne $f(x)$ qui vérifie le critère SAC si et seulement si $\delta = 2^{n-1}$ pour tout i avec $0 \le i \le n-1$

Pour cela, on doit calculer une matrice de dépendance A [65] telle que :

1- Un vecteur aléatoire x de n-bits est généré avec son vecteur crypté de m-bits, y = f(x).
2- Construire l'ensemble de n vecteurs ($x_1, x_2, \ldots, x_n$) tel que x et x_j diffèrent d'un seul bit pour tout $1 \le j \le n$.
3- On crée l'ensemble de vecteurs ($y_1, y_2, \ldots, y_n$) tel que $y_j = f(x_j)$.
4- L'ensemble des vecteurs d'avalanche ($v_1, v_2, \ldots, v_n$) est obtenu tel que :

$$v_j = y \oplus y_j \tag{3.24}$$

5- La valeur du $i^{ème}$ bit de v_j est ajoutée à l'élément a_{ij} de la matrice $m \times n$ de dépendance A.
6- Cette procédure est répétée pour un grand nombre k de vecteurs aléatoires x, et chaque élément de la matrice A est divisée par k.

7- Chaque élément a_{ij} donne une idée sur la relation entre le bit j du texte clair et le bit i du texte crypté.
8- La valeur 1 indique que si le bit j est complémenté, alors le bit i devra changer sa valeur, tandis que la valeur 0 indique que le bit du texte crypté est complètement indépendant du bit du texte clair.
9- Si tout élément de la matrice A, a une valeur non nulle alors la transformation cryptographique est complète, et si chaque élément a une valeur proche de 0.5, alors la transformation cryptographique satisfait le critère SAC.

On donne les matrices de dépendance obtenue pour :

- la S-Box basée sur la suite PWLCM en utilisant la méthode du paragraphe 3.4.1.1.
- la S-Box basée sur la combinaison des deux suites en utilisant la 1ère méthode et la deuxième méthode.
- La S-Box basée sur la suite de Lorenz (x = 0 ; y = 220; z = 62100).
- La S-Box basée sur la N-Logistic-Tent suite (y = 30000 x = 1200).

Tableau 3. 7. Matrice de dépendance de la PWLCM S-Box.

0.5176	0.5020	0.4549	0.5961	0.4863	0.5647	0.4706	0.5176
0.4863	0.5176	0.4549	0.4863	0.5490	0.4863	0.5490	0.5961
0.5647	0.5333	0.5333	0.5647	0.5647	0.5176	0.5020	0.4549
0.5647	0.5490	0.5020	0.5176	0.4863	0.5490	0.5647	0.5333
0.4549	0.4235	0.5647	0.4235	0.5804	0.4706	0.5176	0.5333
0.4549	0.5490	0.5490	0.4549	0.5176	0.5647	0.5176	0.5490
0.3765	0.4078	0.4706	0.4863	0.5020	0.4706	0.4549	0.4863
0.4549	0.4863	0.6118	0.4392	0.5490	0.5333	0.5647	0.5020

Tableau 3. 8. Matrice de dépendance de la S-Box des deux suites combinées utilisant la 1ère Méthode.

0.4392	0.5490	0.4392	0.3765	0.4706	0.5020	0.4863	0.5020
0.4392	0.5176	0.5333	0.5647	0.5490	0.4549	0.4863	0.5804
0.5020	0.5333	0.5490	0.5490	0.4706	0.5176	0.5020	0.5333
0.4863	0.4549	0.4392	0.5804	0.5804	0.4235	0.4549	0.4863
0.4706	0.4863	0.5020	0.5804	0.4549	0.4706	0.5647	0.4235
0.5333	0.5490	0.4863	0.4863	0.5333	0.5647	0.5490	0.4549
0.4863	0.5804	0.4863	0.5176	0.5020	0.5176	0.4392	0.5333
0.5020	0.4392	0.4706	0.4863	0.5176	0.5176	0.4392	0.4549

Tableau 3. 9. Matrice de dépendance de la S-Box des deux suites combinées utilisant la 2ème Méthode.

0.5804	0.4706	0.5490	0.5020	0.5333	0.5020	0.4549	0.5804
0.5020	0.4392	0.4863	0.5020	0.5020	0.4549	0.5176	0.4863
0.4706	0.5020	0.4863	0.5020	0.5020	0.5176	0.4863	0.4392
0.5176	0.6118	0.4863	0.5333	0.5020	0.5020	0.4392	0.4863
0.4078	0.4392	0.5333	0.5490	0.4706	0.5333	0.5647	0.5333
0.5490	0.5647	0.4706	0.5333	0.4706	0.5333	0.5490	0.4706
0.5333	0.4706	0.5176	0.4549	0.4863	0.5333	0.4549	0.4392
0.5647	0.5020	0.4706	0.4392	0.4549	0.5020	0.5176	0.4235

Tableau 3. 10. Matrice de dépendance de la S-Box basée sur la suite de Lorenz.

0.4549	0.5020	0.5176	0.5020	0.5647	0.4549	0.4706	0.5020
0.5490	0.4863	0.5333	0.5176	0.6118	0.5020	0.5333	0.4549
0.4863	0.5176	0.5176	0.5020	0.4392	0.5804	0.4549	0.5647
0.5176	0.5647	0.5490	0.5333	0.4549	0.5020	0.5176	0.5333
0.5020	0.5020	0.5490	0.5020	0.5020	0.5020	0.5490	0.4863
0.5490	0.4235	0.5333	0.4863	0.4706	0.4706	0.4863	0.5176
0.5176	0.4392	0.4392	0.5333	0.5333	0.4863	0.5020	0.4549
0.5333	0.5333	0.4235	0.5020	0.5490	0.5020	0.4706	0.5020

Tableau 3. 11. Matrice de dépendance de la S-Box basée sur la suite N-Logistic-Tent.

0.5020	0.5333	0.5176	0.5020	0.5020	0.5647	0.4235	0.5176
0.5176	0.5333	0.4706	0.5020	0.5647	0.5020	0.4863	0.5020
0.4235	0.5020	0.6118	0.5490	0.4706	0.5490	0.5020	0.5490
0.4706	0.4706	0.5020	0.5804	0.5020	0.4863	0.4706	0.5333
0.5647	0.5490	0.5333	0.5020	0.5333	0.5020	0.5020	0.5490
0.4706	0.4863	0.4863	0.5020	0.4863	0.3922	0.4863	0.4706
0.4863	0.4863	0.5490	0.4706	0.4863	0.5804	0.4863	0.5647
0.5333	0.5490	0.5490	0.5647	0.4863	0.5176	0.5176	0.5176

Les valeurs moyennes des matrices des tableaux 3.7 à 3.11 sont respectivement: 0.5103, 0.4993, 0.4998, 0.5054 et 0.5105 ; donc les S-Box satisfont le critère SAC puisque ces valeurs sont proches de la valeur moyenne 0.5. Le tableau 3.12 regroupe les valeurs moyennes des matrices de dépendance de suite PWLCM, 3D ainsi que leur combinaison en utilisant les deux méthodes citées auparavant dans 3.4.1.4. On peut remarquer que les S-Box basées sur la suite PWLCM ainsi que la combinaison de deux suites (PWLCM et 3D) satisfont le plus le critère d'avalanche stricte. De plus, ces résultats semblent être meilleurs que ceux obtenus dans les références [43], [53], [55], [60] et [61] ainsi que la S-Box statique (de l'AES) qui sont respectivement 0.4972, 0.5125, 0.4993, 0.5048, 0.4923 et 0.5069.

Tableau 3. 12. Critère d'avalanche stricte.

Récurrence chaotique	**SAC[1]**	**SAC[2]**
3D $x_0 = 0.8$; $y_0 = 0.5$; $z_0 = 0.1$;	0.5044	0.5051
PWLCM $p = 0.15$; $x_0 = 0.2$	0.5049	0.5
Combinaison de 2 suites $p = 0.15$; $x_0 = 0.7$;	0.4993	0.4998

1 et 2 sont la première et la deuxième méthode.

3.5.2. Nonlinéarité / Linéarité

La cryptanalyse linéaire consiste à extraire la clé en faisant des approximations linéaires précises du cryptosystème. La précision des approximations est reliée au nombre important des pairs textes clairs/textes cryptés.

La S-Box est le seul élément non linéaire de l'AES, on essaie donc de construire des équations linéaires reliant l'entrée avec la sortie et d'énumérer toutes les approximations linéaires de la S-Box dans la table d'approximation linéaire. Chaque élément de la table représente le nombre d'égalités par rapport à la moitié des entrées possibles entre l'équation linéaire dans la "input sum" (colonne) et l'équation linéaire de la "output sum" (ligne).
Diviser chaque élément de la table par le nombre d'éléments donne sa probabilité.

Les valeurs nulles de la table, indiquent un ensemble non linéaire, et les valeurs absolues importantes indiquent des ensembles linéaires/affines ou proches de linéaires/affines.

La mesure de non linéarité se fait avec la probabilité d'approximation linéaire (LP) définie dans [66] par :

$$L_p = \max\left|P_{a,b} - \frac{1}{2}\right| \tag{3.25}$$

$$P_{a,b} = \# \frac{\{x / x.a = S(x).b\}}{2^n} \tag{3.26}$$

x.a signifie le calcul de parité du produit binaire de *x* et *a*.

La probabilité d'une entrée (a, b) dans une table d'approximation linéaire est la partie des entrées x tel que :

$$x.a \oplus S(x).b = 0 \tag{3.27}$$

Cette probabilité est alors égale à 1/2 plus la valeur de la table divisée par le nombre des entrées possibles.

Dans la cryptanalyse linéaire, on étudie le comportement des parités des sous-ensembles des bits durant le cryptage. On veut retrouver les informations statistiques sur la parité des bits chiffrés et des bits des clés connaissant la parité de l'entrée.

La probabilité que l'approximation dans une S-Box soit valide, est donnée comme étant la distance par rapport à la moitié. Si une valeur de la table est – 20, on aura alors:

$$p' = \frac{12}{64} = \frac{1}{2} - \frac{20}{64}.$$

Une valeur nulle a une probabilité p'=1/2, cette entrée est inutile pour construire une attaque. Toute autre entrée qui a une valeur différente de 0 peut être utilisée.

Une approximation peut inclure plus qu'une S-Box, exemple dans le cas du mode CCM.

La probabilité d'approximation avec 2 S-Box est :

$$p_1^{'}.p_2^{'} + (1 - p_1^{'})(1 - p_2^{'}) \tag{3.28}$$

On note : $p_i^{'} = \frac{1}{2} + p_i$

La probabilité combinée est donnée par l'expression suivante :

$$p + \frac{1}{2} = \frac{1}{2} + 2p_1p_2 \tag{3.29}$$

En général, pour *l* S-Box, on a :

$$p + \frac{1}{2} = \frac{1}{2} + 2^{l-1}\prod p_i \tag{3.30}$$

Lorsqu'une approximation linéaire avec une probabilité $\frac{1}{2} + p$ est connue à l'attaquant, il peut faire une attaque avec p^{-2} textes connus. Ces textes clairs peuvent être aléatoires mais ils doivent être cryptés avec la même clé, et les textes cryptés doivent être connus par l'attaquant [67].

La formule suivante est une approximation de l'équation (3.25) donnée par Jakimoski:

$$Lp = \max_{a,b \neq 0}\left(\frac{\#\{x \in X \;\; x \bullet a = f(x) \bullet b\} - 2^{n-1}}{2^{n-1}}\right)^2 \tag{3.31}$$

Le tableau 3.13 regroupe la probabilité d'approximation linéaire de chaque S-Box créée par chaos en utilisant les deux méthodes que nous avons proposées dans la sous-section 3.4.1.4.

On remarque que la S-Box basée sur la combinaison des deux suites (3D et PWLCM) par la 1ère méthode a la plus faible L_p, elle est plus sécurisée contre les attaques linéaires. Ce résultat est égal à celui de Jakimoski dans [43] et meilleure que ceux dans [55], [56] et [45] dont les valeurs sont respectivement 0.0706, 0.0706 et 0.088135.

La S-Box créée à partir de la suite N-Logistic-Tent a une Lp égale à 0.0881.

Par contre, la probabilité d'approximation linéaire pour la S-Box générée par l'équation de Lorenz pour les conditions initiales (x = 0 ; y = 220 ; z = 62100) dépasse les résultats précédents avec une valeur égale à 0.0976.

La probabilité d'approximation linéaire de la S-Box créé par chaos est comparable à celle des S-Box statiques de l'AES. Ces valeurs varient de (2^{-2} à 2^{-3})

Cette probabilité diminuera en fonction du nombre d'itérations effectuées dans l'algorithme de cryptage. Par conséquent, réduire L_P revient à augmenter la complexité de l'attaque linéaire.

Tableau 3. 13. Approximation de probabilité linéaire.

Récurrence chaotique	L_P^1	L_P^2
3D $x_0 = 0.8$; $y_0 = 0.5$; $z_0 = 0.1$;	0.0625	0.0881
PWLCM $p = 0.15$; $x_0 = 0.2$	0.0791	0.0705
Combinaison de 2 suites $p = 0.15$; $x_0 = 0.7$;	0.0625	0.0705

1 et 2 sont la première et la deuxième méthode.

3.5.3. Distribution équiprobable des différentielles d'entrée /sorties

La cryptanalyse différentielle a été introduite par Biham et Shamir dans l'article [62].
Cette attaque est une méthode générale pour casser les algorithmes de cryptage par bloc ou les fonctions de hachages. Elle exploite la prévisibilité de la propagation de la différence de texte clair choisi. L'analyse des différences des paires de textes claires et des différences des paires de textes chiffrés correspondantes mènera à extraire la clé ou à diminuer l'espace des clés et à entamer une attaque par force brute rapide. La différence entre les paires de texte est généralement réalisée par la fonction OU exclusif (XOR). On nomme les *différentielles* les différences entre les paires de textes chiffrés de sortie..

Leurs propriétés statistiques dépendent de la nature des S-Box de l'algorithme de chiffrement. Pour chaque S-Box, l'attaquant peut calculer une paire de différentielles (Δx, Δy) avec :

$\Delta y = S(x) \oplus S(x \oplus \Delta x)$: La différence en sortie.
Δx : La différence appliquée au texte en entrée.

Dans le cas idéal, la transformation non linéaire S-Box doit avoir une uniformité différentielle. A une entrée différentielle Δx lui correspond une sortie unique différentielle Δy assurant ainsi une probabilité uniforme d'application pour chaque i. La complexité d'une attaque différentielle est déterminée par le maximum de probabilité différentielle donnée par l'équation suivante:

$$DP_f = \max_{\Delta x \neq 0, \Delta y} \left(\frac{\#\{x \in X / f(x) \oplus f(x \oplus \Delta x) = \Delta y\}}{2^n} \right) \quad (3.32)$$

Où x est l'ensemble de toutes les valeurs d'entrée possibles et 2^n est le nombre de ces éléments. Δx est la différence d'entrée, et Δy est la différence de sortie.

La complexité de l'attaque diminue si la valeur du maximum de probabilité différentielle augmente. En calculant DP_f pour les S-Box, on peut estimer la robustesse contre les attaques différentielles.
Pour la méthode citée dans 3.4.1.1, le maximum de probabilité est de 10/256, ce qui est bien inférieur à la DP_f dans le cas de l'article [43] qui est égal à 12/256.
Nous avons calculé en premier lieu la fréquence des occurrences des sorties XORées les plus probables pour les S-box basées sur la combinaison des suites. Le tableau 3.14 montre que les sorties les plus probables pour la première méthode sont 6 et 8 qui se produisent 154 et 86 fois respectivement. Le tableau 3.15 révèle les mêmes sorties les plus probables et ceci pour la deuxième méthode avec une fréquence de 161 et 80 respectivement.

On en déduit les maxima de probabilité différentielle regroupés dans le tableau 3.16 des S-Box construites selon 3.4.1.4. Le résultat pour la combinaison des deux suites est égal à 10/256.
La fréquence des occurrences des sorties XORées les plus probables des deux méthodes basées sur la suite de Lorenz et sur la suite N-Logistic-Tent proposées dans la section 3.4.2.2 est présentée respectivement dans les tableaux 3.17 et 3.18. Les sorties les plus probables des tableaux 3.17 et 3.18 sont aussi 6 et 8 avec des fréquences respectives 161, 78 pour le tableau 3.17 et 161, 80 pour le tableau 3.18. Cependant, le maximum de probabilité différentielle dans les deux cas reste 10/256.
Nous déduisons que pour les S-Box que nous avons proposées à l'aide des suites chaotiques à sortie réelle ou bien à sortie entière, le maximum de probabilité différentielle est plus faible de celui de Jakimoski's [43] et Chen's [56] avec $(2^{-5}) < Dp_f = (12/256) < (2^{-4})$ et égal aux résultats trouvés dans les références [53], [54], [55], [60] et [61].

Cependant, la distribution des différentielles n'est pas uniforme comme pour le cas des AES S-Box pour laquelle la fréquence d'occurrence des sorties XOR les plus probables est toujours 4. Ces probabilités d'approximation différentielles que nous avons trouvées sont légèrement supérieures à celle de l'AES, mais nous pouvons dans une certaine mesure dire que ce sont des valeurs comparables à la DP de la S-Box statique de l'AES avec des valeurs de DP typiques variant entre 2^{-4} et 2^{-6}.

Tableau 3. 14. La fréquence d'occurrence des sorties XOR les plus probables pour la 1ère méthode.

-	8	6	6	10	6	6	6	6	6	6	8	6	6	8	6
6	8	8	6	6	8	6	6	6	6	6	4	8	6	6	8
6	6	8	6	8	8	8	6	6	8	6	6	10	6	8	6
8	8	4	6	8	8	8	6	6	6	6	6	6	6	8	8
6	8	6	8	6	6	6	6	6	6	6	6	8	8	6	6
8	6	6	8	8	8	6	8	6	10	6	6	8	6	8	8
6	8	6	4	6	6	8	6	6	6	6	6	6	8	6	6
6	6	6	6	8	8	8	8	6	6	6	6	8	6	8	6
6	8	6	10	8	6	8	6	10	6	8	6	6	6	8	8
4	8	6	6	6	6	6	8	6	6	6	6	6	8	6	6
6	6	6	6	10	6	8	6	8	6	8	6	6	6	8	8
6	8	8	6	8	8	6	6	8	8	6	6	8	8	6	8
4	6	6	6	6	6	8	10	6	8	6	6	8	6	6	6
6	8	8	8	10	6	6	6	8	8	6	8	6	8	8	6
6	10	6	8	6	8	6	6	6	8	6	10	6	8	8	8
8	8	6	6	6	6	6	6	6	6	6	6	6	8	8	6

Tableau 3. 15. La fréquence d'occurrence des sorties Xor les plus probables pour la 2ème méthode.

-	6	6	8	8	8	8	6	6	8	8	6	8	6	8	6
6	8	6	6	6	6	8	6	6	6	6	6	6	10	10	6
6	6	6	6	6	6	6	8	8	8	6	8	8	6	6	8
4	6	6	6	8	6	6	6	6	8	6	8	6	6	6	6
6	6	6	8	8	6	6	6	8	8	8	8	6	8	8	6
6	6	6	8	6	6	6	6	6	6	6	10	6	8	6	6
8	6	6	6	8	6	8	6	8	6	6	6	8	6	6	6
6	8	8	6	6	8	8	6	8	8	8	6	8	10	8	6
6	6	6	8	8	6	6	4	6	8	6	6	8	6	6	6
8	8	6	6	6	6	8	6	6	8	6	4	6	8	6	6
6	6	8	6	6	6	6	8	6	6	6	8	8	8	6	6
6	6	8	4	6	6	6	6	6	10	6	6	6	8	6	8
6	6	8	6	6	6	8	8	6	6	8	6	8	6	6	6
10	6	6	6	8	6	8	6	8	6	8	8	6	10	10	6
8	6	6	6	8	10	8	6	6	6	6	6	8	6	6	6
10	6	6	6	8	8	6	6	6	8	8	6	8	6	8	6

Tableau 3. 16. Approximation de Probabilité Differentielle.

Chaotic map	**DF[1]**	**DF[2]**
3D $x_0 = 0.8$; $y_0 = 0.5$; $z_0 = 0.1$;	12/256	12/256
PWLCM $p = 0.15$; $x_0 = 0.2$	10/256	12/256
2 maps combination $p = 0.15$; $x_0 = 0.7$;	10/256	10/256

1 et 2 sont la première et la deuxième méthode

Tableau 3. 17. La fréquence d'occurrence des sorties XOR les plus probables pour la S-Box basée sur l'équation de Lorenz.

-	6	8	6	6	6	8	8	6	6	6	8	6	8	8	6
6	8	8	8	6	6	8	8	8	6	6	6	4	8	8	6
8	8	8	6	6	10	8	8	6	6	6	10	6	8	6	8
6	8	6	6	8	6	6	8	6	6	8	6	6	6	8	6
6	6	10	6	8	6	6	6	6	6	6	6	6	6	6	8
8	8	6	6	6	6	6	8	6	8	6	8	6	8	6	8
6	6	6	6	6	6	6	6	10	6	6	6	6	8	8	8
6	6	10	10	10	8	8	6	6	8	6	6	8	6	6	6
8	8	6	6	6	6	6	8	6	8	8	10	8	6	6	6
6	6	6	6	6	6	6	6	8	8	6	6	6	10	6	6
6	4	6	6	6	6	8	6	6	6	8	6	6	6	6	6
6	8	6	6	6	6	6	8	6	6	6	6	8	6	8	8
6	6	6	6	8	6	8	6	6	6	6	6	10	6	6	8
6	8	8	6	6	6	8	8	4	6	6	8	6	6	8	4
6	4	6	8	6	8	6	6	6	6	8	6	6	4	8	8
8	6	8	8	8	8	8	6	6	6	6	6	6	6	8	6

Tableau 3. 18. La fréquence d'occurrence des sorties XOR les plus probables pour la S-Box basée sur la suite N-Logistic-Tent

-	8	8	6	8	6	6	6	6	8	6	10	8	6	6	6
8	6	4	6	8	6	8	6	6	6	6	6	8	6	6	6
6	6	6	8	6	6	6	8	8	8	6	6	6	6	6	6
8	6	6	6	8	8	6	8	6	6	6	8	8	6	8	6
8	8	6	10	8	8	6	6	8	6	6	8	8	8	6	8
10	6	6	6	6	8	6	8	8	8	8	8	8	8	6	8
8	8	6	8	6	6	6	6	6	6	8	6	10	6	6	8
8	6	6	8	8	6	6	6	6	6	6	6	6	6	8	8
6	6	8	6	6	6	8	8	6	8	6	6	6	8	6	6
6	6	6	8	6	6	6	4	6	8	6	6	6	8	6	8
6	6	8	6	8	6	8	6	6	6	10	6	6	6	6	8
6	8	8	8	6	8	6	6	6	6	6	6	6	6	8	8
8	6	8	6	6	6	6	6	8	6	8	6	6	6	6	6
8	10	8	10	6	6	6	6	6	6	6	8	6	6	8	6
6	6	8	8	8	8	6	10	6	8	8	4	6	6	6	6
6	6	6	6	8	6	8	6	6	6	6	6	6	6	6	8

3.5.4. Bijectivité

Pour vérifier la propriété bijective, une méthode est proposée par Adams et Tavares dans [63]. Si une fonction f_i vérifie l'équation suivante alors elle est bijective :

$$wt(\sum_{i=1}^{n} a_i f_i) = 2^{n-1} (\bmod 2) \tag{3.33}$$

Avec $a_i \in \{0,1\}$, wt() est le poids de hamming et $(a_1, a_2,\ldots,a_n) \neq (0,0,\ldots,0)$.

La bijectivité d'une fonction peut être vérifiée d'une façon plus simple, si pour un nombre d'entrée allant de e=0,1,2…,2^n-1, le nombre d'éléments de sortie est égal à 2^n alors la fonction est bijective. Pour nos méthodes de construction de S-Box proposées, la sortie est toujours 256 valeurs différentes de 0 à 255 (256=2^8), nous avons appliqué l'équation 3.33 sur les différentes S-Box et nous avons obtenu pour chacune une valeur égale à 128. Pour la S-Box proposé par Tang le poids de hamming est égal à 129. Nous pouvons déduire alors que les fonctions de substitution que nous avons conçues sont bijectives.

3.5.5. Discussion des inconvénients des cryptosystèmes chaotiques numériques

Bien que les algorithmes chaotiques numériques proposés dans la littérature ne dépendent pas de la synchronisation et que leurs clés soient essentiellement basées sur les conditions initiales et les paramètres de contrôle des récurrences chaotiques, ils présentent certaines failles. Ces failles exigent une attention particulière accordée aux types des récurrences et à l'environnement d'implémentation étant donné que nous avons choisi comme application les réseaux de capteurs sans fil et non pas un simple ordinateur à 32 ou 64 bits.

En effet, les attracteurs chaotiques de certaines suites chaotiques montrent des fenêtres périodiques non adéquates pour le choix des paramètres et constituent ainsi des clés non robustes. L'algorithme proposé par Pareek [38] a été cryptanalysé dans [68] parce que des valeurs des paramètres de la suite logistique utilisées dans la clé appartenaient à une fenêtre de période 3.

Pour les réseaux de capteurs sans fil, l'algorithme proposé par Chen [48] a été cryptanalysé dans [49].

La représentation en nombre fini de la sortie des suites chaotiques dissimule le problème de bruit et des divergences des paramètres de contrôle. Cependant, elle est la cause des redondances et des cycles courts et non prévisibles.

Les systèmes chaotiques fonctionnent dans le domaine réel ; la transformation réel-entier est nécessaire. Ce processus conduit à une dégradation du comportement chaotique du générateur et à un temps trop long durant l'exécution. Une étude récente montre que la suite logistique est inadéquate pour les applications cryptographiques [69], [70].

Si une suite est vraiment chaotique sur tout un intervalle I, l'occurrence d'un point périodique est restreinte à un sous intervalle de longueur nulle. Un ordinateur possède une précision finie et la condition de transitivité topologique d'un système dynamique chaotique ne sera pas alors vérifiée puisque ce système n'est pas situé dans un intervalle de nombres réels. Il est donc nécessaire d'adopter la définition d'un système dynamique dans un contexte discret ou de précision finie. Avec un ordinateur, les itérations donnent souvent un comportement périodique.

La question est comment étendre la définition du système dynamique chaotique à ces itérations discrètes.

Les erreurs de l'arrondi dans les calculs sont cumulatives, alors la vraie valeur x_n et celle calculée y_n sont différentes. Pour résoudre le problème, le « shadowing lemma » (voir annexe) réclame qu'il existe une trajectoire réelle (x_n) proche de la trajectoire (y_n), ainsi les résultats statistiques théoriques demeurent vrais [71].

Par contre, Silva et al dans [23] affirment qu'il est complètement impossible d'assurer le théorème d'ombre (shadowing lemma) en utilisant la précision finie.

Ce sont deux points de vue différents sur l'utilité et l'applicabilité du shadowing lemma dans le cas de précision finie.

Dans les cryptosystèmes basés sur la suite linéaire par morceaux, les états voisins peuvent être projetés sur le même segment de ligne. Dans ce cas, le recouvrement des valeurs sera plus facile en connaissant une partie du bloc texte clair-texte crypté [72].

La conséquence de ces faits conduit à la répétition de valeurs après le parcours de toutes les valeurs possibles.

Les cycles courts peuvent mener le système à tourner dans un nombre réduit de valeurs se stabilisant sur une orbite [73], [33]. On en parlera de ce problème avec plus de détails dans la dernière partie.

3.6. Conclusion

Nous avons identifié en premier lieu les classes des cryptosystèmes chaotiques qui se divisent en deux classes : analogique et numérique. Les inconvénients des cryptosystèmes analogiques, l'importance des S-Box dans plusieurs algorithmes de chiffrement conventionnels et récents ainsi que l'application souhaitée pour l'ensemble de nos travaux, nous ont orientés vers la proposition d'algorithmes de construction de S-Box dynamiques à base de récurrences chaotiques dans le but de les utiliser dans des algorithmes comme l'AES standard en remplaçant son S-Box statique. Nous avons prouvé la sécurité de nos S-Box contre les attaques classiques à travers des critères d'évaluation précis. Parmi les récurrences testées nous avons constaté que la combinaison de deux de ces récurrences (PWLCM et 3D), pour générer ces S-Box dynamiquement, engendre de meilleures propriétés statistiques.

Chapitre 4

Application sur les images et Implémentation sur un réseau de capteurs sans fil

« Si tu ne peux pas le mesurer, tu ne peux pas l'améliorer. »

Lord Kelvin

4.1. Introduction

Le développement inédit de la technologie d'information et de miniaturisation a accéléré la progression et la propagation d'un nouveau concept appelé : Les réseaux de capteurs sans fil. Un réseau de capteurs sans fil RCSF ou WSN (Wireless Sensor Network) est un réseau composé de nœuds actifs qui récoltent les données de l'environnement via une unité de captage puis les communiquent à l'aide des unités de traitement et de transmission.

L'applicabilité des réseaux de capteurs sans fil s'est étendue ces dernières années sur plusieurs domaines s'approchant de plus en plus de la vie quotidienne. Allant des applications militaires vers l'agriculture, le transport, l'environnement, etc, la sécurité des informations collectées et transmises est devenue impérative mais aussi critique [74].

Les préoccupations de sécurité des données dans un WSN sont confrontées à des capteurs avec un volume réduit, un espace de stockage limité et surtout une puissance de batterie restreinte. La conception d'un algorithme de chiffrement doit répondre à ces restrictions, ce qui crée un compromis entre la sécurité d'une part et la rapidité et la consommation d'énergie d'autre part.

Le type d'informations collectées par les capteurs sans fil peut être des températures, une luminosité ou aussi une image qui est une entité plus complexe à chiffrer.

Selon [75] et [76], les algorithmes de cryptage conventionnels tel que DES ou AES ne sont pas adéquats au cryptage d'image et vidéo. Ils argumentent ceci par les contraintes de rapidité en temps réel réduite de ces algorithmes. L'auteur dans l'article [77] montre que les images cryptées par l'algorithme AES restent encore intelligibles. Plusieurs recherches proposent de remplacer alors ces algorithmes par des algorithmes de chiffrement chaotique [75], [76], [77], [78].

Dans ce chapitre, deux sections sont présentées : Dans la première section, nous exposons dans un premier temps les principaux algorithmes de cryptage par chaos conçus pour les images ainsi que notre contribution là où nous avons focalisé sur l'introduction des opérations de diffusion et de confusion et sur la simplicité de l'algorithme proposé. Dans un deuxième temps, nous démontrons l'efficacité de nos contributions : algorithme de cryptage chaotique et l'AES modifié (S-Box dynamique chaotique au lieu de la S-Box statique), par une analyse de sécurité.

La deuxième section décrit l'implémentation sur un réseau de capteurs (WSN) dotés de microcontrôleurs 16 bits. Cette partie comporte deux phases : une phase simulation et une phase application réelle. Une étude sur la consommation d'énergie des approches proposées est exposée dans les deux cas.

4.2. Application sur le cryptage d'image et contributions

Les données multimédia sont en grande partie emmagasinées dans différents supports et échangées sur diverses sortes de réseaux. Souvent, ces données contiennent des informations privées ou confidentielles ou parfois même des intérêts financiers. Par exemple, les menaces de sécurité des Smartphones peuvent être contrées par l'identification biométrique. L'accès à ces Smartphones sera limité aux propriétaires qui seront identifiés par leur image d'identité.

Un échange sécurisé est assuré par des techniques qui garantissent l'intégrité, la confidentialité et l'authenticité. Le chiffrement d'image est une discipline largement étudiée et les dynamiques chaotiques par leur sensibilité aux conditions initiales sont un rival potentiel des crypto-systèmes conventionnels.

4.2.1. Présentation des algorithmes étudiés

Le premier algorithme que nous avons étudié est l'algorithme ECKBA décrit par figure 4.1. Il était proposé dans [75] et il était cité et décrit comme références pour plusieurs propositions d'algorithme de cryptage d'image [79], [80], [81], [82].

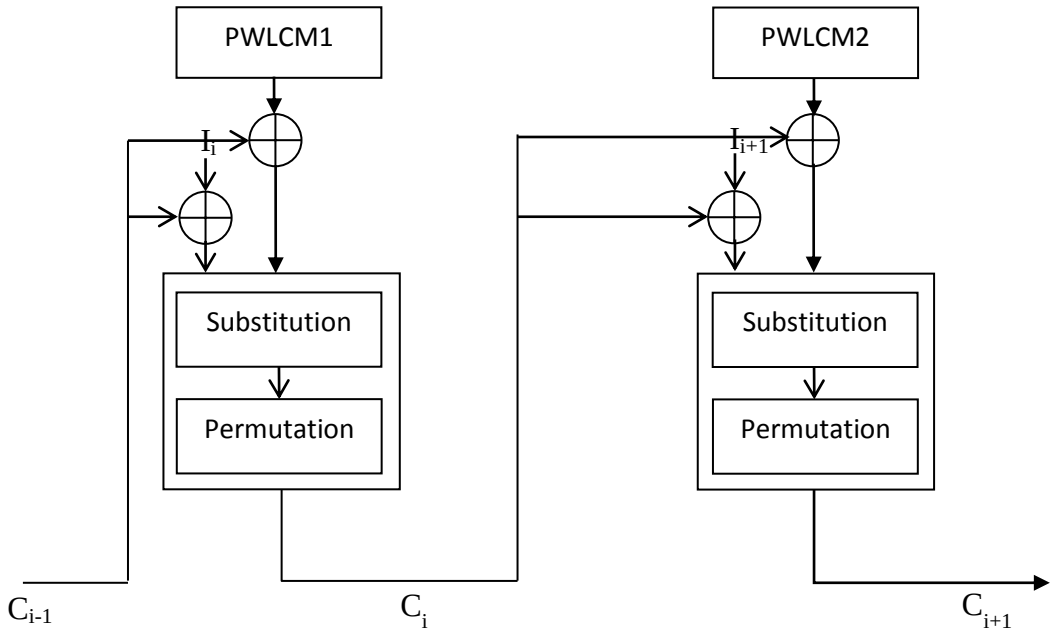


Figure 4. 1. Algorithme de cryptage ECKBA.

Il crypte une image I en utilisant un réseau de substitution-permutation (SP) contrôlé par la fonction linéaire par morceaux (PWLCM). Il effectue r tours du réseau SP sur chaque pixel. L'itération (i+1) de la fonction chaotique est contrôlée par le bloc chiffré précédemment C_i. Cet algorithme met en œuvre le mode de cryptage CBC (Cipher Block Chaining) : un XOR est appliqué entre le pixel I_i et le pixel crypté C_{i-1}.

Deux séquences chaotiques pseudo-aléatoires sont utilisées dans l'étape de substitution.
La S-Box et son inverse sont construites à partir des équations 4.1 et 4.2 :

$$\sigma_r(u,v) = \begin{cases} \mathrm{u} \oplus \mathrm{v}, & \text{si r est pair;} \\ \mathrm{u} + \mathrm{v} \mod 256, & \text{si r est impair,} \end{cases} \tag{4.1}$$

$$\sigma_r^{-1}(u,v) = \begin{cases} u \oplus v, & \text{si r est pair;} \\ u - v \mod 256, & \text{si r est impair,} \end{cases} \tag{4.2}$$

Où u et v constituent deux octets.

La P-Box est obtenue à partir de l'opération de permutation π_i qui est appliquée sur chaque bit de chaque bloc (pixels) avec i $\in$ (0,8 !) est l'indice de permutation ; exemple pour i=17331, le résultat de permutation de l'octet O = [b_1, b_2, b_3, b_4, b_5, b_6, b_7, b_8] sera $\pi(O)$ = [b_4, b_5, b_1, b_6, b_2, b_7, b_8, b_3].

L'utilisation des « Lookup Table » ou table de correspondance augmentera la rapidité du code mais les besoins en mémoire seront considérablement élevés [75]. Pour une application sur les réseaux de capteurs (avec une faible capacité de mémoire) la permutation des bits de chaque pixel sera calculée à chaque tour r.

La figure 4.2 illustre le cryptage de deux octets du deuxième algorithme que nous avons testé [78]. Les deux fonctions PWLCM sont utilisées pour obtenir deux nombres pseudo-aléatoires X_i et $Y_i \in [0,1]$. Les résultats sont perturbés par deux LFSR avec deux seeds (Conditions initiales). Le chiffrement est effectué par un XOR entre les sorties perturbées et le bloc d'entrée. Pour crypter 128 octets, il suffit de répéter le processus décrit 64 fois.

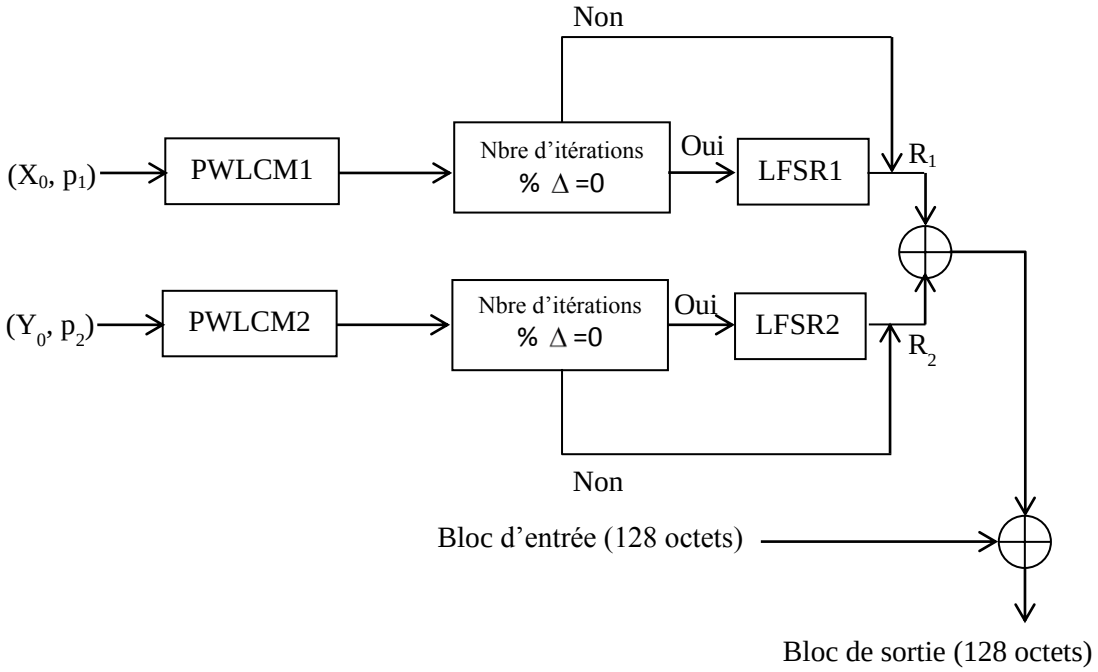


Figure 4. 2. Algorithme proposé par Mansour et al [78].

4.2.2. Contribution : Algorithme de Cryptage d'Image basé sur la suite de Lorenz (ACIL)

Certes la sécurité des images constitue une phase importante au cours de leur transmission mais la simplicité, la rapidité et la consommation des ressources du système restent un challenge à emporter surtout pour des applications orientées aux réseaux de capteurs. La technique de diffusion est souvent réalisée par des permutations de blocs de pixels ou des permutations binaires [83], [13]. Les exemples des algorithmes qui appliquent des

permutations sont divers, on cite : DES, AES, RC4, etc. Nous avons choisi alors d'utiliser une fonction de permutation simple pour assurer la diffusion. La confusion dans notre algorithme est réalisée par le biais d'une suite chaotique.

Le choix de la suite chaotique était basé sur le fait que pour un réseau de capteurs, la puissance de calcul est limitée et la précision aussi ; ce qui fait que les opérations en virgule flottante ne peuvent pas être gérées en totalité. Pour cette raison, nous avons sélectionné la suite de Lorenz discrétisée où les variables et les paramètres sont des entiers.

Cette suite a comme entrées trois variables dynamiques (x_k, y_k et z_k). Les données claires sont donc cryptées par bloc de 3 octets chacun. La taille du bloc peut être aussi un multiple de 3 puisque la taille des variables dynamiques peut être étendue à plus qu'un octet en utilisant des entiers non signés.

Les étapes principales de notre algorithme proposé ACIL sont les suivantes :

- **Etape 1 :** un XOR entre des portions du bloc d'entrée et une partie de la clé est appliqué pour augmenter la complexité des attaques.

$$\begin{cases} x_0 = x_0 \oplus Entrées(i); \\ y_0 = y_0 \oplus Entrées(i+7); \\ z_0 = z_0 \oplus Entrées(i+15); \end{cases} \tag{4.3}$$

Tel que l'ensemble {x_0, y_0, z_0} constitue les conditions initiales de la suite de Lorenz discrétisée.

- **Etape 2 :** Comme nous l'avons cité au paragraphe précédent, la permutation lexicographique consomme beaucoup de ressources en mémoire dans le cas des « lookup table » et semblent ajouter un temps supplémentaire dans le cas où la permutation des bits se fait au fur et à mesure. Nous avons opté alors pour une permutation cyclique à gauche des pixels comme le montre la figure 4.3. La première ligne reste intacte, les éléments de la deuxième ligne sont décalées à gauche par 1, ceux de la 3ème ligne par deux et enfin la 4ème ligne par 3. La permutation cyclique à gauche est utilisée dans l'AES dans la fonction shiftrow.

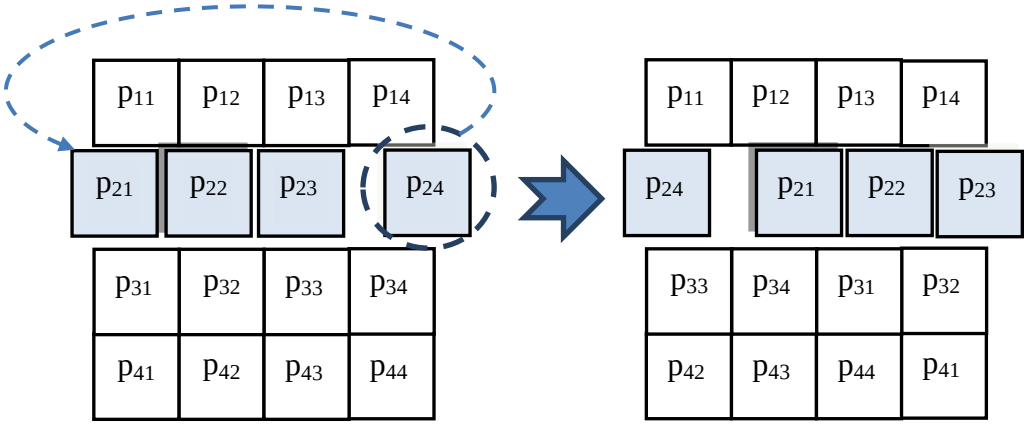


Figure 4. 3. Permutation cyclique des pixels.

- **Etape 3 :** Nous appliquons le mode CBC par un XOR entre un bloc de pixels permutés et un bloc de pixels cryptés précédemment :

$$\begin{cases} C_p(i) = C(i-1) \oplus PixPermuté(i), \\ C_p(i+1) = C(i) \oplus PixPermuté(i+1), \\ C_p(i+2) = C(i+1) \oplus PixPermuté(i+2); \end{cases} \quad (4.4)$$

L'avantage d'employer le mode CBC est de ne pas pouvoir déduire que le même message a été envoyé avec une écoute passive du réseau. Toutefois, le décryptage séquentiel et non pas parallèle reste son inconvénient. .

- **Etape 4 :** Itérer la suite de Lorenz discrétisée et perturbée par un LFSR, donnée par les équations (3.10) et (3.11) du chapitre précédent. Chaque itération génère 3 variables x_k, y_k et z_k. Appliquer par la suite un XOR entre le résultat de l'étape précédente et les sorties de la suite de Lorenz :

$$\begin{cases} C(i) = C_p(i) \oplus (x_k \oplus y_k), \\ C(i+1) = C_p(i+1) \oplus (x_k \oplus z_k), \\ C(i+2) = C_p(i+2) \oplus z_k; \end{cases} \quad (4.5)$$

- **Etape 5 :** un XOR est appliqué entre des portions de la clé et les octets cryptés C.

$$\begin{cases} C(i) = \Delta t \oplus C(i); \\ C(i+1) = x_0 \oplus C(i+1); \\ C(i+2) = z_0 \oplus C(i+2); \end{cases} \quad (4.6)$$

- **Etape 6 :** Répéter les étapes 4 et 5 jusqu'à ce que toute l'entrée claire soit cryptée.

Les étapes 1 et 5 constituent une variante du « key whitening » ou blanchissement de la clé, définie par l'application d'un XOR entre certains éléments de la clé avec l'entrée, et d'un autre XOR entre une autre clé avec la sortie [13].

Cette notion apparait pour la première fois avec l'algorithme DES-X développé par RSA, puis s'étend à plusieurs autres algorithmes comme FEAL où la même clé est utilisée à l'entrée et à la sortie. Le « whitening » oblige le cryptanalyste de deviner une des valeurs de l'opération du whitening en plus de la clé, il n'est pas vulnérable à l'attaque MITM et permet d'augmenter la sécurité de l'algorithme en question [13].

L'organigramme simplifié de notre algorithme (ACIL) que nous proposons est donné par la figure 4.4.

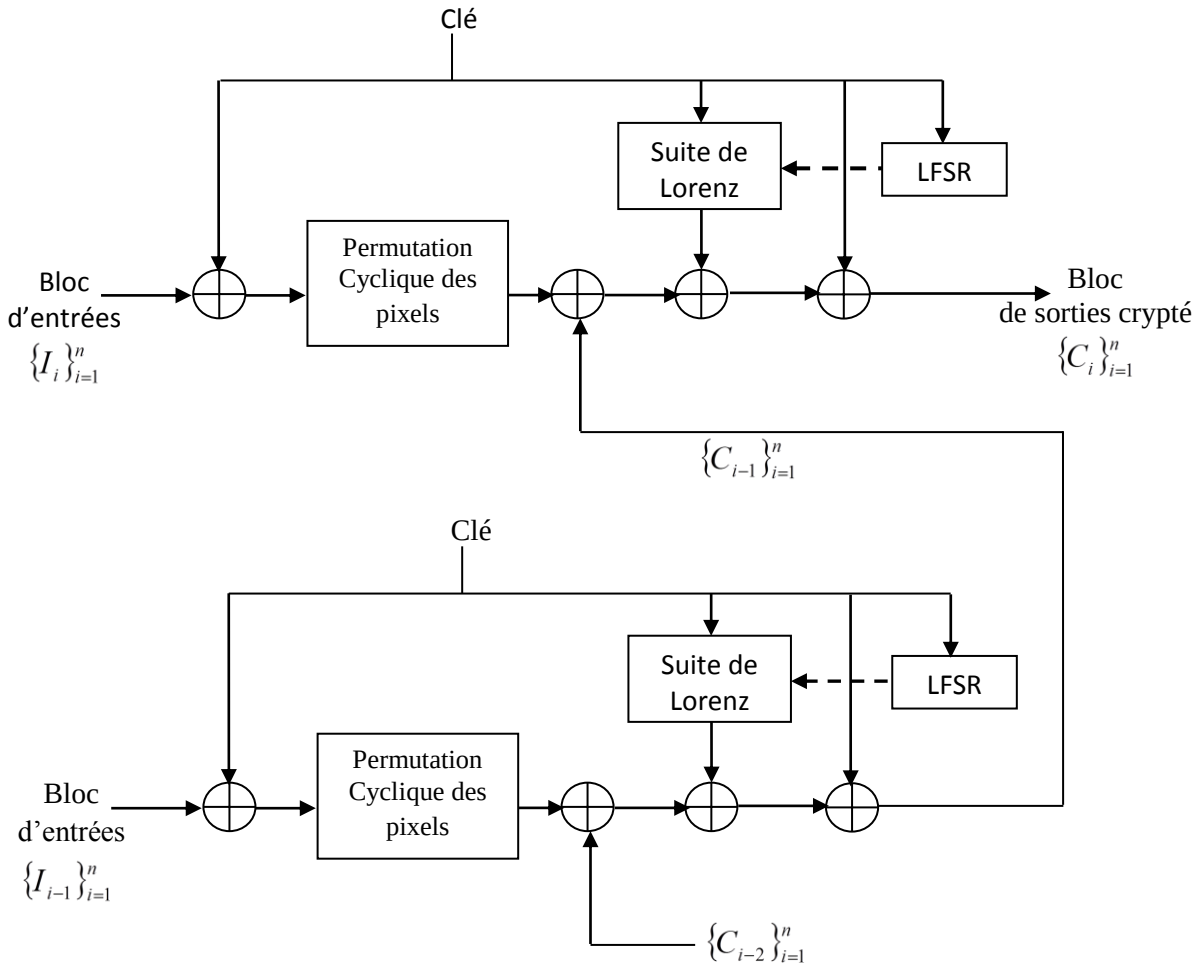


Figure 4. 4. Algorithme ACIL proposé en mode CBC.

4.2.3. Analyse de sécurité et résultats des tests

En appliquant un algorithme de chiffrement sur une image, les pixels de l'image chiffrée doivent être différents et indépendants (faible corrélation) de ceux de l'image originale. Ceci peut être visible par simple visualisation de l'image cryptée. La simple inspection visuelle reste insuffisante pour juger le chiffrement d'une image. On peut classer les métriques d'évaluation du degré de cryptage en analyse statistique, analyse différentielle et une analyse de la sensibilité et de l'espace de la clé secrète.

Dans tout ce qui suit l'image que nous avons crypté est l'image de lena de type PNG de taille 80×80.

4.2.3.1. Analyse statistique

L'analyse statistique permet de déchiffrer plusieurs algorithmes de cryptage comme l'a mentionné Shannon dans [84].

Durant cette partie, nous allons étudier les histogrammes des images cryptées ainsi que la corrélation de pixels adjacents.

- ***Analyse des histogrammes***

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

Les figures 4.5, 4.6 et 4.7 représentent respectivement l'histogramme de lena originale, les histogrammes des images cryptées par les algorithmes étudiés dans les sous sections 4.2.1 et 4.2.2 et enfin les histogrammes de l'AES statique et les différentes variantes proposées dans le chapitre précédent. Nous pouvons apercevoir que la distribution des pixels des images cryptées est uniforme et considérablement différente de celle de l'image originale.

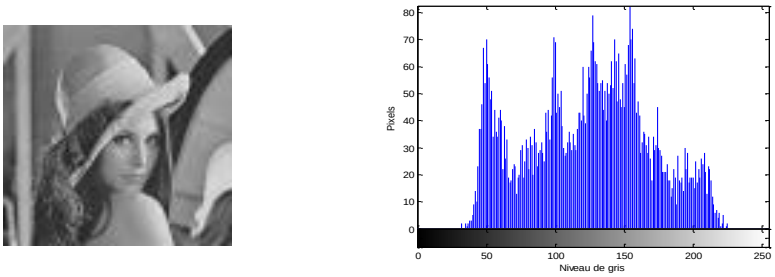


Figure 4. 5. Image originale et son histogramme.

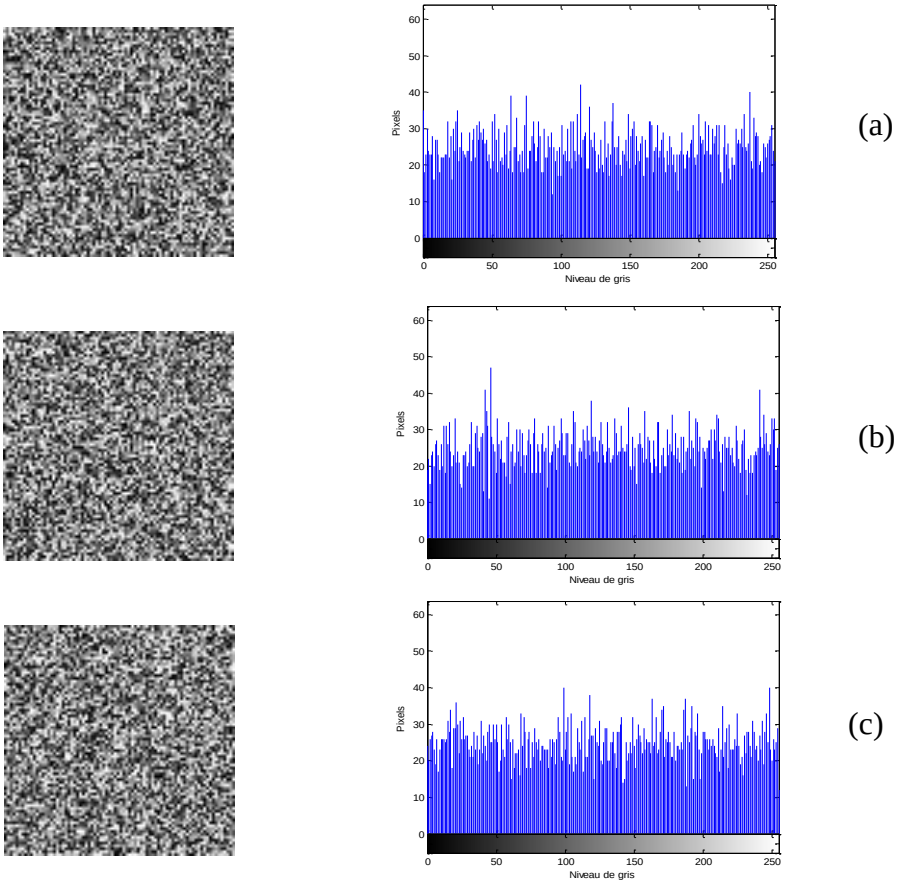


Figure 4. 6. Histogramme de l'image « lena » cryptée par l'algorithme ECKBA (a) algorithme mansour et al. (b), (c) algorithme ACIL.

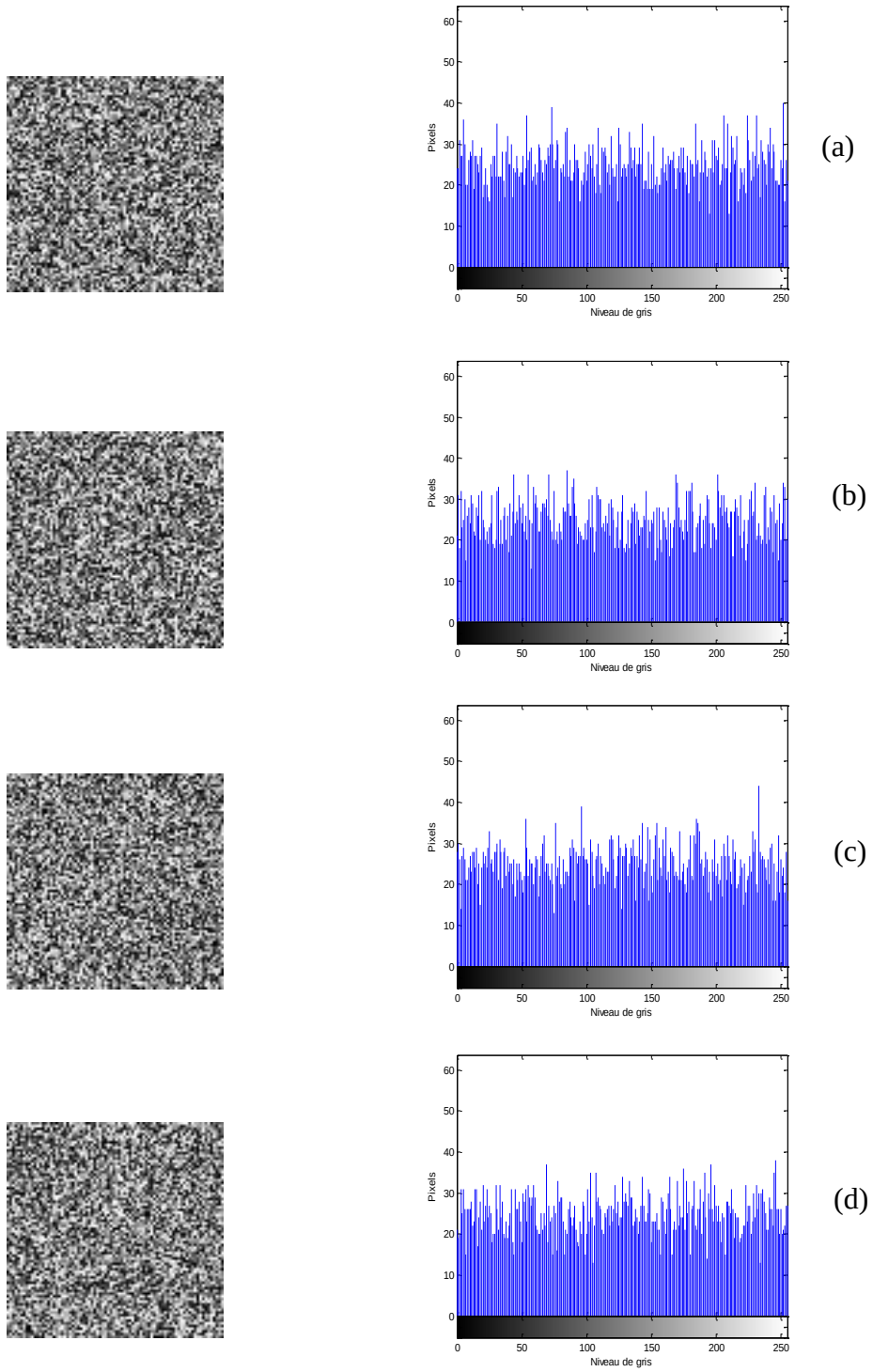


Figure 4. 7. Histogramme de lena cryptée par l'algorithme AES avec : (a) Sbox basée sur Combinaison de deux suites (2ème méthode citée dans le chapitre précédent) ; (b) Sbox de Lorenz ; (c) Sbox de N-Logistic-tent. ; (d) Sbox statique.

- ***Corrélation des pixels adjacents***

Le calcul du coefficient de corrélation entre les pixels permet l'évaluation de la qualité de cryptage des crypto-systèmes. Si deux pixels sont étroitement associés, le coefficient de corrélation sera proche de 1 ou -1. Une valeur proche de 0 indique que les deux pixels ne sont pas liés et on ne peut pas prévoir l'un de l'autre. Cette métrique est calculée à partir de la formule suivante:

$$r_{xy} = \frac{\mathrm{cov}(x,y)}{\sqrt{D(x)}\sqrt{D(y)}} \tag{4.7}$$

Où x et y sont les valeurs du niveau de gris des pixels au même indice des images I et I'.

La covariance et la variance sont données par les équations suivantes :

$$\mathrm{cov}(x,y) = \frac{1}{L}\sum_{i=1}^{L} E(x_i - E(x))(y_i - E(y)) \tag{4.8}$$

$$E(x) = \frac{1}{L}\sum_{i=1}^{L} x_i \tag{4.9}$$

$$D(x) = \frac{1}{L}\sum_{i=1}^{L} (x_i - E(x))^2 \tag{4.10}$$

Avec L est le nombre de pixels utilisés.
Nous avons calculé le coefficient de corrélation de 3000 pixels adjacents de l'image originale et celles cryptées pris horizontalement, verticalement et diagonalement. Le tableau 4.1 regroupe les coefficients de corrélation obtenus pour les algorithmes suivant : ECKBA, Algorithme proposé par Mansour et al, Notre proposition ACIL, AES avec S-Box statique, AES avec S-Box basée sur la combinaison de deux suites, AES avec S-Box basée sur la suite N-Logistic-Tent et AES avec S-Box basée sur la suite de Lorenz discrétisée. Le coefficient de corrélation mesuré pour l'image originale est proche de 1, alors que les coefficients de corrélation pour les images cryptées s'approchent de 0, on en déduit que le chiffrement a atténué considérablement la corrélation entre les pixels des images cryptées.

Les figures 4.8 et 4.9 représentent respectivement les distributions des corrélations des pixels adjacents horizontales de l'image originale et l'image cryptée par ACIL. Ces figures confirment les résultats du tableau 4.1, puisque la distribution des intensités des pixels de l'image originale se concentre sur la diagonale, les pixels sont alors fortement corrélés, tandis que ceux de l'image cryptée sont non-corrélés et ont une distribution uniforme.

Afin de comparer les performances de nos propositions (ACIL et l'AES modifié par des S-Box chaotiques) avec les algorithmes existants dans la littérature, nous avons pris comme référence notre algorithme ACIL et nous avons calculé en valeur absolue la différence entre les coefficients de corrélation des pixels adjacents des images cryptées à l'aide des algorithmes étudiés et les valeurs obtenues pour ACIL (figure 4.10). Pour la direction verticale, ACIL a des performances qui excèdent les autres algorithmes étudiés en ayant un coefficient de corrélation des pixels adjacents beaucoup plus inférieur. Les pixels adjacents diagonalement cryptés par ACIL ont un coefficient de corrélation largement inférieur comparé aux autres algorithmes à l'exception d'ECKBA. Les deux variantes de l'AES avec

des S-Box chaotiques basées sur une combinaison de deux suites et sur la suite N-Logitic-Tent possèdent le plus faible coefficient de corrélation horizontal.

Nous pouvons conclure que l'algorithme ACIL dévoile de bonnes propriétés de confusion et de diffusion en diminuant la corrélation des pixels adjacents et surpasse les performances des algorithmes étudiés dans la littérature en plus des variantes de l'AES avec S-Box chaotique.

Tableau 4. 1. Coefficient de corrélation des pixels adjacents.

Direction des pixels adjacents / Algorithmes proposés	Horizontale	Verticale	Diagonale
Image originale	0,8426	0,9317	0,8012
ECKBA	0,0760	0,0227	-0,0012
Mansour et al.	0,0479	-0,0414	-0,0416
ACIL	-0,0172	-0,0029	-0,0018
AES-Sbox statique	0,0237	-0,0139	-0,0162
AES-Sbox à base de combinaison de deux suites	0,0151	0,0071	-0,0448
AES-Sbox à base de la suite N-Logistic-Tent	-0,0117	-0,0217	0,0724
AES-Sbox à base de la suite de Lorenz	0,0206	0,0277	-0,0423

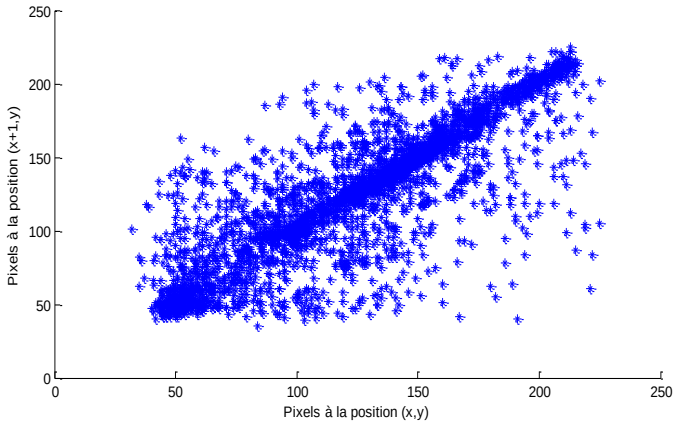


Figure 4. 8. Distribution de corrélation des pairs de pixels adjacents horizontalement dans l'image lena originale (3000 points).

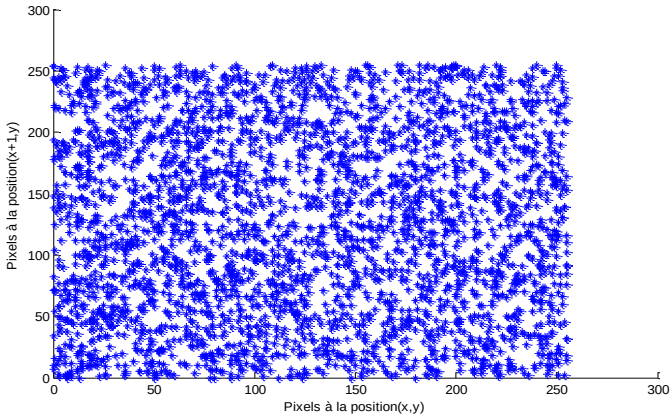


Figure 4. 9. Distribution de corrélation des pairs de pixels adjacents horizontalement dans l'image Lena cryptée par ACIL (3000 points).

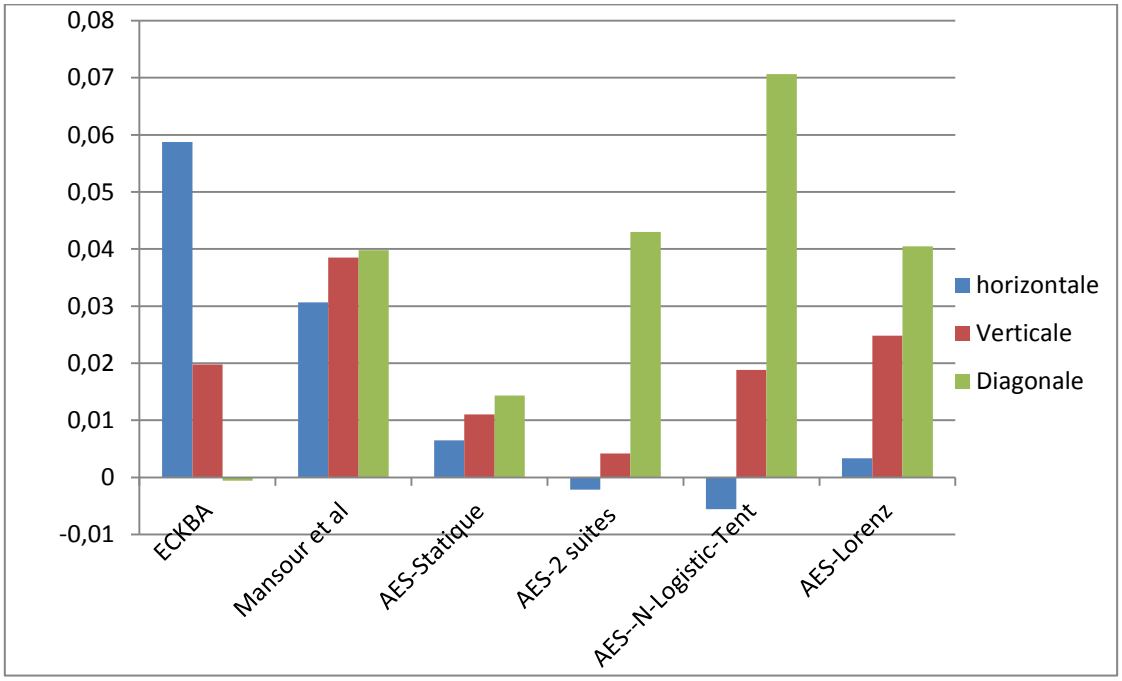


Figure 4. 10. Comparaison entre les coefficients de corrélation des pixels adjacents de l'algorithme ACIL avec les algorithmes étudiés.

4.2.3.2. Analyse différentielle

Une propriété désirable 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. Si un changement mineur dans l'image en clair peut provoquer un changement significatif dans

l'image cryptée, à l'égard de la diffusion et la confusion, alors l'attaque différentielle devient inutile.

Pour calculer l'influence d'un changement d'un seul pixel sur l'image cryptée par n'importe quel algorithme et par conséquent la résistance aux attaques différentielles, deux grandeurs communes peuvent être utilisées ; NPCR (taux de changement du nombre de pixels), et UACI (moyenne unifiée du changement d'intensité) définies par les formules suivantes :

$$NPCR = \frac{\sum_{i=0}^{M-1}\sum_{j=0}^{N-1} D(i,j)}{M \times N} \times 100 \tag{4.11}$$

$$UACI = \frac{1}{M \times N}\sum_{i=0}^{M-1}\sum_{j=0}^{N-1}\frac{\left|I_1(i,j) - I_2(i,j)\right|}{255} \times 100 \tag{4.12}$$

Où M et N représentent respectivement la largeur et la hauteur d'une 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. I_1 et I_2 sont parfois utilisées comme l'image originale et l'image cryptée.

D(i,j) est une matrice de la même taille que I_1 et I_2 tel que :

$$D(i,j) = \begin{cases} 1 & si\ \mathrm{I}_1(i,j) \neq I_2(i,j) \\ 0 & \text{sinon} \end{cases} \tag{4.13}$$

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.
Un score NPCR/UACI élevé se traduit, généralement, par une forte résistance aux attaques différentielles.

Le tableau 4.2 montre que la valeur du NPCR (NPCR = 99,7187 %) entre l'image originale et celle cryptée par notre proposition ACIL est meilleure que les autres algorithmes testés, seulement pour l'AES avec S-Box basée sur la combinaison de deux suites, où les deux NPCR sont très proches. Les valeurs d'UACI des algorithmes testés sont inférieures à celle d'ACIL sauf pour l'AES avec S-Box statique et l'AES avec S-Box basée sur la combinaison de deux suites.

Nous avons aussi calculé les valeurs NPCR et UACI entre deux images cryptées différentes d'un seul pixel à l'image originale de départ (tableau 4.3).

Le score NPCR/UACI de notre algorithme ACIL est meilleur que celui d'ECKBA, il est plus sensible au texte clair. Par contre, les algorithmes de Mansour et al. et l'AES (avec S-Box statique ou chaotique), sont indifférents au changement mineur de l'image originale ; le changement d'un seul pixel n'influe que sur son image cryptée et non pas sur les autres pixels de l'image cryptée.

Nous pouvons conclure que l'algorithme ACIL est plus performant contre les attaques différentielles.

Tableau 4. 2. Valeurs de NPCR et UACI entre les images originales et les images cryptées.

Algorithmes	NPCR (%)	UACI (%)
ECKBA	99,5625	13,4146
Mansour et al.	99,5937	13,0731
ACIL	99,7187	28,1804
AES-Sbox statique	99,5781	28,2150
AES-Sbox à base de combinaison de deux suites	99,75	28,6328
AES-Sbox à base de la suite N-Logistic-Tent	99,5312	28,0985
AES-Sbox à base de la suite de Lorenz	99,57812	28,0224

Tableau 4. 3. Valeurs de NPCR et UACI entre deux images cryptées ayant un pixel différent à l'origine.

Algorithmes	NPCR (%)	UACI (%)
ECKBA	100	15,9849
Mansour et al.	0,0156	0,0059
ACIL	100	50,3355
AES-Sbox statique	0,25	0,0800
AES-Sbox à base de combinaison de deux suites	0,25	0,1107
AES-Sbox à base de la suite N-Logistic-Tent	0,25	0,0917
AES-Sbox à base de la suite de Lorenz	0,25	0,0759

4.2.3.3. Sensibilité de la clé secrète

Tout algorithme de cryptage fiable doit être extrêmement sensible au changement mineur de la clé secrète pour garantir, dans une certaine mesure, la sécurité contre les attaques par force brute.

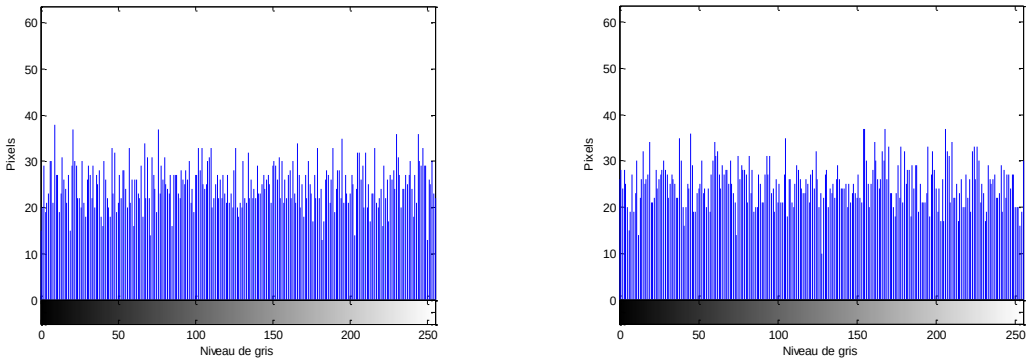
La sensibilité de la clé d'un crypto-système peut être observée par deux méthodes différentes :

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

La figure 4.11 contient les histogrammes de l'image de Lena cryptée à l'aide d'ACIL. Les clés utilisées sont différentes uniquement d'un seul bit au niveau de la condition initiale x_k((a) x_k=3589 ; (b) x_k=3588). Le changement d'un seul bit donne alors deux histogrammes totalement différents avec un très faible coefficient de corrélation entre les deux images (corrcoef = - 0.0008). Dans le tableau 4.4 nous avons regroupé les coefficients de corrélation pour les différents algorithmes étudiés, le tableau contient aussi les paires de clés différentes

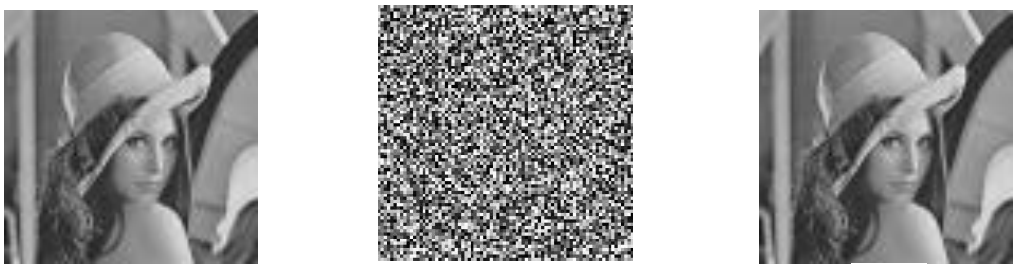
d'un seul bit utilisées pour crypter l'image Lena (80×80). Notre proposition ACIL permet d'avoir le plus faible coefficient de corrélation parmi les algorithmes testés.

Nous avons essayé de plus de décrypter deux images en employant pour l'une la vraie clé secrète tel que $x_k = 3589$ et pour la deuxième une clé différente d'un seul bit ($x_k = 3588$). Nous remarquons que le processus de décryptage a échoué lorsque la clé secrète a légèrement changé et que l'image décryptée est totalement brouillée (figure 4.12). Nous concluons que les images cryptées par ACIL ont une extrême sensibilité à la clé secrète et ne sont pas vulnérable aux attaques par force brute.



(a) (b)

Figure 4. 11. Histogrammes de Lena cryptée par ACIL (a) $x_k = 3589$, (b) $x_k = 3588$.



(a) (b) (c)

Figure 4. 12. (a) Image originale (b) Image Décryptée avec $x_k = 3588$ (c) Image décryptée avec z = 3589.

Tableau 4. 4. Corrélation entre deux images cryptées par deux clés différentes d'un seul bit.

Algorithmes	**Paires de clés**	**Corrélation**
ECKBA	Clé1 = (132CC12DFB03D6403DFD240); Clé2 = (132CC12DFB03D6403DFD241);	-0.0082
Mansour et al.	Clé1 = (x_0=0.95031;y_0=0.567217;p_1=0.372134;p_2=0.292134) Clé2 = (x_0=0.95031005; y_0=0.567217; p_1=0.372134 ; p_2=0.292134)	0.0352
ACIL	Clé1 = (x_0=3589; y_0=47542; z_0=32294;Δ=1 ;$LFSR_0$=44257) Clé2 = (x_0=3588; y_0=47542; z_0=32294;Δ=1 ;$LFSR_0$=44257)	-0.0008
AES-Sbox statique	Clé1 = (000102030405060708090a0b0c0d0e0f); Clé2 = (000102030405060708090a0b0c0d0e0e);	0.0049

4.2.3.4. Analyse de l'espace de la clé

L'espace de la clé d'un algorithme de cryptage/décryptage est le total des clés différentes qui peuvent être utilisées dans la procédure de cryptage/décryptage. Il doit être assez large (supérieur à 128 bits) pour s'assurer qu'une attaque par force brute est non faisable [85].

La clé de notre algorithme ACIL est composée de 3 parties :

- Les conditions initiales (x_k, y_k, z_k) de taille 4 octets chacune.
- le pas d'intégration initial codé sur 4 octets.
- La condition initiale (seed) du LFSR appelée $LFSR_0$ codée sur 2 octets.

Ce qui fait une totale de 144 bits, et le nombre d'opérations possibles pour récupérer la clé est $2^{144}=2.2301\times10^{43}$. On remarque que l'espace de la clé est suffisante pour résister aux attaques par force brute.

La taille de la clé initiale de ACIL est supérieure à celles de l'algorithme AES, l'algorithme ECKBA et l'algorithme de Mansour et al opérants avec 128 bits, 128 bits et 96 bits respectivement.

Au début nous avons pensé à étendre la clé en ajoutant les paramètres de la suite de Lorenz (ρ, σ et β). Comme nous l'avons cité au chapitre II pour le système de Lorenz continu, le chaos apparait si : $\sigma > \beta+1$; $\rho > 0$ et $\rho > \frac{\sigma(\sigma+\beta+3)}{\sigma-\beta-1}$. L'intervalle de recherche peut être limité en calculant les exposants de Lyapunov pour localiser le chaos, or le système qu'on a utilisé est discret là où on a localisé des cycles courts et on a recourt à l'application de perturbation (LFSR). L'étude de ces paramètres sera parmi les perspectives des travaux de thèse.

4.2.3.5. Mesure du temps d'exécution

Les résultats du tableau 4.5 ont été évalués par MATLAB 7.10.0 (R2010a). Les expériences ont été menées à l'aide d'un processeur Intel® Core i5 avec une fréquence de 2.5 GHz. Dans ce tableau, le temps d'exécution nécessaire pour chaque algorithme pour crypter une image

Lena (80×80) est fourni. L'algorithme ACIL est plus rapide avec un temps ne dépassant pas 2.028 secondes. En effet, le temps d'exécution de l'algorithme ECKBA est 269.242 secondes, il est alors 132.76 fois plus lent que ACIL. L'algorithme AES codé sur Matlab s'exécute en 11.94 secondes et celui proposé par Mansour et al. en 4.1496 secondes ; ce qui est un temps d'exécution proche de celui d'ACIL. La rapidité discernée de la simulation sous Matlab de notre proposition ACIL nous a encouragés à le tester sur un simulateur de réseaux de capteurs et à l'implémenter sur des capteurs d'une plateforme réelle.

Tableau 4. 5. Mesure du temps d'exécution en seconde des différents algorithmes testés.

Algorithmes	**Temps d'exécution(s)**
ECKBA	269.242
Mansour et al.	4.1496
ACIL	2.028
AES	11.94

4.2.3.6. Calcul de la distance de Hamming

La distance de hamming est le nombre de bits différents entre deux séquences binaires. Ce critère mesure l'effet d'avalanche et il se définit par l'équation suivante :

$$d_H(C_1, C_2) = \sum_{i=0}^{n-1} \left| C_1(i) - C_2(i) \right| \tag{4.14}$$

Nous avons crypté deux séquences binaires E_1 et E_2, qui diffèrent d'un seul bit, par notre algorithme ACIL. Nous avons calculé la distance de Hamming entre les deux séquences cryptées résultantes C_1 et C_2, en modifiant à chaque fois la position du bit à changer pour les séquences de texte en clair. La figure 4.13 représente le pourcentage de la distance de Hamming en bits donné par l'équation 4.15 :

$$pd_H(j) = \frac{d_H(j)}{L_b} \times 100 \tag{4.15}$$

La taille des séquences binaires testées est L_b=128 bits et j représente la position du bit différent. Nous obtenons 76.92% de paires de séquences cryptées avec une distance de Hamming supérieure à 48%, et 23.08 paires ayant une distance inférieure à 48%, qui reste un pourcentage non négligeable. Toutefois la valeur moyenne de la distance de Hamming pour ACIL est 48.77%. L'algorithme ACIL vérifie alors le critère d'avalanche, mais une amélioration de la propriété de confusion en ajoutant une fonction de substitution par S-Box chaotique est envisageable. Le changement de chaque bit à n'importe quelle position pourrait changer uniformément.

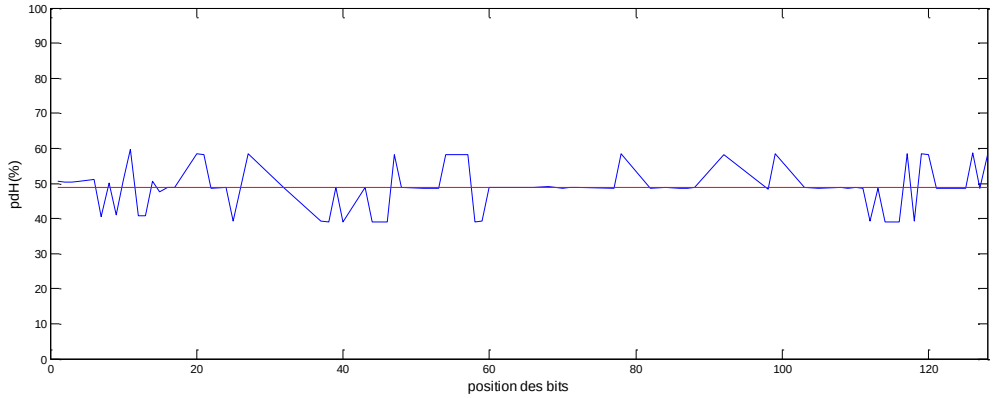


Figure 4. 13. Pourcentage en bits de la distance de Hamming en fonction des positions des bits changés pour des paires des textes cryptés par ACIL.

4.3. Implémentation sur un réseau de capteurs

L'objectif est d'implémenter les dynamiques chaotiques dans un réseau de capteurs sans fil afin de sécuriser les informations envoyées d'un nœud à un autre. Le défi à emporter est de concevoir une approche qui réponde aux exigences d'un tel réseau. Un réseau de capteurs sans fil est constitué de dispositifs à faible puissance de calcul, une faible mémoire de stockage et surtout de faibles ressources en énergie [86], [87], [88]. La consommation d'énergie est très critique puisque les nœuds déposés dans la nature comprennent des batteries à durée de vie limitée et leur changement ne s'avère pas évident à court terme ; d'où la nécessité de focaliser sur cet aspect pour prolonger la durée de vie de la batterie.

Deux phases s'avèrent essentielles : la simulation et la pratique.

Certes la simulation est une étape importante qui aide à déboguer les codes, à estimer leurs performances et les optimiser, mais reste limitée et souffre de plusieurs inconvénients telles que les hypothèses et les approximations qui nuisent parfois à la précision. Nous avons besoin alors de valider nos algorithmes en utilisant une plateforme réelle. Nous allons tester nos contributions sur la construction des S-Box par chaos, et notre algorithme proposé pour le cryptage d'image ACIL, les implémenter sur un réseau de capteurs et comparer leur performances avec l'AES standard..

4.3.1. Outils de simulation

Plusieurs outils de simulation ont été développés pour les réseaux de capteurs comme les simulateurs Tossim, Atemu, Avrora et Wsim/Wsnet. Ces simulateurs diffèrent par leur langage de développement (NesC pour Tossim et Java pour Avrora) et le type de microcontrôleur supporté, exemple Atemu et Avrora sont des émulateurs des processeurs AVR utilisés dans les capteurs MICA2. Pour ces raisons, nous avons choisi d'utiliser Wsim/Wsnet puisqu'ils supportent les microcontrôleurs MSP430 intégrés sur les nœuds WSN430 de la plateforme réelle SensLab, et les codes sont écrits en langage C.

4.3.1.1. Wsim et Wsnet

Wsim [89] et Wsnet [90] sont deux outils complémentaires de simulation intégrée.

Wsim est un émulateur de système pour les plateformes basées sur des microcontrôleurs. Il s'appuie sur le cycle exact de la simulation de plateforme. Il peut réaliser une simulation complète d'événement matériel (hardware) et redonner au développeur une analyse de timing précise du logiciel (software) simulé. Une estimation précise du timing, de la consommation mémoire et de la puissance peuvent être obtenues durant la simulation.

De plus, Wsim peut être utilisé en mode autonome, pour déboguer les codes lorsqu'il n'y a pas un dispositif radio utilisé.

Wsnet est un simulateur événementiel de réseaux de capteurs qui comprend des modèles réalistes de nœuds capteurs utilisés dans Wsim.

Nous avons choisi d'exécuter nos codes sur une plateforme MSP430 de WSIM avec un microcontrôleur MSP430f1611 16 bits et un chipset Radio de type Ti CC1100.

La figure 4.14 représente la Wconsole de Wsim affichant un texte crypté et décrypté par l'algorithme AES modifié.

Figure 4. 14. Wconsole de Wsim montrant le texte crypté et décrypté.

4.3.1.2. Esimu

Esimu est un modèle complet d'énergie de système basé sur des mesures non-intrusives.
Ce modèle vise à être intégré dans des outils de simulation à cycle précis et rapide pour donner la rétroaction de consommation d'énergie à la programmation logicielle des systèmes embarqués tout en étant indépendant des outils de compilation ou des composants logiciels tels que les protocoles de réseau ou les systèmes d'exploitation [91]. Les évaluations tiennent compte de la consommation du système entier y compris les périphériques. Esimu est donc un simulateur de consommation d'énergie pour des plateformes de système embarqué tels que les nœuds de réseau de capteurs. Les expériences des développeurs sur une plateforme ARM9

complexe prouvent que les estimations du modèle ont une erreur inférieure de 10 % par rapport à la consommation d'un système réel, alors que pour une plateforme WSN430 le taux d'erreur est inférieur à 5 %, qui sont assez précises et acceptables pour la conception d'un code source [92]. Dans notre cas, nous avons utilisé Esimu pour estimer l'énergie consommée à partir des traces d'exécution de Wsim comme le montre la figure 4.15.

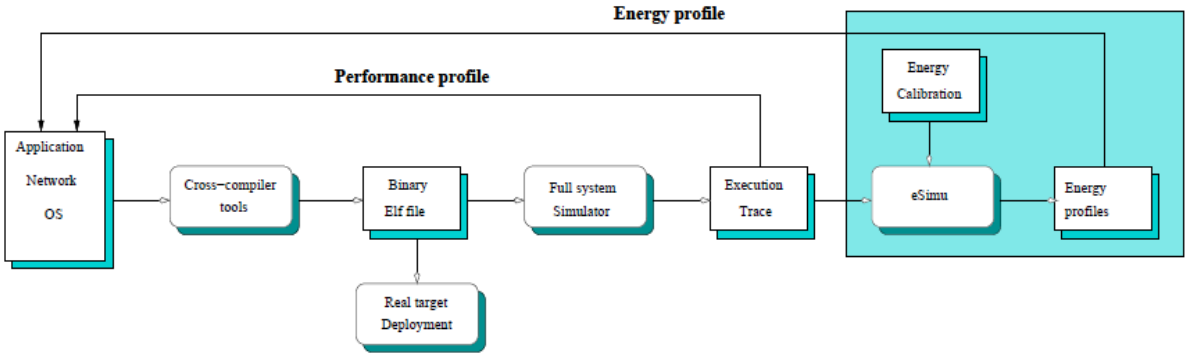


Figure 4. 15. Intégration du simulateur Esimu dans l'outil de simulation [89].

La lecture et la visualization du profil d'énergie générée par Esimu sera faite à l'aide de KCachegrind [93] (figure 4.16). KCachegrind est un outil de visualisation de profil.

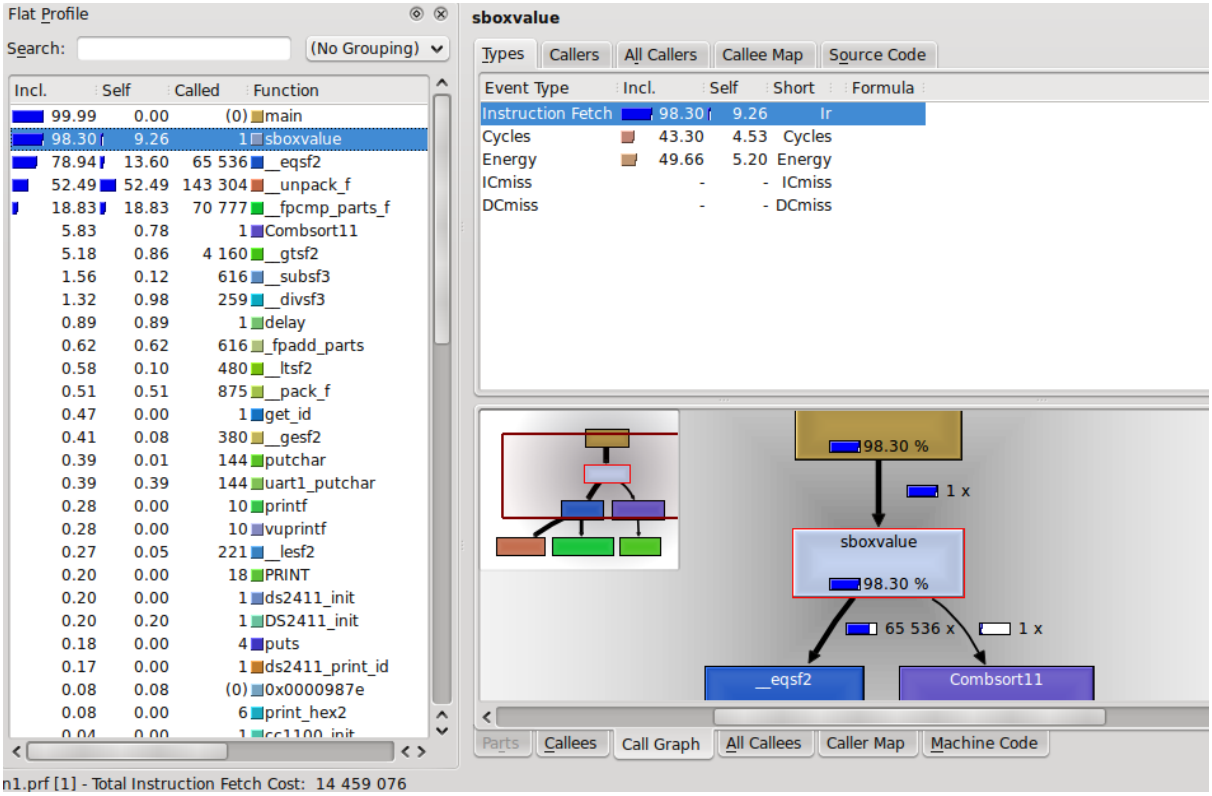


Figure 4. 16. KCachegrind: visualise un rapport de l'estimation du temps et d'énergie donnés par Esimu.

4.3.2. Implémentation de la méthode d'allocation d'intervalle

Afin de simplifier l'assignation des valeurs entières à chaque output des suites chaotiques, nous avons choisi l'affectation par dichotomie. Le mécanisme de recherche par dichotomie consiste à découper à chaque fois l'espace de recherche en deux, raffiner la recherche en découpant par des multiples de deux et vérifier l'existence de la valeur réelle (output) dans l'intervalle en question, chaque intervalle lui correspond un label ou étiquette qui n'est que la valeur entière du rang de l'intervalle.

Le nombre maximal M de tests correspond à la valeur du premier exposant entier de 2 supérieur ou égal au nombre N d'intervalles possibles, on peut le définir par cette équation :

$$M=\left(\log_2(N)\right) \tag{4.16}$$

Dans notre cas, étant donné le nombre maximal d'intervalles est 256, on aura 8 tests pour chaque valeur réelle ; ce qui donne 2048 tests au max pour 256 valeurs souhaitées dans le cas où on n'aura pas de répétitions. La figure 4.17 montre le nombre d'itérations nécessaires pour construire des S-Box en partant d'une seule condition initiale. La valeur moyenne des itérations est 2376 ; ce qui signifie que plusieurs valeurs appartiendront à un même intervalle.

RQ : La méthode basée sur la binairisation des sorties réelles des suites chaotiques proposée

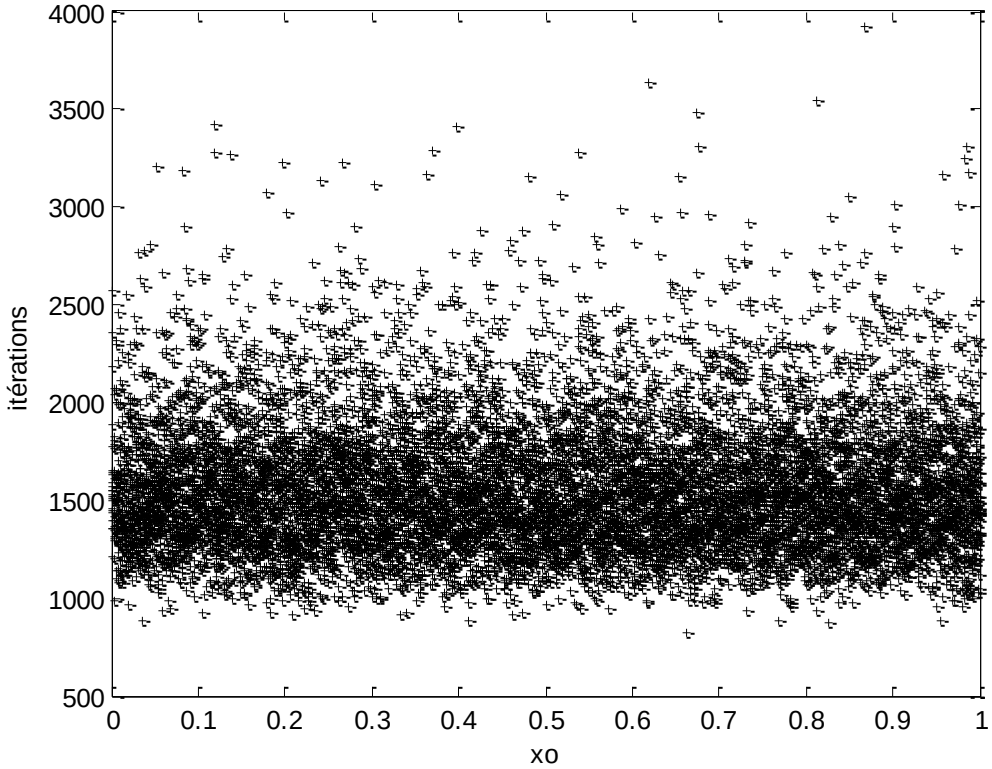


Figure 4. 17. Nombre d'itérations nécessaires pour construire des S-Box en fonction des conditions initiales avec un pas de 0.0001.

dans l'article [55] est implémentée aussi ; mais étant donné la complexité de son code, seule la partie de binairisation va être testée.

4.3.3. Implémentation de notre proposition avec la méthode du tri

Nous avons choisi de nous restreindre à une seule dimension des suites chaotiques pour ne pas augmenter la complexité des calculs et donc la consommation. Nous avons fixé le paramètre de contrôle, et les conditions initiales formant une partie de la clé du système modifié. La fonction de tri constitue la fonction la plus consommatrice de notre proposition. Pour cela, nous avons choisi un algorithme simple, rapide et non récursif qui est l'algorithme de tri comb sort (voir annexe).

Pour 10^6 conditions initiales testées, les sorties ne montrent pas des redondances. On aura pour chaque entrée une S-Box après un nombre d'itérations fixe 256.

4.3.4. Adaptation et implémentation du chaos modulaire

Les systèmes chaotiques complexes impliquent l'utilisation de la virgule flottante arithmétique, qui affecte la vitesse de l'algorithme ainsi que la réalisabilité matérielle et logicielle. L'arithmétique à virgule fixe et les systèmes chaotiques simples sont recommandés pour accélérer la vitesse du cryptage et pour veiller à la réalisation simple matérielle et logicielle. Cependant, les nombres à virgule fixe risquent d'être tronqués ou arrondis ; d'où une perte d'information.

Puisque les systèmes chaotiques sont des systèmes déterministes, il existe des outils de la théorie du chaos permettant de discerner le chaos. Une fois qu'un intrus trouve quelques informations sur les orbites des systèmes chaotiques, il peut utiliser ces informations pour réduire la complexité de trouver la clé sécurisée. Avec l'utilisation de plusieurs systèmes chaotiques, la cryptanalyse du chaos est plus difficile puisque la sortie est déterminée par de nombreuses orbites chaotiques mixtes.

L'exécution des algorithmes chaotiques sur des ordinateurs ou sur des microcontrôleurs se confronte à une limitation gênante qui est la précision finie des nombres représentés. Cette précision finie engendre un problème majeur qui est l'apparition des cycles courts ainsi que les convergences vers un point fixe.

Les problèmes du cycle court sont l'un des plus graves problèmes qui ont empêché la progression de la cryptographie chaotique de la théorie à la pratique [94].

Tao dans [94] suggère l'injection d'une perturbation au système pour renforcer ses propriétés cryptographiques. La modification affectera la valeur de l'orbite ou les paramètres ou bien les deux à la fois. Ceci mène à un cycle de longueur plus étendue.

La perturbation peut être réalisée à l'aide d'un LFSR ou bien Linear Congruential genrators [94], [95].

La perturbation est choisie selon les principes suivants :

(i) Elle doit avoir un cycle long de longueur contrôlable et une distribution uniforme ;

(ii) Elle ne doit pas dégrader les propriétés statistiques des dynamiques chaotiques, l'amplitude du signal perturbateur doit être beaucoup moins inférieure à celle du signal chaotique. On définit le rapport signal sur bruit : SNR :

$$SNR = 10\log_{10}\left(\frac{amplitude \quad \max imale \quad du \quad signal \quad chaotiquel}{amplitude \quad \max imale \quad du \quad signal \quad perturbateur}\right) \tag{4.17}$$

Le SNR doit être supérieur à 40 dB.

(iii) Le temps de perturbation doit être faible par rapport au temps total d'exécution du système.

Le générateur probable de perturbation qu'on peut utiliser est "maximal length LFSR : registre à décalage à rétroaction linéaire de longueur maximale" puisqu'il génère des séquences (m-séquence) ayant les avantages suivants :

(i) Une longueur définie du cycles (2^L-1) (L est le degré).

(ii) Une distribution uniforme.

(iii) Une implémentation facile

(iv) Une amplitude maximale du signal contrôlable donnée par : $2^p(2^L-1)$ quand il est utilisé dans un système à P-Précision (nombre de bits).

La perturbation se déclenche à t = 0, avec une période Δ. Elle est appliquée par un XOR avec le signal chaotique en question avec t = kΔ, k = 0, 1, 2, ...

$$x_{t+1,i} = \begin{cases} [F(x_t)]_i & 1 \le i \le P-L \\ [F(x_t)]_i \oplus a_{k+P+1-i} & P-L+1 \le i \le P \end{cases} \tag{4.18}$$

$[F(x_t)]_i$ représente le $i^{ème}$ bit de $F(x_t)$. Quand $t \neq k\Delta$, dans chaque intervalle Δ, aucune perturbations n'arrive, alors $x_{t+1} = F(x_t)$.

Une autre alternative est l'utilisation du LCG (générateur congruentiel linéaire) [95].

Dans l'article de Silva [51], les auteurs utilisent un LFSR de période maximale 128. Un m-LFSR assure un cycle de (2^{128}-1) valeurs différentes. Le polynôme primitif est : $x^{128}+x^7+x^2+x+1$.

Pour construire la S-box, nous avons pris l'équation discrétisée de Lorenz [51]. Elle est donnée par l'équation (3.10).
On choisit les valeurs x_0, y_0, z_0 et le pas d'intégration aléatoirement et on itère la suite n fois tel que $n \geq 10$.
L'application d'une perturbation s'avère nécessaire comme l'a indiqué Silva dans son article [51] étant donné l'apparition de redondance à partir d'un certain nombre d'itérations. On ajoute donc une perturbation qui consiste à appliquer la sortie d'un LFSR à chacune des deux composantes (y_{k+1}, z_{k+1}) à l'aide d'un XOR, alors que x_{k+1} est XORé avec le pas d'intégration Δt. Le système de perturbations est le suivant [51]:

$$\begin{aligned} y_{k+1} &= y_{k+1} \oplus LFSR \\ z_{k+1} &= z_{k+1} \oplus LFSR \\ \Delta t &= \Delta t \oplus x_{k+1} \end{aligned} \tag{4.19}$$

Le pas d'intégration Δt n'est pas fixe, il change pour chaque itération.

Contrairement à l'article [51], nous n'avons choisi la taille des variables ainsi que le pas d'intégration égaux à 2 octets (16 bits). Ce pas d'intégration prend la valeur du même LFSR

et non pas d'un autre LFSR auxiliaire. Si les 4 bits les plus hauts du LFSR sont tous nuls, le pas d'intégration prend la valeur de l'octet le plus haut ; sinon l'octet le plus bas lui sera affecté.
L'octet le plus haut et l'octet le plus bas de chaque valeur de l'ensemble (x_{k+1}, y_{k+1}, z_{k+1}) sont stockés dans un tableau.

Les étapes précédentes sont répétées jusqu'à avoir 256 valeurs différentes sans répétition.

Pour avoir une idée sur le nombre d'itérations nécessaires pour construire une S-Box pour les différentes conditions initiales, nous avons essayé de faire une recherche exhaustive ; ce qui nous parait difficile étant donné le nombre de possibilités de combinaisons de conditions initiales (2^{48} combinaisons pour la Sbox basée sur la suite de Lorenz et 2^{32} combinaisons pour la S-Box sur la suite N-Logistic-Tent). Nous divisons alors les intervalles, nous les parcourons avec un pas variable et nous récupérons par la suite le nombre minimal d'itérations nécessaire et ses occurrences. La figure 4.18 montre les occurrences du nombre minimal d'itérations. Le nombre d'itérations le plus probable est 151 itérations et la valeur minimale est 124.

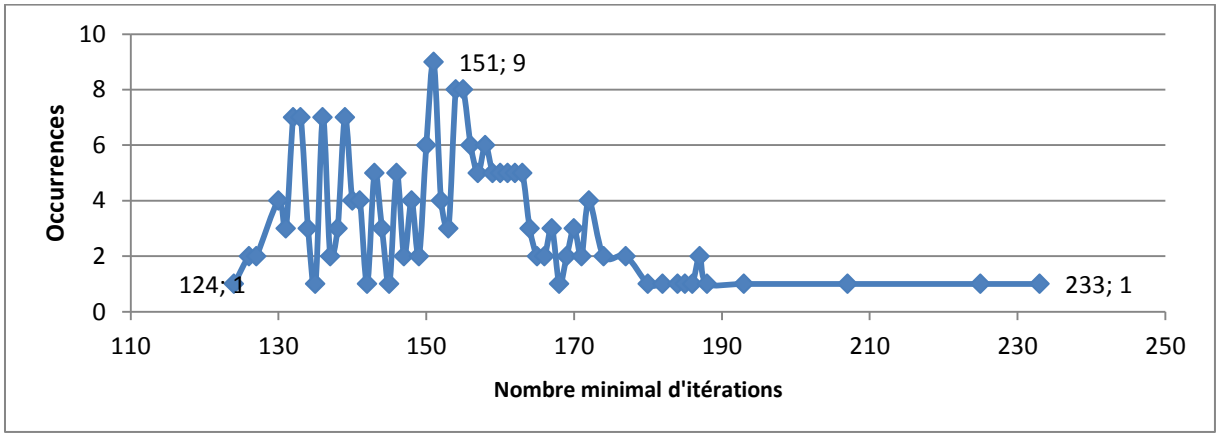


Figure 4. 18. Occurrences du nombre minimal d'itérations pour S-box basée sur la suite de Lorenz discrétisée.

La figure 4.19 représente le nombre d'occurrences pour le nombre minimal d'itérations pour la S-Box basée sur la suite N-Logistic-Tent. Le nombre d'itérations qui a le plus d'occurrences est 320 avec 212 occurrences.

Remarque : Dans cette partie de S-Box à base de suite chaotique discrétisée, nous avons implémenté aussi l'approche proposée dans l'article [61] en choisissant la valeur minimale de K = 35.
La consommation en énergie de toutes ces approches va être étudiée et comparée dans la section suivante.

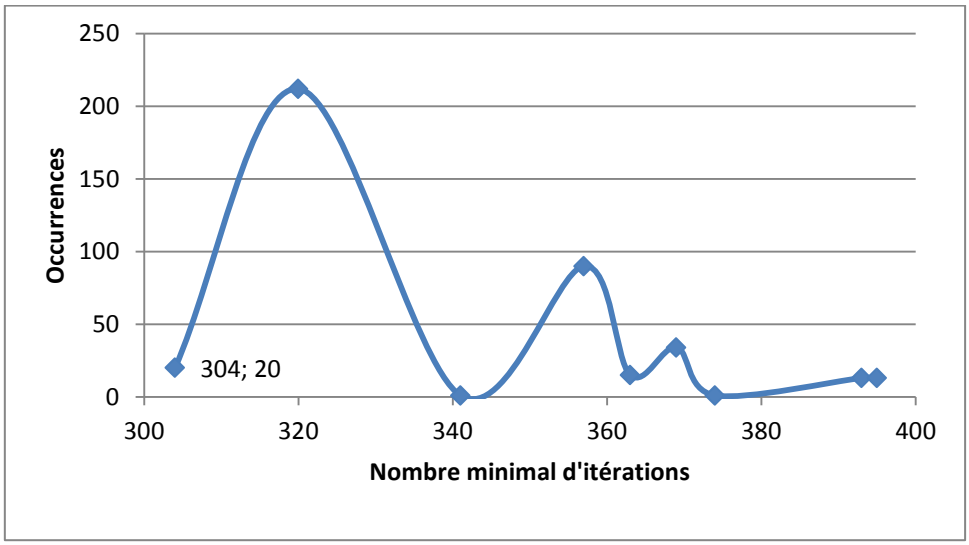


Figure 4. 19. Occurrences du nombre minimal d'itérations pour S-box basée sur la suite N-Logistic-tent.

4.3.5. Etude expérimentale sous la plateforme SensLab

Les tests des algorithmes de cryptage ont été réalisés sur la plate-forme SensLab. SensLab est une plate-forme ouverte à large échelle qui dispose de 1024 nœuds déployés sur quatre sites à savoir Rennes, Lilles, Strasbourg et Grenoble là où on avait accès aux nœuds (Figure 4.20).

- **Description technique de la plate-forme**

Contrairement aux nœuds commercialisés dont une description détaillée est donné au premier chapitre, les nœuds SensLab dispose de trois modules: une passerelle et deux nœuds wsn430 alimentés par une batterie et une alimentation DC (figure 4.21).

- Nœud ouvert: L'utilisateur a accès à ce nœud pour le programmer. Les applications sont exécutées sur ce nœud.
- Nœud de contrôle: L'utilisateur n'a pas accès à ce nœud. Toutefois, il permet d'enregistrer des informations utiles comme la qualité du signal, la consommation d'énergie et de fixer par la suite le type d'alimentation (secteur ou batterie).
- Passerelle : Elle permet de reprogrammer, de commander, d'alimenter et de relier le nœud ouvert et le nœud de contrôle avec le serveur. La liaison est établie grâce à son adresse IP.

Des éléments supplémentaires peuvent être ajoutés comme le GPS ou un accéléromètre.

Chaque nœud wsn430 est équipé de :

- Un microcontrôleur : Ti MSP430-f1611 à 16 bits et une fréquence 8MHz. Il est doté de 10ko de mémoire vive (RAM) et 48 ko de mémoire morte (ROM). Les périphériques sont: deux modules USART avec des interfaces SPI, I2C et UART; watchdog ; DMA; convertisseur ADC/ DAC (résolution 12 bits). Il peut être programmé par JTAG ou BSL.
- Trois capteurs physiques: acoustique (microphone omnidirectionnel KECG1540), de luminosité (Taos TSL2550) et de température (Maxim DS1722).

- Une interface radio Ti CC2420 (2,4GHz) ou CC1101 (868MHz).
- Une mémoire flash externe ST M25P80 (1Mo).
- Une batterie Varta Polyflex [96].

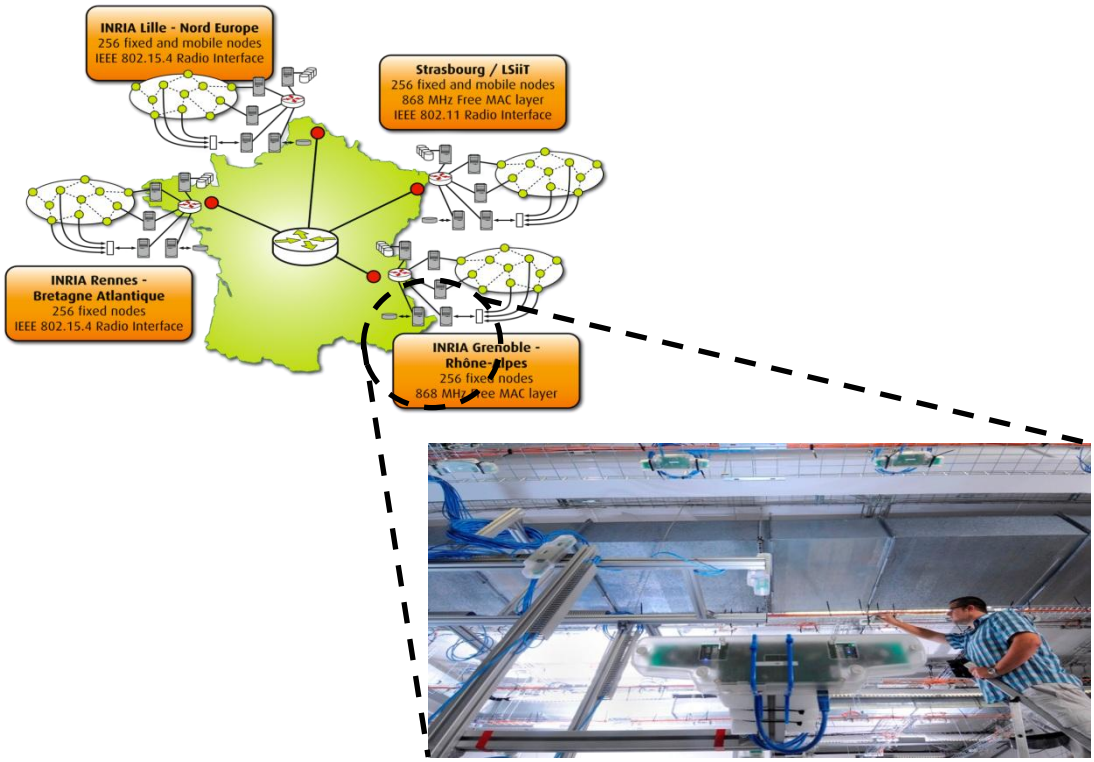


Figure 4. 20. Plateforme SensLab, site de Grenoble [96].

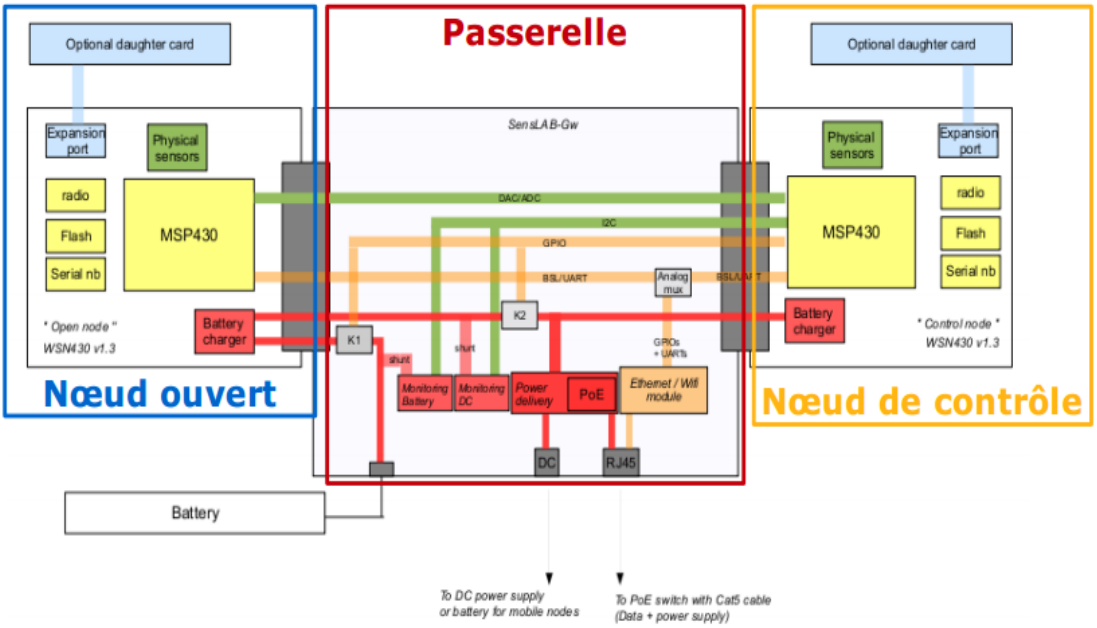


Figure 4. 21. Architecture d'un nœud SensLab [96].

- **Mode de fonctionnement**

Pour créer une expérience, il faut tout d'abord s'inscrire sur le portail SensLab. Chaque utilisateur aura accès à une machine virtuelle SensLab liée à sa machine en utilisant une connexion ssh. Le schéma de la figure 4.22 illustre le fonctionnement et l'interconnexion entre la machine virtuelle, les nœuds et l'utilisateur.

La première étape consiste à lancer une nouvelle expérience comme le montre la figure 4.23 où on peut voir à travers la page d'accueil le nombre d'expérience passées, celles en cours, l'état de la machine virtuelle et le quota occupé (limité à 2 Gb).

Cette page contient aussi des liens vers des outils de surveillance de la plate-forme permettant d'observer le statut individuel de chaque nœud, et la charge de la plate-forme en termes de tâches planifiées.

Le choix des nœuds peut se faire selon leur localisation ou leur disponibilité. L'utilisateur a accès à une carte 2D ou 3D (figure 4.24) pour choisir les nœuds et interagir avec eux en choisissant le temps de lancement des expériences, leur durée, les mesures à récupérer (luminosité, température, consommation etc..) et affecter à chaque nœud le micro-logiciel (firmware) qui lui correspond. On peut associer des micro-logiciels et des profils différents pour chaque nœud.

Le type de la radio est fixé avant le démarrage de l'expérience et peut changer avec un nouveau lancement.

L'utilisateur de SensLab peut accorder un nom à son expérience qui doit être supérieur à quatre caractères. La durée des expériences est fixée en minutes. Les nœuds à choisir sont groupés par 8 formant un cluster. Ils sont alors proches géographiquement.

Il faut noter que le micro-logiciel associé au nœud doit être un fichier MSP430 Intel Hex, qu'on le génère lors de la simulation des codes sur Wsim.

La configuration des nœuds est accessible avant le démarrage de l'expérience où l'utilisateur peut choisir le type d'alimentation de ces nœuds (batteries ou DC). La plateforme SensLab permet aussi de mesurer le courant, la tension et la puissance consommée instantanément lors des expériences lancées.

L'utilisateur a le choix d'accéder instantanément aux données et les récupérer en temps réels ou bien les consulter après la fin de l'expérience à travers sa machine virtuelle.
Au lancement d'une expérience, la seule action disponible est de l'arrêter.

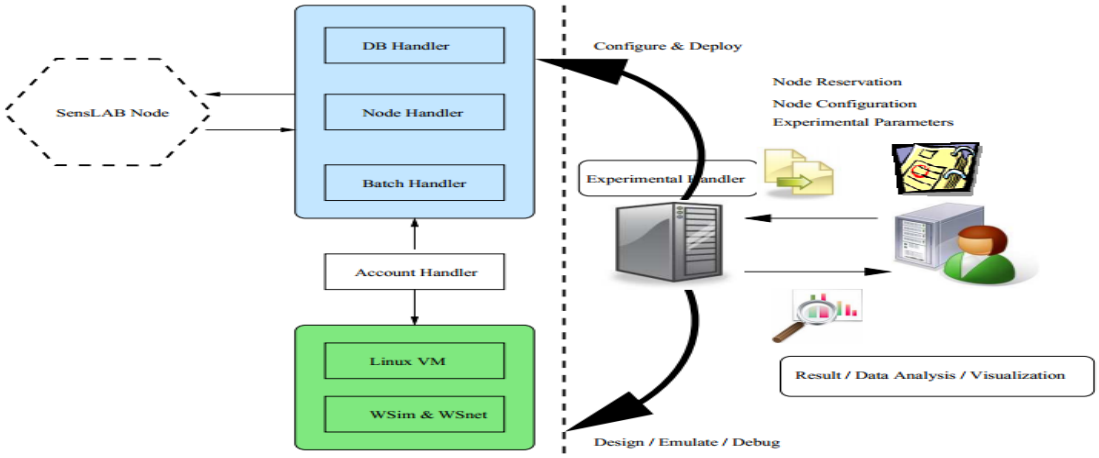


Figure 4. 22. Interconnexion entre utilisateur et nœud.

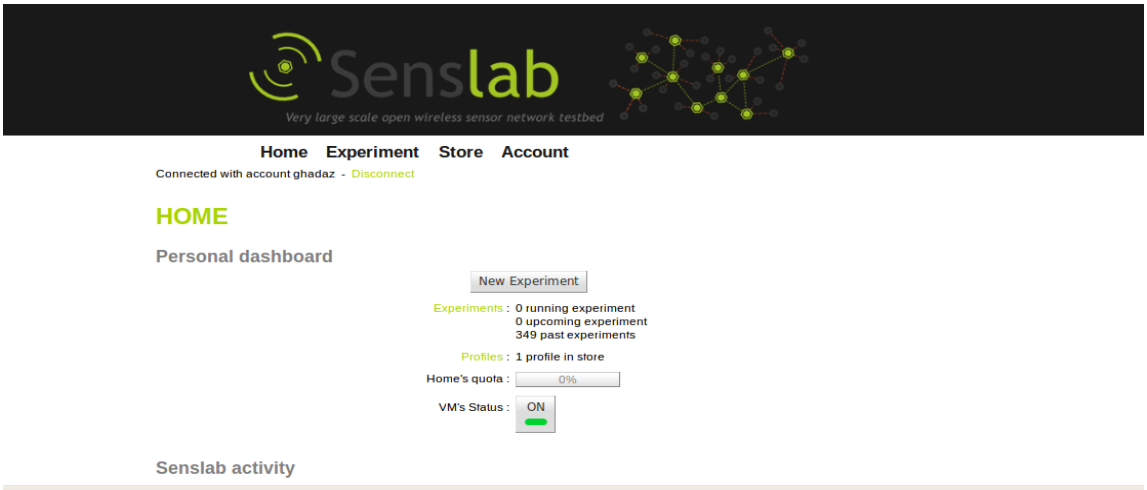


Figure 4. 23. Page d'accueil de la plateforme SensLab.

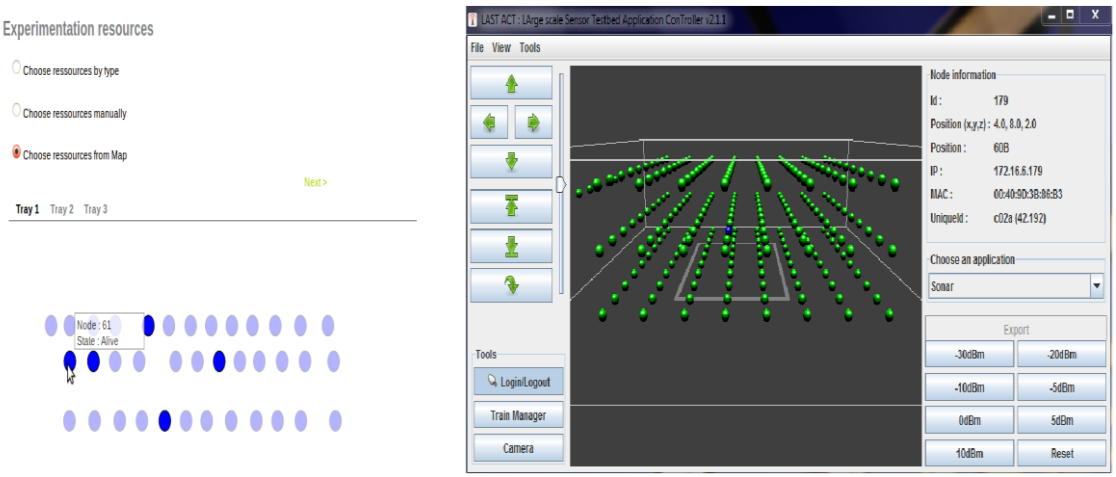


Figure 4. 24. Interface 2D et 3D de la plateforme SensLab.

En tenant compte de tout ce qui précède, nous avons en premier lieu testé nos codes sur plusieurs nœuds en utilisant l'interface radio Ti CC1101 comme le montre la figure 4.25. Un nœud crypte l'information et l'envoie au reste des nœuds à l'écoute, chaque nœud recevant l'information cryptée la décrypte en utilisant la clé qu'il possède déjà.

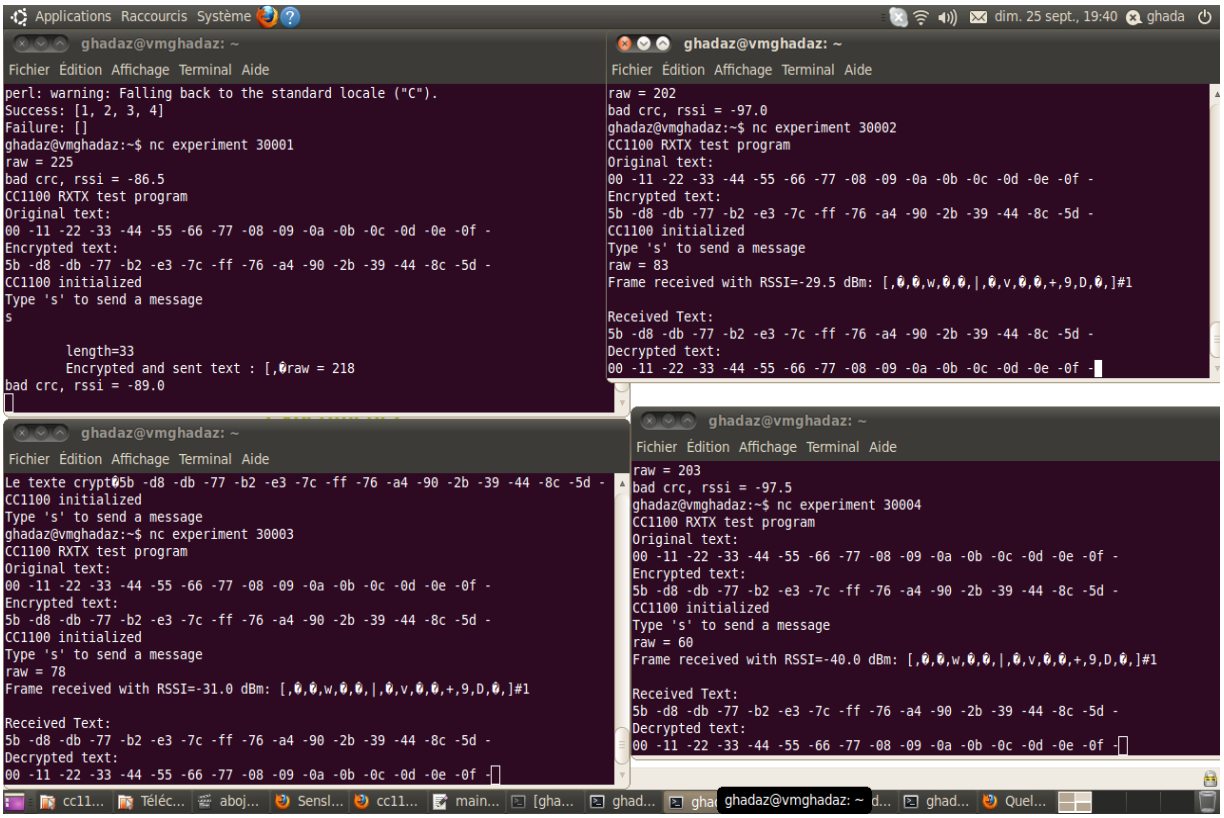


Figure 4. 25. Communication entre 4 nœuds de SensLab et cryptage et décryptage de 16 octets.

4.3.6. Etude de la consommation des algorithmes proposés

Dans cette section, nous allons présenter les résultats de simulations détaillés de la consommation d'énergie réalisées à l'aide de Esimu ainsi que les résultats expérimentaux des codes implémentés sur une plateforme réelle de réseaux de capteurs appelée SensLab. Parmi les méthodes de constructions des S-Box que nous avons citées dans les sections précédentes, nous en avons sélectionné quelques-unes en nous basant sur la ressemblance entre certaines ([52], [53] et [54]) ([59] et [60]) et la complexité d'autres ([55] et [56]). Nous n'avons pas testé la méthode à base de système chaotique spatiotemporel, décrite dans la sous-section 3.4.1.3, parce qu'elle utilise la suite logistique comme suite locale. Le nœud capteur crypte les données par exemple par un algorithme AES modifié, là où on injecte à chaque fois une S-Box générée dynamiquement par chaos selon les différentes propositions citées, et les envoie sur le canal. Plusieurs nœuds reçoivent l'information et la décrypte puisqu'ils possèdent déjà la clé avec les conditions initiales pour la construction des S-Box chaotiques. L'énergie consommée est mesurée dans les deux cas.

4.3.6.1. Etude de la consommation via Esimu

a- Energie consommée par les algorithmes de construction des S-Box chaotiques

Le tableau 4.6 et la figure 4.26 regroupent les consommations en énergie de cinq approches. Les deux premières approches font partie des méthodes basées sur des suites chaotiques à sorties réelles. Diviser l'espace de phase par dichotomie (2ème approche) semble ne pas être la méthode la plus convenable pour assigner des valeurs entières aux trajectoires des suites chaotiques. Le choix de trier les sorties (1ère approche) consomme 63,73 10^{-4} J qui est largement inférieure à la méthode de dichotomie, et cette valeur reste constante pour toutes les conditions initiales choisies aléatoirement puisqu'il n y a pas de répétitions (10^6 points testés). Nous avons essayé de tester une autre méthode dans cette même partie mentionnée dans l'article [55], elle nous parait assez complexe et la partie de binairisation seulement consommait 178,83 10^{-4}J.

Les 3 dernières approches sont basées sur des suites à sorties entières. La méthode de Silva (4ème approche) que nous avons adaptée pour les S-Box est la moins consommatrice d'énergie avec seulement 1,92 10^{-4} J. Par contre la 3ème approche proposée dans [61] (avec un $K_{min} = 35$) consomme le plus parmi les 5 méthodes étudiées. La consommation de la 5ème approche proposée par Fang [50] que nous avons adaptée pour construire les S-Box est proche de celle de la première méthode mais reste inappropriée pour un réseau de capteurs.

Intuitivement, choisir une suite chaotique discrétisée nécessite moins de calcul et ce sera plus adéquat pour les réseaux de capteurs sans fil (WSN) en raison de la simplicité et la rapidité de la représentation en point fixe. Cependant, la 3ème méthode consomme plus que les approches basées sur des récurrences à sorties réelles. On remarque que la suite chaotique employée "Skew tent map" repose sur des opérations relativement complexes comme la division, les opérateurs ceil et floor ainsi que la fonction de seuil qui est la fonction de base de l'opération de recherche par dichotomie.

Le choix de la suite, la simplicité de l'algorithme et la représentation arithmétique convenable sont des facteurs déterminants dans l'implémentation des S-Box dynamiques à faible consommation.

Tableau 4. 6. Consommation en énergie pour la construction des S-Box.

	1ère approche ($x_0 = 0.76$)	2ème approche ($x_0 = 0.76$)	3ème approche (K = 35)	4ème approche ($x_k = 0$; $y_k = 220$; $z_k = 62100$)	5ème approche (x = 1200, y = 30000)
Energie (10^{-4}J)	63,72656	286,50891	779,33877	1,92234	59,39766

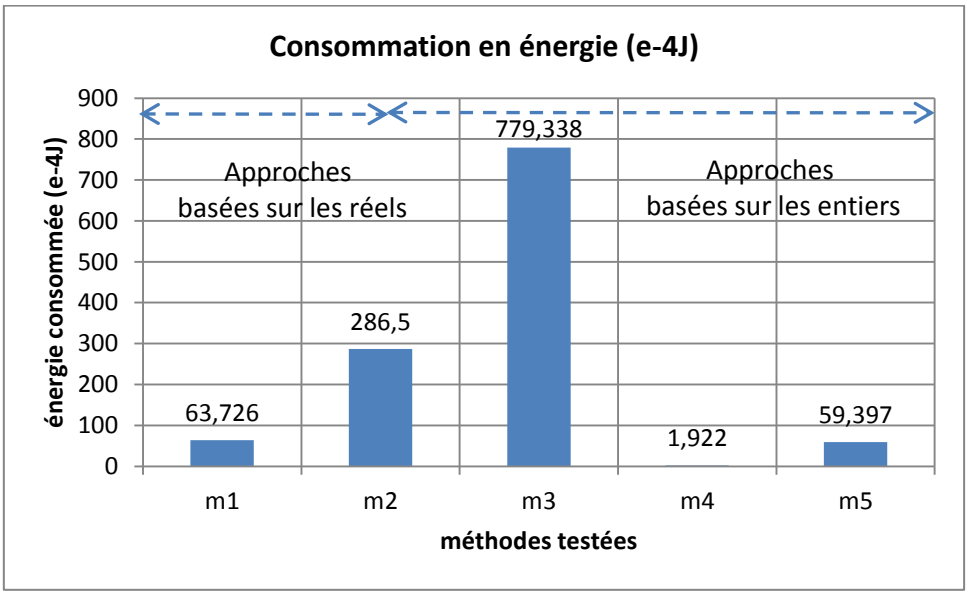


Figure 4. 26. Energie consommée pour les 5 approches de S-Box chaotiques.

b- *Energie consommée par l'algorithme ACIL*

Nous avons testé notre algorithme ACIL sous Wsim/Wsnet en cryptant et décryptant les données sur un même nœud et sur des nœuds différents. La clé utilisée est citée dans le tableau 4.4 (clé1). Avec Esimu nous avons évalué le nombre de cycle et l'énergie consommée en Joule pour chaque fonction du code.

La figure 4.27 montre l'énergie consommée en Joule des fonctions de cryptage et de décryptage pour un bloc de 16 octets. L'énergie consommée par les deux fonctions de cryptage et de décryptage est presque la même ($3{,}7756\times10^{-5}$ J et $3{,}7767\times10^{-5}$ J respectivement). L'algorithme ACIL consomme en total 7.5523×10^{-5} J. Ce résultat montre que notre algorithme préserve l'énergie et il peut être implémenté sur des nœuds capteurs réels sans nuire à leur performance. Cette consommation réduite s'explique par le bon choix de la suite chaotique et la simplicité des fonctions de base de notre algorithme. Une optimisation et diminution des nombres d'instructions est envisagée en adoptant une programmation parallèle et non séquentielle du code.

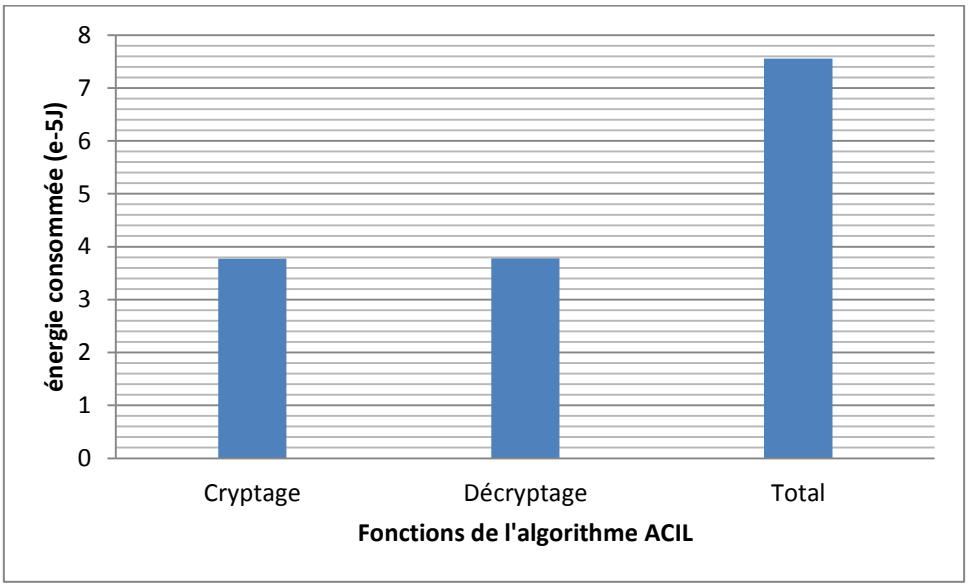


Figure 4. 27. Consommations des fonctions de cryptage et décryptage de ACIL mesurées sous wsim.

c- Comparaison avec l'AES standard

La sécurité de la norme 802.15 est assurée par l'algorithme AES (128 bits). Dans le but d'intégrer des S-Box dynamiques chaotiques dans cet algorithme, nous devons tout de même voir le coût de consommation qu'ajoute cet aspect dynamique à l'AES standard avec une S-Box statique. La S-Box statique de l'AES standard que nous avons implémentée n'est qu'une look up table.

Le tableau 4.7 et la figure 4.28 représentent la consommation détaillée des fonctions de l'algorithme AES que nous avons implémenté sur le simulateur de la plateforme MSP430 pour crypter 16 octets avec une clé de 128 bits. On constate que les fonctions qui consomment le plus sont l'établissement des clés et les mixsubcolumn et inv-mixsubcolumn. L'algorithme AES standard consomme donc en total 8.07×10^{-5}J.

En comparant avec la consommation des S-Box dynamiques, nous obtenons le tableau 4.8 et la figure 4.29 qui regroupent le surplus de consommation par rapport à l'AES statique.

Le tableau 4.9 est un tableau récapitulatif de consommation d'énergie et de cycle de toutes les S-Box étudiées.

Malgré la faible consommation de certains algorithmes de construction de S-Box dynamiques, le surplus d'énergie ($4{,}31\times10^{-4}$J pour la meilleure approche) représente un inconvénient qu'ajoute l'aspect dynamique à l'algorithme AES statique.

On peut compenser cet inconvénient par la diminution de la taille de la S-Box tout en augmentant la fréquence de leur génération. Cette idée reste à vérifier et à valider.

Tableau 4. 7. Consommation détaillée de l'algorithme AES (pour 16 octets et une clé de 128 bits)

Fonctions implémentées	**Consommation**
Fonction de cryptage	993002892,75 fJ = 0.00993 10^{-4} J
Fonction mixsubcolumn (subbyte+mixcolumn)	0,000015283 J
Fonction shiftrow	6,62 10^{-6} J
Fonction de décryptage	1081808029,5 fJ = 0,01082 10^{-4} J
Fonction invmixsubcolumn	0,000023816 J
Fonction invshiftrow	6,62 10^{-6} J
Établissement de la clé	16477389464,25 fJ = 0.164 10^{-4} J
Fonction Add round Key	0,000007774 J

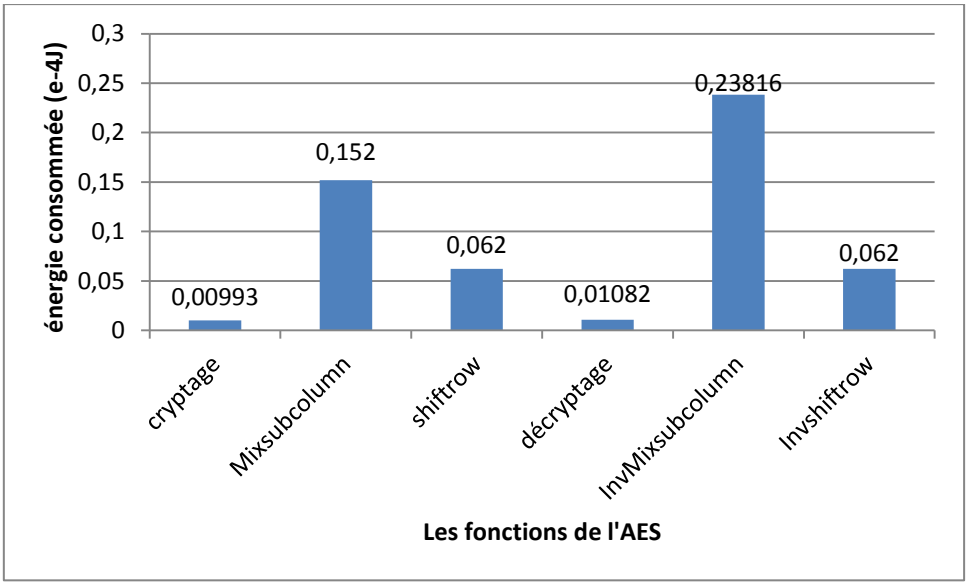


Figure 4. 28. Consommation détaillée de l'algorithme AES (pour 16 octets et une clé de 128 bits).

Tableau 4. 8. Surplus de consommation par rapport à l'AES statique.

	1ère méthode (x_0=0.76)	2ème méthode (x_0=0.76)	3ème méthode (K=35)	4ème méthode (x_k=0 ; y_k= 220; z_k=62100)	5ème méthode (x=1200, y=30000)
Energie (10^{-4}J)	66,083	288,976	781,724	4,319	61,763

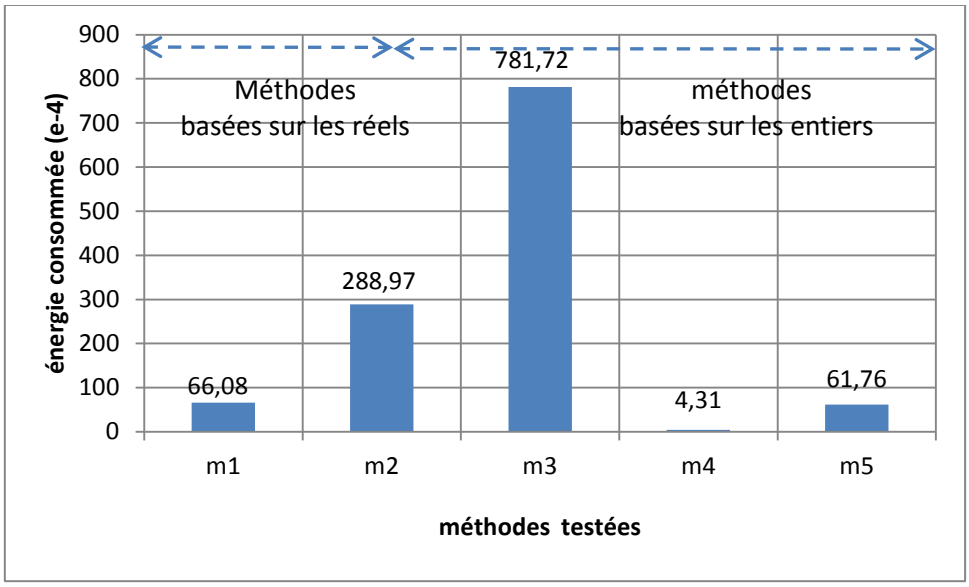


Figure 4. 29. Surplus de consommation par rapport à l'AES statique.

Tableau 4. 9. Tableau récapitulatif de consommation d'énergie et de cycles.

Types de S-Box	**Méthodes**	**consommation**	**cycles**
SBox statique	AES Enc/Dec	80731942500 fJ ($8.073194\ 10^{-5}$ J)	6866950
	AES Enc	50456340000 fJ ($5.045634\ 10^{-5}$ J)	4637825
	Etablissement des clés	$0.164\ 10^{-4}$ J	
Sbox dynamique	tri combsort x_0=0.76 (la même pour toute conditions initiales)	Total:$6.689048\ 10^{-3}$ J	499237950
		Sboxfunction=0,000108363J Combsortfunction=0,000512381J Sbox = 0,006372656 J	
		Gmul = 0,000228097 J	
		Mixsubcolumn =0,000022074 J	
		Invmixsubcolumn =0,000024081J	
	Dichotomie x_0=0.76	Total = $2.897835\ 10^{-2}$ J	2161161325
		Sbox = 0,028650891 J Sboxfunction = 0,001701029 J	
		Gmul = 0,000228929 J	
		Mixsubcolumn = 0,000023183 J	
		Invmixsubcolumn = 0,000023183J	
	Skew tent map discrétisé k_{min}=35	Total: $7.853071\ 10^{-2}$ J	5870395200
		Sboxfunction : 0,001596364 J Sbox : 0,077933877	
		Gmul = 0,000227739J	
		Mixsubcolumn = 0,000251298 J	
		Invmixsubcolumn = 0,000023559 J	
	Lorenz map conditions initiales (x_k=0 ; y_k= 220; z_k=62100)	Total : $5,1262365\ 10^{-4}$ J	38445256
		Sbox : 0,000192234	
		Gmul : 0,000227246J	
		Mixcolumn : 0,000021735	
		Invmixcolumn : 0,000024401	
	N-Logistic-tent map (x=1200, y=30000);	Total : $6.256996\ 10^{-3}$ J	-
		Sbox : 0,005939766 J	
		Gmul : 0,000229006 J	
		Mixcolumn : 0,00025466J	
		Invmixcolumn : 0,000024402 J	

4.3.6.2. Etude de la consommation via SensLab

Pour évaluer la consommation en énergie, nous avons mesuré le temps d'exécution (via Timer B) et la puissance (mesure de tension et de courant) nécessaires pour chaque code pour crypter et décrypter 16 octets.

a- Energie consommée par les algorithmes de construction des S-Box chaotiques

La figure 4.30 regroupe les valeurs de consommations en énergie de l'algorithme AES standard ainsi que les algorithmes AES modifiés avec différentes méthodes (proposées et existantes) de construction de S-Box chaotiques.

La méthode 3 reste la plus consommatrice avec 1604.477 10^{-4} J, tandis que la méthode 4 a eu 16.32 10^{-4} J seulement.

Nous avons vérifié le surplus d'énergie qu'ajoute la construction des S-Box chaotiques par rapport à l'AES standard. En effet, l'énergie consommée par la méthode 4 est presque dix fois supérieure à celle consommée par l'AES standard. De ce fait, l'AES standard garde les meilleures performances de conservation d'énergie pour un réseau de capteurs sans fil.

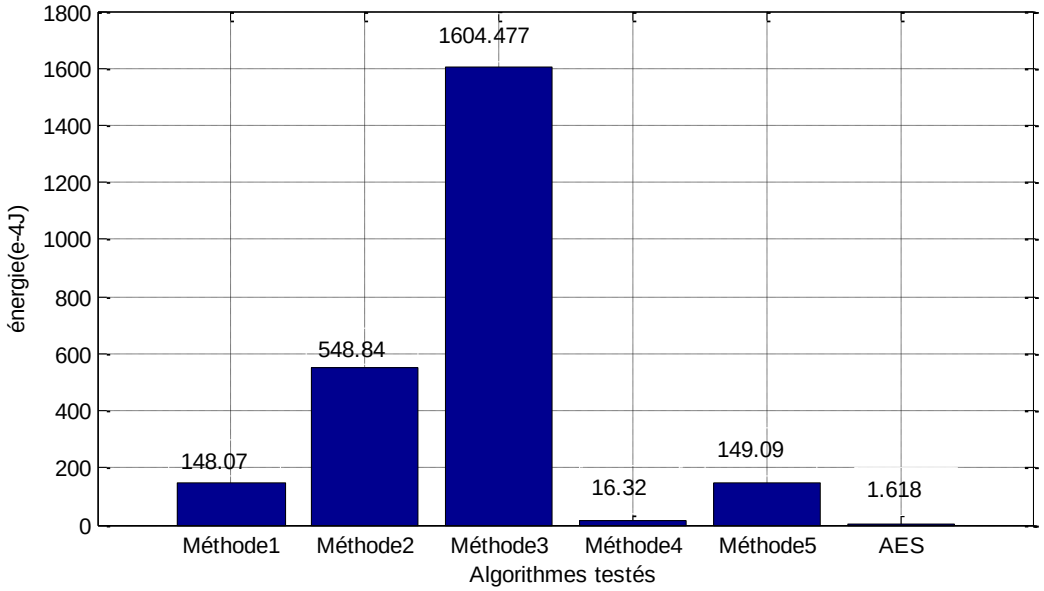


Figure 4. 30. Consommation en énergie pour la construction des S-Box sur la plateforme SensLab.

b- *Energie consommée par l'algorithme ACIL*

Nous avons mesuré l'énergie que consomme le cryptage et décryptage des données par notre algorithme ACIL sous les nœuds capteurs de la plateforme SensLab. Le résultat est illustré par la figure 4.31. L'énergie consommée par ACIL est 8,97 10^{-5} J. Cette valeur confirme les bonnes performances obtenues et observées par la simulation. Comparés avec les résultats obtenus par l'AES standard et l'AES avec S-Box chaotique (l'algorithme ayant la meilleure consommation), nous identifions clairement à travers la figure 4.32 que notre algorithme

ACIL est l'algorithme le plus adapté à la sécurité des réseaux de capteurs puisqu'il a les meilleures performances de préservation d'énergie.

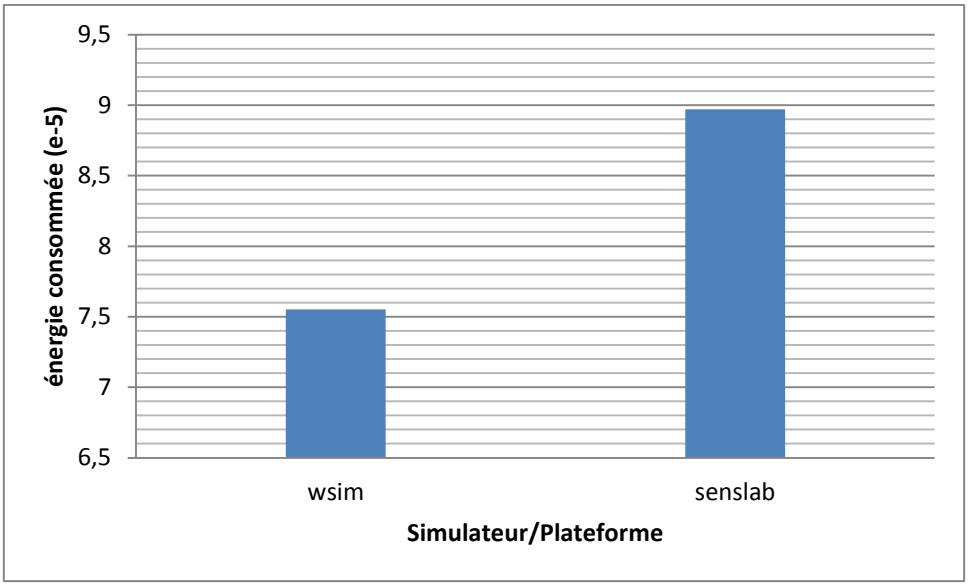


Figure 4. 31. Consommation de l'algorithme ACIL sous WSIM et SensLab.

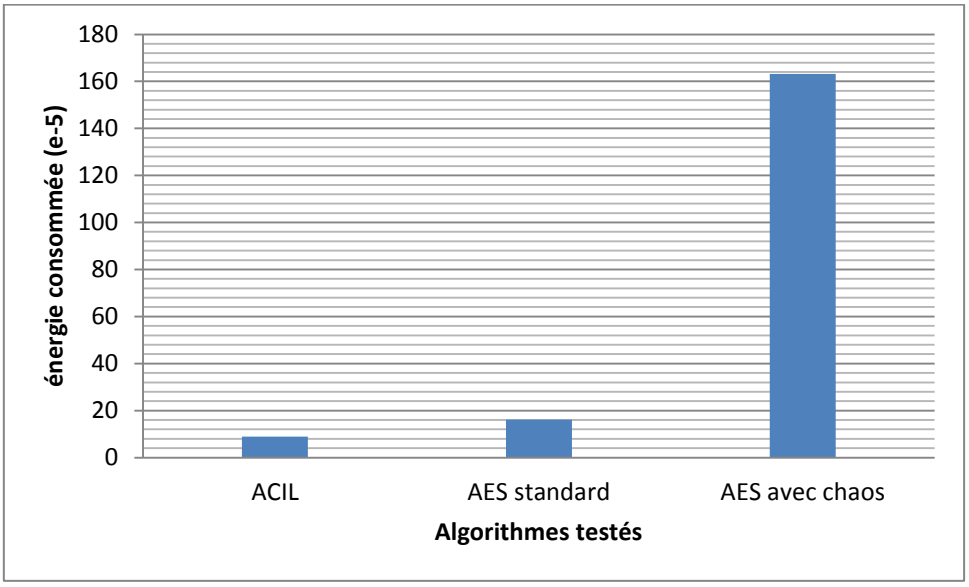


Figure 4. 32. Comparaison de l'énergie consommée des algorithmes ACIL, AES standard et AES avec chaos sous SensLab.

Les résultats expérimentaux montrent une augmentation de la consommation en énergie par rapport aux valeurs évaluées par simulation (Esimu). Ceci peut être expliqué par la simple raison que les modèles appliqués pour estimer l'énergie dans Esimu sont plus optimistes que la réalité et restent destinés à des tests de longues durées.

4.4. Conclusion

Dans ce chapitre, nous avons proposé un nouvel algorithme de cryptage d'image dédié au réseau de capteurs sans fil. ACIL est un algorithme simple mais efficace en mode CBC qui applique les techniques de diffusion, de confusion et de whitening. Nous avons montré par une analyse de sécurité statistique et différentielle que notre algorithme ACIL est fiable et peut résister aux techniques de cryptanalyses classiques.

Comparé à quelques algorithmes de cryptage d'image par chaos de la littérature, ACIL dévoile de meilleures performances. Ces résultats nous ont incités à implémenter nos algorithmes proposés et à tester leur consommation d'énergie par la simulation et par la mesure pratique sur la plateforme nationale SensLab. En effet, nous avons utilisé Wsim, Wsnet et Esimu pour simuler les codes et mesurer l'énergie consommée par chacun. Nous avons constaté que les S-Box chaotiques calculées instantanément pour le cryptage de chaque bloc de données, ajoutent un surplus d'énergie, par rapport à l'algorithme AES standard. Ce surplus est variable suivant la suite chaotique appliquée mais reste borné. En revanche, la consommation en énergie de l'algorithme ACIL est très réduite comparée à ceux de l'algorithme AES standard et à l'AES avec S-Box chaotiques.

Nous avons par la suite validé nos contributions par une implémentation pratique sur un réseau de capteurs sans fil accessible via la plateforme SensLab. Les mesures expérimentales confirment les résultats de simulation par Esimu. Nous pouvons conclure que ACIL, avec ses bonnes performances, est adapté à la sécurité des réseaux de capteurs sans fil.

Conclusion Générale

Nos travaux de thèse nous ont conduits à étudier dans un premier temps les réseaux locaux sans fil puis les réseaux de capteurs sans fil. En énumérant les caractéristiques et les propriétés des capteurs et des réseaux de capteurs sans fil, nous avons pu remarquer que les contraintes matérielles telles que la puissance de calcul réduite, la mémoire limitée, le faible débit et principalement la faible puissance des batteries avec un approvisionnement contraint d'énergie, constituent un challenge pour la sécurité des données et les techniques de chiffrement à appliquer.

Dans un deuxième temps, nous avons présenté les techniques de cryptographie et de cryptanalyse classiques et les différentes attaques probables pour un réseau de capteurs sans fil. La protection des données transmises par chiffrement est l'une des solutions pour contrarier ces menaces. Cependant, la sécurité des informations dans un réseau de capteurs par des algorithmes de chiffrement conventionnel présente des inconvénients (mémoire occupée, énergie dissipée, robustesse…).

En l'occurrence, le chaos est une alternative pour remplacer ou améliorer les techniques de chiffrement classiques déjà existantes.
Nous avons analysé, dans cette optique, les propriétés des signaux chaotiques. En effet, le chaos est un phénomène observé dans la nature, l'écologie, les systèmes électriques et des nombreux domaines. Il se caractérise par une large bande, une extrême sensibilité aux conditions initiales et il est difficilement prédictible.

Nous avons validé par une série de tests statistiques de NIST l'aspect aléatoire d'un ensemble de récurrences chaotiques à des dimensions différentes. Nous avons montré que la méthode de conversion en binaire affecte l'aspect aléatoire de ces récurrences. Le passage du réel au binaire a été réalisé à l'aide de deux méthodes : la troncature et la représentation IEEE 754 simple précision.

Nous avons mis en relief par la suite deux axes de modélisation de cryptosystèmes chaotiques : analogiques et numériques. Etant donné que les cryptosystèmes chaotiques analogiques présentent plusieurs inconvénients tel que la difficulté de synchronisation, et ne conviennent pas à notre application, nous avons focalisé sur les cryptosystèmes numériques et nous avons étudié des algorithmes de chiffrement par bloc pour les réseaux de capteurs. Nous avons proposé par ailleurs une classification des algorithmes de construction des S-Box chaotiques qui se divisent en deux classes : les S-Box à base de suite chaotique avec des nombres réels et les S-Box à base de suite chaotique avec des nombres entiers. Nous avons proposé de nouvelles méthodes de construction de S-Box pour les deux cas. Nous avons montré que nos S-Box satisfont les critères d'évaluation de sécurité tels que le critère d'avalanche stricte, la probabilité d'approximation linéaire et la distribution équiprobable des différentielles d'entrée/sortie. De plus, nous avons prouvé que nos algorithmes de construction de S-Box assurent une meilleure sécurité en comparant leurs résultats avec la littérature.

Malgré le grand nombre d'études effectuées sur la cryptographie par chaos, l'impact de ces recherches reste minimal sur la cryptographie conventionnelle puisqu'elles utilisent les

nombres réels et elles ne sont pas adéquates pour une implémentation et une réalisation pratique.

De ce fait, nous nous sommes orientés vers une application pratique pour les S-Box chaotiques. Nous avons choisi d'apporter des modifications à l'algorithme AES standard en générant son S-Box dynamiquement par chaos pour chaque session établie et s'assurer que les résultats de cryptage et de décryptage sont corrects. Nous avons proposé aussi un nouveau cryptosystème chaotique nommé ACIL applicable au réseau de capteurs sans fil. La permutation cyclique des blocs de pixels assure la diffusion et l'utilisation d'une récurrence chaotique permet de réaliser la confusion. Nous avons montré ensuite que ACIL est robuste contre les attaques de cryptanalyse classiques. Nous avons effectué une comparaison de ses performances par rapport à des algorithmes de chiffrement d'image dans la littérature. ACIL dévoile des résultats meilleurs avec un temps d'exécution inférieur sous MATLAB. Pour implémenter nos algorithmes sur une plateforme de réseaux de capteurs, nous les avons testés en premier lieu avec les simulateurs Wsim et Wsnet. Nous avons alors adapté les algorithmes de construction des S-Box chaotique aux contraintes de limitation de puissance de calcul qui nous ont mené à avoir des cycles courts pour les récurrences dicrétisées de Lorenz et N-Logistic-Tent. Une solution à ce problème était d'introduire des perturbations à l'aide d'un LFSR. Nous avons mesuré ensuite leur consommation en énergie à l'aide de Esimu.

Le chaos modulaire ou libre bien qu'il ajoute un aspect dynamique et une sécurité vérifiée contre les attaques classiques linéaires et différentielles ainsi qu'une sécurité potentielle contre les attaques encore théoriques (algébrique), il permet d'augmenter la consommation de l'algorithme AES modifié. Le coût de consommation d'une S-Box de taille relativement grande par rapport aux faibles ressources dont dispose un réseau de capteurs sans fil reste relativement élevé. La simulation a montré que le surplus de la meilleure méthode dépasse 10^{-4} J ; d'où l'idée de diminuer la taille de la S-Box et augmenter par contre la fréquence de sa génération afin de réduire la consommation. Ceci n'est pas le cas de l'algorithme ACIL puisque sa consommation en énergie est inférieure à l'algorithme AES standard et à l'AES avec S-Box chaotiques.

Le dernier volet de cette thèse concerne la mise en pratique des différentes contributions à l'aide de SensLab qui est une plateforme de réseaux de capteurs à large échelle avec 1024 nœuds. Les nœuds capteurs sont dotés d'un microcontrôleur Ti MSP430-f1611 à 16 bits et une interface radio Ti CC2420 ou CC1101. Les résultats expérimentaux ont été en concordance avec les résultats de simulation. Nous avons pu alors confirmer les bonnes performances de notre contribution qui est bien adaptée à la sécurité des trames de données dans un réseau de capteurs sans fil.

Parmi les perspectives de recherches que nous envisageons, tout d'abord concevoir un protocole d'auto-organisation du réseau pour la génération dynamique des S-Box chaotiques dans le mode CCM. La taille des S-Box générées peut être réduite en fonction des besoins de sécurité afin de permettre une consommation adaptative d'énergie.
Le deuxième point est d'étendre la clé de notre algorithme ACIL en étudiant les paramètres de l'équation de Lorenz discrétisée à l'aide des exposants de Lyapunov et des tests de NIST. L'optimisation du processus de décryptage en utilisant la programmation parallèle constitue aussi une perspective envisageable de nos travaux.
Nous proposons l'exploitation et l'adaptation des mécanismes de constructions des S-Box proposés pour générer des sous clés et d'intégrer ACIL dans un processus de chiffrement basé sur le réseau Feistel où la génération des sous clés et des S-Box pour la phase de substitution

sera automatisée. Il est indispensable d'étudier la consommation d'énergie dans le cas où l'application sera les réseaux de capteurs sans fil.
Toute proposition d'algorithme de cryptanalyse fera évoluer nos contributions et ouvrira des pistes vers le développement de la cryptographie par chaos appliquée aux réseaux de capteurs sans fil.

Bibliographie

[1] Sanjay Ahuja, Pavan Potti, “Evolution of Wireless LAN Security”, Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA) , 2 Volumes, pp 599-605, Las Vegas, Nevada, USA, 14-17 July, 2008.

[2] Ke-Lin Du, Madisetti N. S. Swamy, Wireless Communication Systems: From Rf Subsystems to 4g Enabling Technologies, Cambridge: Univ. Press, 2010.

[3] Umar Amjad, Mobile Computing and Wireless Communications: Applications, Networks, Platforms, Architectures, and Security. [S.l.]: NGE Solutions, 2004.

[4] Saroj Kumar Panigrahy, Sanjay Kumar Jena, Ashok Kumar Turuk, “Security in Bluetooth, RFID and wireless sensor networks”. ICCCS 2011: 628-633. Proceedings of the 2011 International Conference on Communication, Computing & Security, (ICCCS), Odisha, India,12-14 February, 2011.

[5] J. Misic and V. B. Misic. Wireless Personal Area Networks: Performance, Interconnections And Security With IEEE 802.15.4. John Wiley & Sons, Mar. 2008.

[6] T.B. Gosnell, J.M. Hall, C.L. Ham, D.A. Knapp, Z.M. Koenig, S.J. Luke, B.A. Pohl, A. Schach von Wittenau, and J.K. Wolford. Gamma-Ray Identification of Nuclear Weapon Materials. Technical Report DE97053424. Lawrence Livermore National Lab., Livermore, CA (USA). February 1997.

[7] M. J. Brown. Users Guide Developed for the JBREWS Project. Technical Report LA-UR- 99-4676. Los Alamos National Laboratory of California University. 1999.

[8] I.F. Akyildiz, Su. Weilian, Y. Sankarasubramaniam, E. Cayirci, A survey on sensor networks, IEEE communications magazine,Volume: 40 , Issue: 8 Page(s): 102 – 114, August 2002.

[9] T. J. Dishongh, M. McGrath, B. Kuris. Wireless Sensor Networks for Healthcare Applications. Boston, Mass: Artech House, 2008.

[10] Zheng, Jun. Wireless Sensor Networks: A Networking Perspective. Hoboken, N.J: Wiley, 2009.

[11] A. Menezes, P. VanOorschot, S. Vanstone, Handbook of applied cryptography, 1997 by CRC Press.

[12] Teddy Furon, Thèse présentée pour obtenir le grade de docteur de l’Ecole Nationale Supérieure des Télécommunications Spécialité : Signal et Images Application du tatouage numérique à la protection de copie Soutenue le 22 mars 2002.

[13] Bruce Schneier, Applied Cryptography, Second Edition: Protocols, Algorithms, and Source Code in C (cloth), John Wiley & Sons, Inc, 01/01/96.

[14] Estelle Cherrier, Estimation de l’état et des entrées inconnues pour une classe de systèmes non linéaires, Thèse spécialité automatique et traitement du signal, 26 octobre 2006.

[15] Touradj Ebrahimi, Franck Leprévost, Bertrand Warusfel, Cryptographie et sécurité des systèmes et réseaux, Hermès - Lavoisier 2006.

[16] Aida Jnaidi, Anas Tarah**, "**AES immunity Enhancement against algebraic attacks by using dynamic S-Boxes", Information and Communication Technologies: From Theory to Applications, ICTTA 2008. 3rd International Conference, Issue, Page(s):1 – 6, 7-11 April 2008.

[17] Yang Xiao, Hsiao-Hwa Chen, Bo Sun, Ruhai Wang, and Sakshi Sethi, "MAC Security and Security Overhead Analysis in the IEEE 802.15.4 Wireless Sensor Networks", Hindawi Publishing Corporation EURASIP Journal on Wireless Communications and Networking, Volume 2006, Pages 1–12, 2006.

[18] Broer Vakgroep, The how and what of chaos, NAW 5/1 nr.1 maart 2000, pp 34-43, 2000.

[19] A statistical test suite for random and pseudorandom number generators for cryptographic applications. Revised NIST Special Publication 800-22. May 15, 2001. http://csrc.nist.gov/rng/.

[20] Pascal Chargé, Danièle Fournier-Prunaret, Véronique Guglielmi, Features analysis of a parametric PWL chaotic map and its utilization for secure transmissions, Chaos, Solitons and Fractals 38, pp 1411–1422, 2008.

[21] G Manjunath, D. Fournier-Prunaret, Construction of chaotic maps on non-convex spaces. A solution to a CSK Deciphering problem, *International Journal of Bifurcation and Chaos,* Vol. 20, 8, pp 2553-2559, August 2010.

[22] G Manjunath, D. Fournier-Prunaret, A.K. Taha, "A 3-dimensional Piecewise Affine Map used as a Chaotic Generator", European Conference on Iteration Theory September (ECIT), pp. 7 - 13, Yalta, Ukraine, 2008.

[23] R.Silva, R.Crespo, M. Nunes, "LoBa128, a Lorenz Based PRNG for Wireless Sensor Networks", Int. J. Communication Networks and Distributed Systems, 3(4):301-318, 2009.

[24] Gonzalo Alvarez and Shujun Li, Some Basic Cryptographic Requirements for Chaos-Based Cryptosystems, International Journal of Bifurcation and Chaos, vol. 16, no. 8, pp. 2129-2151, 2006.

[25] L. Pecorra, , T. Carrol, Synchronization in Chaotic Systems, Physical review letters, Vol. 64, No. 8, pp.821-824, 1990.

[26] S. Boccaletti, J. Kurths, G. Osipov, D. Valladares, C. Zhou, "The synchronization of chaotic systems," Phys. Rep. 366, 1–101, 2002.

[27] Tao Yang, "A Survey of Chaotic Secure Communication Systems", International Journal of Computational Cognition, Volume 2, Number 2, Pages 81–130, June 2004.

[28] Wolfgang Schwarz and Andreas Abel, Chaos communications-principles, schemes, and system analysis Proceedings of the IEEE [Proc IEEE]. Vol. 90, no. 5, 2002.

[29] Thomas Schimming, Statistical analysis and optimization of Chaos based broadband communications, Thèse doctorat en science technique école polytechnique fédérale de Lausanne 2002.

[30] C. M. Austin Andrew, an investigation into chaos-based communication systems, Auckland, New Zealand, 2007.

[31] Shujun Li, Analog Chaos-Based Secure Communications: A Survey. Talk presented at College of Information Engineering, Shenzhen University, Novembre, 2005, http://www.hooklee.com

[32] Soumyajit Mandal and Soumitro Banerjee, “A Chaos-based Spread Spectrum Communication System”, National Conference on Nonlinear Systems & Dynamics 1, Indian Institute of Technology, Kharagpur, December 28-30, 2003.

[33] G. Alvarez, F. Montoya, M. Romera and G. Pastor, “Chaotic cryptosystems,” in L. D. Sanson, ed., Proc. 33rd Annual International Carnahan Conference on Security Technology, 332–338, 1999.

[34] N. Masuda, and K. Aihara, “Cryptosystems with discretized chaotic maps,” IEEE Trans. Circuits Syst. I 49, 28–40, 2002.

[35] G. Pérez and H. A. Cerdeira, “Extracting messages masked by chaos,”Phys. Rev. Lett., vol. 74, no. 11, pp. 1970–1973, 1995.

[36] T. Yang, L. Yang, and C. Yang, “Breaking chaotic switching using generalized synchronization: Examples”, IEEE Transactions on Circuits and Systems I, Vol. 45, No. 10, pp.1062-1067, 1998.

[37] G. Alvarez, F. Montoya, M. Romera, G. Pastor,’Cryptanalyzing a discrete-time chaos synchronization secure communication system’, Chaos, Solitons & Fractals, Vol. 21, No. 3, pp.689-694, 2004.

[38] N.K. Pareek, Vinod Patidar, K.K. Sud, Discrete chaotic cryptography using external key, Physics Letters A 309, 75–82, 2003.

[39] Tong Zhou, Mingyan Yu, Yizheng Ye, “A Robust High-Speed Chaos-Based Truly Random Number Generator for Embedded Cryptosystems”, In proceeding of: Circuits and Systems, MWSCAS '06. 49th IEEE International Midwest Symposium on, Volume: 2, 2006.

[40] Shujun Li, Analyses and New Designs of Digital Chaotic Ciphers, Ph. D. Dissertation of Xi’an Jiaotong University, 20 juin 2003.

[41] Shujun Li, Xuanqin Mou, Yuanlong Cai, Zhen Ji and Jihong Zhang, On the security of a chaotic encryption scheme: problems with computerized chaos in finite computing precision Computer Physics Communications, 153(1):52-58, 2003.

[42] H. Zhou, X.-T. Ling, Problems with the chaotic inverse system encryption approach, IEEE Trans. Circuits and Systems–I 44 (3), 268–271, 1997.

[43] G. Jakimoski, L. Kocarev, “Chaos and Cryptography: Block Encryption Ciphers Based on Chaotic Maps”, IEEE transactions on circuits and systems: Fundamental Theory and Applications, vol. 48, N°2, pp. 163–169, February 2001.

[44] CHEN Shuai, ZHONG Xian-xin, Confidential Communication Through Chaos Encryption in Wireless Sensor Network, Journal of China University of Mining & Technology, 17(2): 0258–0261, 2007.

[45] A. Benjeddou, A.K. Taha, D. Fournier-Prunaret, R. Bouallegue, “A New Cryptographic Hash Function Based on Chaotic S-box”, CSNDSP, Austria, 23-25 July, 2008.

[46] Tao Xiang, A novel symmetrical cryptosystem based on discretized two-dimensional chaotic map, Physics letters. A, vol. 364, no3-4, pp. 252-258, 2007.

[47] E. Biham, “Cryptanalysis of the chaotic-map cryptosystem suggested at EUROCRYPT’91,” in Proc. Advances in Cryptology—EUROCRYPT’ 91. Berlin, Germany: Springer-Verlag, pp. 532–534, 1991.

[48] Chen S, Zhong XX, Wu ZZ, Block Chaos. Cipher for wireless sensor network. Sci China Ser F – Inform Sci; 51:1055–63, 2008.

[49] Jiyun Yang, Di Xiao, Tao Xiang Cryptanalysis of a chaos block cipher for wireless sensor network, 2010, Elsevier Commun Nonlinear Sci Numer Simulat 16, pp844–850, 2011.

[50] Qian Fang, Ying Liu, Xiaoqun Zhao, “A chaos-based secure cluster protocol for wireless sensor networks”, Kybernetika, Vol. 44, No. 4, 522—533, 2008.

[51] Rui Miguel Silva, Rui Gustavo Crespo, Mario Serafim Nunes, “Enhanced chaotic stream cipher for WSNs”, IEEE International Conference on Availability, Reliability and Security, p 210-215, February 2010.

[52] Chung-Ming Ou, “Design of Block Ciphers by Simple Chaotic Functions”, IEEE computational intelligence magazine, pp. 54–59, May 2008.

[53] M. Asim, V. Joeti, Efficient and Simple Method for Designing Chaotic S-Boxes, ETRI Journal, Volume 30, Number 1, pp. 170–172, February 2008.

[54] Yong Wang, Kwok-Wo Wong, Xiaofeng Liao, Tao Xiang, “A block cipher with dynamic S-boxes based on tent map”, Communications in Nonlinear Science and Numerical Simulation, 2008.

[55] Guoping Tang, Xiaofeng Liao, Yong Chen, “A novel method for designing S-boxes based on chaotic maps”, Chaos, Solitons and Fractals 23, 413–419, 2005.

[56] Guo Chen, Yong Chen, Xiaofeng Liao, An extended method for obtaining S-boxes based on three-dimensional chaotic Baker maps, Chaos, Solitons and Fractals 31, 571–579, 2007.

[57] Hao YUAN, Longyan LUO, Yong WANG, “An S-box Construction Algorithm based on Spatiotemporal Chaos”, International Conference on Communications and Mobile Computing IEEE Computer society, 2010.

[58] Ghada ZAÏBI, Fabrice Peyrard, Abdennaceur Kachouri, Danièle Fournier-Prunaret, On Dynamic chaotic S-Box, Global Information Infrastructure Symposium (IEEE GIIS) Juin 23th ~ 26th, Hammamet, Tunisia, 2009.

[59] Ghada Zaïbi,Fabrice Peyrard, Abdennaceur Kachouri, Daniele Fournier-Prunaret, Mounir Samet. "A new design of dynamic S-Box based on two chaotic maps” (regular paper). Dans: ACS/IEEE International Conference on Computer Systems and Applications (AICCSA 2010), HAMMAMET TUNISIA, 16/05/10-19/05/10.

[60] Fatih Özkaynak, Ahmet Bedri Özer, “A method for designing strong S-Boxes based on chaotic Lorenz system”, Elsevier Physics Letters A 374, 3733–3738, 2010.

[61] Guoping Tang, Xiaofeng Liao, “A method for designing dynamical S-boxes based on discretized chaotic map”, Chaos, Solitons and Fractals 23, 1901–1909, 2005.

[62] Biham E, Shamir A. Differential cryptanalysis of DES-like cryptosystems. J Cryptol; 4(1):3–72, 1991.

[63] Adams C, Tavares S. Good S-boxes are easy to find. In: Advances in cryptology: Proceedings of CRYPTO’89. Lecture Notes in Computer Science, p. 612–5, 1989.

[64] Rohiem, A.E.; Elagooz, S.; Dahshan, H , “A novel approach for designing the s-box of advanced encryption standard algorithm (AES) using chaotic map”,Radio Science Conference, 2005. NRSC 2005. Proceedings of the Twenty-Second National Volume, Issue, 15-17, Page(s):455 – 464, March 2005.

[65] On the design of S-Boxes A.F Webster and S. E Tavares, Copyright © 1998, Springer-Verlag, 1998.

[66] M. Matsui, “Linear Cryptanalysis Method of DES Cipher,” *Advances in Cryptology, Proc. Eurocrypt’93, LNCS 765*, pp. 386-397, 1994.

[67] Eli Biham, On Matsui’s Linear Cryptanalysis, 341-355 Springer, 1998.

[68] Alvarez, G., Montoya, F., Romera, M. & Pastor, G. “Cryptanalysis of a discrete chaotic cryptosystem using external key,” Phys. Lett. A 319, 334–339, 2003.

[69] Christophe Guyeux, Qianxue Wang, Jacques M. Bahi, A Pseudo Random Numbers Generator Based on Chaotic Iterations: Application to Watermarking WISM 2010, LNCS 6318, pp. 202–211, Springer-Verlag Berlin Heidelberg 2010.

[70] Arroyo, D., Alvarez, G., Fernandez, V, “On the inadequacy of the logistic map for cryptographic applications”, X Reniun Espaola Sobre Criptologa y Seguird ad de la Informacin (XRECSI) 1, 77–82, 2008.

[71] Peter Stavroulakis, Chaos applications in telecommunications, by Taylor & Francis, 2006.

[72] Xiao Huijuan Qiu Shuisheng Deng Chengliang, “A Composite Image Encryption Scheme Using AES and Chaotic Series”, IEEE First International Symposium on Data, Privacy and E-Commerce ©, 2007.

[73] R. Matthews, “On the derivation of a chaotic encryption algorithm”, Cryptologra, 13 (1), 29-42, 1989.

[74] T.Kavitha, D.Sridharan, “Security Vulnerabilities In Wireless Sensor Networks: A Survey”, Journal of Information Assurance and Security 5, 031-044, 2010.

[75] D. Socek, S. Li, S. S. Magliveras, B. Furht, “Enhanced 1-D Chaotic Key Based Algorithm for Image Encryption,” IEEE, Security and Privacy for Emerging Areas in Communications Networks, 2005.

[76] B. Furht, D. Socek, and A.M. Eskicioglu. Fundamentals of multimedia encryption techniques, in B. Furht and D. Kirovski (Eds.), Multimedia Security Handbook, Vol. 4 of *Internet and Communications Series*, Ch. 3, CRC Press, December 2004.

[77] S. Lian. A Block Cipher Based on Chaotic Neural Networks Elsevier, Neurocomputing, vol. 72, pp. 1296-1301, 2009.

[78] Ismail Mansour, Gerard Chalhoub, Bassem Bakhache, “Evaluation of a fast symmetric cryptographic algorithm based on the chaos theory for wireless sensor networks”, MWNS (International Symposium on Mobile Wireless Network Security, Liverpool, UK, 25-27 June, 2012.

[79] Abir Awad, Safwan El Assad, Daniel Carragata , “A Robust Cryptosystem Based Chaos for Secure Data”, IEEE, ISIVC Conference On, Image/Video Communications over fixed and mobile networks, Bilbao Spain, July 2008.

[80] Marwa Abd El-Wahed, Saleh Mesbah, and Amin Shoukry , Efficiency and Security of Some Image Encryption Algorithms, Proceedings of the World Congress on Engineering Vol I WCE, July 2 - 4, London, U.K, 2008.

[81] Hassan Noura, Safwan El Assad Calin Vladeanu, Daniel Caragata, An Efficient and Secure SPN Cryptosystem Based on Chaotic Control Parameters 6th International Conference on Internet Technology and Secured Transactions, Abu Dhabi, UnitedArab Emirates, 11-14 December 2011.

[82] Mintu Philip, Asha Das, Survey: Image Encryption using Chaotic Cryptography Schemes, IJCA Special Issue on Computational Science - New Dimensions & Perspectives NCCSE, 2011.

[83] Yanbing Liu, Simei Tian, Wenping Hu, Congcong Xing, “Design and statistical analysis of a new chaotic block cipher for Wireless Sensor Networks”, Communications in Nonlinear Science and Numerical SimulationVolume 17, Issue 8, Pages 3267–3278, August 2012.

[84] Shannon CE., Communication theory of secrecy system, Bell Syst Tech J.28; 656-715, 1949.

[85] V. Patidar, N.K. Pareek, G. Purohit, K.K. Sud , A robust and secure chaotic standard map based pseudorandom permutation-substitution scheme for image encryption, Optics Communications, 284, p. 4331-4339, 2011.

[86] Giuseppe Anastasi, Marco Conti, Mario Di Francesco, Andrea Passarella, “Energy conservation in wireless sensor networks: A survey”, Ad Hoc Networks 7(3): 537–568, 2009.

[87] Guanfeng Li, Hui Ling, Taieb Znati, and Weili Wu, “A Robust on-Demand Path-Key Establishment Framework via Random Key Predistribution for Wireless Sensor Networks”, Hindawi Publishing Corporation EURASIP Journal on Wireless Communications and Networking, issue 2, Pages 1–10, Avril, 2006.

[88] J.P.Kaps and B.Sunar, “Energy comparison of AES andSHA-1 for ubiquitous computing,” Lecture Notes in Computer Science, vol. 4097, pp. 372–381, 2006.

[89] Simulateur WSim http://wsim.gforge.inria.fr/.

[90] Simulateur WSNet http://wsnet.gforge.inria.fr/.

[91] Nicolas Fournel, Antoine Fraboulet, Paul Feautrier, “eSimu: a Fast and Accurate Energy Consumption Simulator for Real Embedded System”, In Proceedings of

WOWMOM: IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks, 2007.

[92] esimu on line http://esimu.gforge.inria.fr/

[93] KCachegrind. online, http://kcachegrind.sourceforge.net/, Mar, 2007.

[94] S. Tao,W. Ruili, et Y. Yixun, “Perturbance-based algorithm to expand cycle length of chaotic key stream,” Electronics. Letters, vol. 34, no. 9, pp. 873–874, 1998.

[95] Ranjan Bose, Member, Saumitr Pathak, “A Novel Compression and Encryption Scheme Using Variable Model Arithmetic Coding and Coupled Chaotic System”, Transactions On Circuits And Systems, Vol. 53, N°. 4, Avril, 2006.

[96] www.senslab.info

[97] Michael T. Rosenstein, James J. Collins, Carlo J. De Luca, “A practical method for calculating largest Lyapunov exponents from small data sets”, Physica D: Nonlinear PhenomenaVolume 65, Issues 1–2, 15, Pages 117–134, May 1993.

[98] Julian I. Palmore, Joseph L. McCauley, Shadowing by computable chaotic orbits, Physics Letters A, Volume 122, Issue 8, Pages 399-402, 29 June 1987.

Liste des publications

Articles dans des revues

1. **Ghada Zaïbi**, Fabrice Peyrard, Abdennaceur Kachouri, Daniele Fournier-Prunaret, Mounir Samet, Efficient and Secure Chaotic S-Box for Wireless Sensor Network, Security and Communication Networks, 2012, Wiley, ISSN: 1939-0114, Impact Factor: 0.414.
2. **Ghada Zaïbi**, Nejah Nasri, Abdennaceur Kachouri, Laurent Andrieux, Mounir Samet, "Cross-Layer Design For Energy-Efficient In Wireless UnderwaterCommunication: Acoustic Frequency IDentification Case Study", IJCSIInternational Journal of Computer Science Issues, Vol. 8, Issue 3, pp 602-609,May 2011 ISSN (Online): 1694-0814. IF: 0.242.
3. **Ghada Zaïbi**, Nejah Nasri, Abdennaceur Kachouri, Mounir Samet, "Survey Of Temperature Variation Effect On Underwater Acoustic Wireless Transmission ",ICGST International Journal on Computer Network and Internet Research, CNIR, vol2, pp 59-64, dec2009, ISSN(Online):1687-4862.

Chapitre de livre

1. **Ghada Zaïbi**, Fabrice Peyrard, Abdennaceur Kachouri, Daniele Fournier-Prunaret, Mounir Samet, A new encryption algorithm based on chaotic map for wireless sensor network, Architectures and Protocols for Secure Information Technology, 2013, IGI Global.(résumé accepté).

Communications dans des conférences internationales

1. Asma Belhaj Mohamed, **Ghada Zaïbi**, Abdennaceur Kachouri, "Implementationof RC5 and RC6 block ciphers on digital images", *8th international multiconference on systems, signals and devices* (*SSD*), pages 1-6, 2011, Sousse,Tunisia.
2. **Ghada Zaïbi**, Fabrice Peyrard, Abdennaceur Kachouri, Daniele Fournier-Prunaret, Mounir Samet, "A new design of dynamic S-Box based on two chaotic maps"(regular paper). Dans: *ACS/IEEE International Conference on Computer Systemsand Applications (AICCSA 2010*), (IEEE computer society), 16/05/10-19/05/10,Hammmet, Tunisia.
3. **Ghada Zaïbi**, Fabrice Peyrard, Danièle Fournier-Prunaret, Utilisation du chaos dans les chiffrements par blocs, 11eme Congrès des Doctorants EDSYS (Ecole Doctorale Systèmes) 6 et 7 mai 2010, Toulouse, France.
4. **Ghada Zaïbi**, Fabrice Peyrard, Abdennaceur Kachouri, Danièle Fournier-Prunaret, "On Dynamic chaotic S-Box", *Global Information Infrastructure Symposium* (*IEEE GIIS 2009*) Juin 23th ~ 26th, 2009, Hammamet**,** Tunisia.
5. **Ghada Zaïbi**, Fabrice Peyrard, Abdennaceur Kachouri, Danièle Fournier-Prunaret, "a comparative study of AES and chaotic cryptosystems", *9th international conferenceon sciences and techniques of automatic control and computer engineering STA 2008*,20-23 December, 2008 , Sousse, Tunisia.
6. Nada Rebhi, **Ghada Zaïbi,** Abdenaceur Kachouri, Pascal Charge, Danièle Fournier-Prunaret, "Chaotic UWB Communicatins for Low Rate WPAN Applications",

International Conference on Signals, Circuits and Systems (*IEEESCS'08),* 7-9 Novembre 2008, Hammamet, Tunisia.

Annexes :
Définitions et théorèmes

- **Algorithme de Tri :Comb sort**

Cet algorithme apparu en 1980 est un des plus simples algorithmes de tri. Il est une amélioration du tri à bulle et un rival de l'algorithme Quicksort.
Dans le tri à bulle, quand deux éléments sont comparés, ils ont toujours un gap (distance qui les sépare) de 1. L'idée de base de combsort est que le gap peut être supérieur à 1. Ce gap est initialement égal à la longueur de la liste à trier diviser par un facteur de retrait appelé "Shrink factor" généralement égal à 1.3. Cette valeur a été choisie après plusieurs essais de liste de valeurs pour conclure que cette valeur est la plus adéquate. La liste des éléments à trier est triée avec le résultat de la division (Cette valeur est tronquée si nécessaire).
Ce gap est divisé par le facteur de retrait encore une fois et le processus est répété jusqu'à avoir trié toutes les valeurs.

- **Bifurcation**

Une bifurcation correspond à un changement des propriétés d'un système comme l'apparition d'attracteur ou le nombre de point d'équilibre. Les bifurcations classiques correspondent aux dédoublements de période ou bifurcation fourche, etc. L'étude de ces bifurcations permet d'identifier l'apparition du chaos.

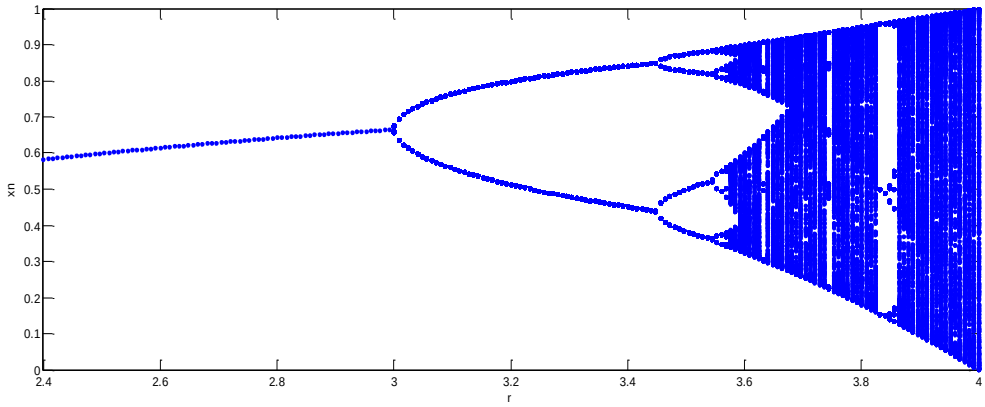


Figure A. 1. Diagramme de bifurcation de la suite logistique.

- **Exposant de Lyapunov**

Il existe des systèmes dynamiques (décrivant dans l'espace un état qui change dans le temps) qui sont très sensibles aux faibles variations de leurs conditions initiales.
Pour mesurer la vitesse d'amplification de ces petites variations, Alexender Lyapunov a conçu "l'exposant de Lyapunov" donné par la formule suivante :

$$\lambda_L = \lim_{k\to\infty} \frac{1}{k} \sum_{i=0}^{k-1} \ln \left| \frac{df(x_i)}{dx_i} \right|$$

λ_L est l'exposant de Lyapunov de la trajectoire $x_k=f^k(x_o)$.

Cet exposant définit le taux de séparation entre deux trajectoires à conditions initiales très proches et il permet de caractériser un attracteur. Si λ_L est positif, les deux trajectoires divergent (sensibilié aux conditions initiales), sinon elles convergent. Un système dynamique peut avoir plus qu'un exposant de Lyapunov, λ_i avec (i=1,2,3,…n). Le nombre de ces exposants sera égal à la dimension de l'espace de phase. Pour la présence d'un attracteur le système doit être dissipatif. La somme de tous les exposants de Lyapunov doit être négative même si plusieurs exposants sont positifs. La présence d'un exposant de Lyapunov positif est suffisante pour diagnostiquer le chaos [97]. Le signe du plus large exposant de Lyapunov (LLE) donne une indication sur l'attracteur. Pour un LLE > 0, l'attracteur est dit chaotique.
Si le nombre d'exposants de Lyapunov positifs dépasse un, on est dans l'hyperchaos**.**

- **Réseau de Feistel**

Le réseau de Feistel inventé par Horst Feistel est un réseau appliqué en premier lieu à l'algorithme Lucifer puis au DES et d'autres algorithmes de cryptages tel que : RC5, Twofish, Blowfish…
Il est simple et le cryptage et décryptage sont similaires. Il est basé sur des opérations de substitutions et de permutations avec une fonction principale changeant de clé à chaque tour.
La figure A.2 représente un exemple de cryptage et décryptage selon la structure de Feistel.

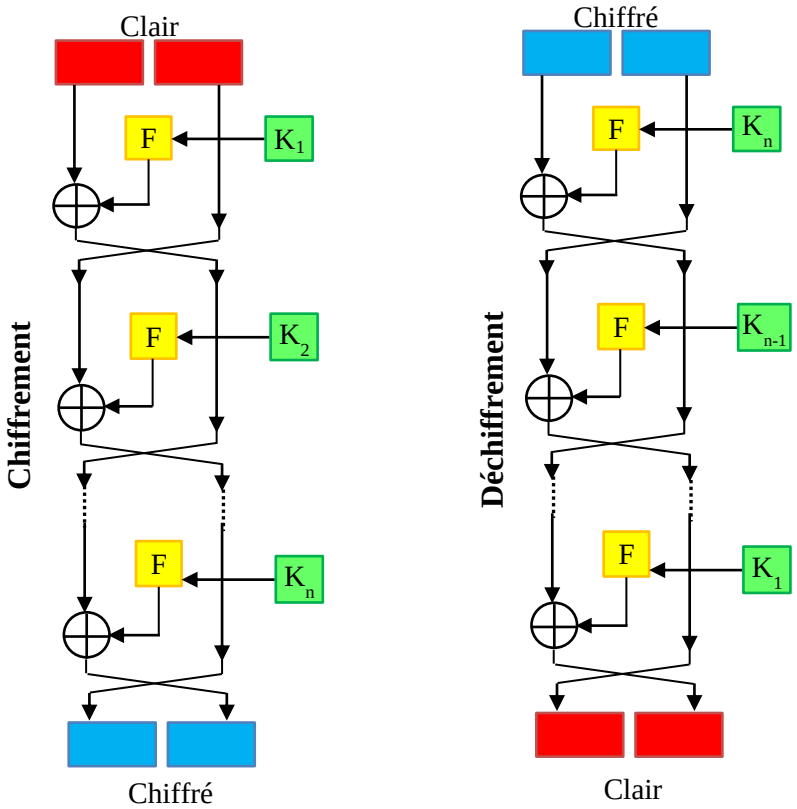


Figure A. 2. Réseau de Feistel à n tours utilisant l'opérateur XOR.

- **Masque jetable (One time Pad)**

Un masque jetable ou (one time Pad) est appelé aussi le chiffrement de Vernam est un chiffrement disant incassable. Il consiste à appliquer à l'aide d'un XOR à une donnée de longueur n octets, une clé aléatoire de même longueur. La clé ne doit être appliquée qu'une seule fois.
La difficulté dans ce type de cryptage est la condition de non réutilisabilité de la clé et l'aspect aléatoire de la clé (et non pas pseudoaléatoire), ainsi que la taille de la clé si la donnée est de taille importante.

- **Shadowing lemma**

Le lemme d'ombre ou le «b-shadowing lemma», cité pour la première fois par Anosov en 1967, est toujours utilisé pour justifier les simulations numériques des systèmes chaotiques. Le lemme d'ombre assure qu'il existe une orbite chaotique exacte proche d'une pseudo-orbite avec seulement une faible erreur.
Soit $\wedge$ un ensemble hyperbolique invariant. Alors pour tout $\beta > 0$, il existe un $\alpha > 0$, tel que chaque α-pseudo-orbite $\{x\}_{i=a}^{b}$ de $\wedge$ est β-ombrée par un point $y \in \wedge$.
Cependant, ce lemme est inutile pour le chaos numérique en raison des faits suivants :
Bien que les orbites d'ombrage existent réellement, elles sont triviales car elles sont généralement de mesure nulle. La mesure nulle des orbites périodiques peut être induite à partir des mesures nulles de l'espace discret dans un continuum. En outre, la discrétisation de l'espace des phases peut déstabiliser des orbites chaotiques stables de sorte que l'observation des orbites d'ombrage instable dans l'espace continu ne peut pas refléter la dynamique réelle des systèmes chaotiques numériques. Cette incapacité de la « b-shadowing lemma » en relation avec les orbites d'ombre chaotiques et instables est abordée dans [98].

- **Simulateur à cycle précis**

Un simulateur à cycle précis (CAS) est un programme machine qui simule une microarchitecture à cycle-précis.
Ils sont souvent employés en émulant un matériel plus ancien, où les précisions de temps est très importante des raisons.

- **Stéganographie**

La stéganographie est le fait de transmettre un message caché dans un autre, par exemple des bits discrets peuvent être cachés dans un texte, une image ou bien un programme. Pour le cas des messages multimédia (images), le bit de poids le plus faible d'un pixel est changé. Toute compression - décompression avec pertes de ces informations multimédias détruira les messages stéganographiques.

- **Variable d'état**

Une fonction qui définit l'état d'équilibre d'un système est appelée fonction d'état. Les variations de cette fonction dépendent uniquement des états initiaux et finals de son système à l'équilibre et non pas de la nature de la transformation.

- **Virgule fixe**

Un nombre fixe de chiffres après la virgule. Il existe des microcontrôleurs qui ne disposent pas d'unité de calcul en virgule flottante (FPU). Elle est utile à appliquer lorsque les processeurs ne possèdent pas un FPU ou bien lorsque ça permet d'améliorer les performances du système. Cependant un résultat en virgule fixe peut avoir plus de bits que les opérandes ; d'où une perte d'information, puisque les données seront par la suite arrondies ou tronquées.

- **Virgule flottante**

Un nombre en virgule flottante est représenté en deux parties la mantisse ou les chiffres significatifs et l'exposant ou la puissance à laquelle la base est élevée.

République Tunisienne
Ministère de l'Enseignement Supérieur,
de la Recherche Scientifique

Université de Sfax
École Nationale d'Ingénieurs de Sfax

Ecole Doctorale
Sciences et Technologies

Thèse de *DOCTORAT*
Génie électrique

N° d'ordre: 246– 2012

Sécurisation par dynamiques chaotiques des réseaux locaux sans fil au niveau de la couche MAC

Ghada ZAIBI

الخلاصة: تندرج اطروحة الدكتوراه في سياق السلامة المعلوماتية للشبكات المحلية اللاسلكية و شبكات الاستشعار اللاسلكية على وجه الخصوص. الهدف منها هو تقديم نظم تشفير تعتمد على الفوضى أكثر ملاءمة لشبكات الاستشعار من حيث استهلاك الطاقة مقارنة بالخوارزميات التقليدية بالإضافة الى تجربتها على منصة استشعار حقيقية. في البداية نقدم الشبكات اللاسلكية والقيود التي تعرقل عملية تأمين المعلومات مع تقديم تقنيات التشفير التقليدية. ثم نعطي لمحة عامة عن نظرية الفوضى و نقترح أساليب جديدة لصنع S-BOX قائمة على الفوضى نثبت قوتها ضد الهجمات التقليدية. وأخيرا، نقترح خوارزمية جديدة لتشفير الصور مخصصة لشبكة استشعار لاسلكية. يتم تنفيذ مساهماتنا عن طريق المحاكاة والقياسات التجريبية على منصة لشبكات الاستشعار الحقيقي.

Résumé : Les travaux de recherche de cette thèse s'inscrivent dans le cadre de la sécurité par chaos des réseaux locaux sans fil, en particulier les réseaux de capteurs sans fil. L'originalité de cette thèse consiste à proposer des cryptosystèmes à base de chaos plus adaptés aux réseaux de capteurs, en termes de consommation d'énergie, que les algorithmes conventionnels et à réaliser une implémentation sur une plateforme réelle. Nous présentons en premier lieu un état de l'art des réseaux, les menaces, les contraintes limitant le processus de sécurité des informations ainsi que les principales techniques de cryptographie. Nous donnons un aperçu sur la théorie de chaos et nous validons l'aspect aléatoire de plusieurs suites chaotiques par les tests statistiques du NIST. Nous proposons ensuite des nouvelles méthodes de construction de S-Box chaotiques tout en prouvant leur robustesse contre les attaques traditionnelles. Nous proposons enfin un nouvel algorithme de cryptage d'image dédié au réseau de capteurs sans fil. La validation de nos contributions est effectuée par simulation et par des mesures expérimentales sur une plateforme de réseaux de capteurs réels (SensLab).

Abstract: The security of wireless sensor network is a growing field of research hampered by limited battery life time and computing constraints. The originality of this thesis is to provide Low Power chaotic cryptosystems for sensor networks more suitable than conventional algorithms and achieve an implementation on a real platform.
We present first a state of the art of wireless networks, threats and constraints of the security process as well as conventional cryptographic techniques. We give an overview of the chaos theory and we validate the randomness of several chaotic maps by the NIST statistical tests. Then, we propose new methods of chaotic S-Box construction, while demonstrating their robustness against traditional attacks.
Finally, we propose a new image encryption algorithm dedicated to wireless sensor network. Validation of our contributions is performed by simulation and experimental measurements on a platform of real sensor networks (SensLab).

المفاتيح: شبكات الاستشعار, تشفير ,الفوضى,اختبارات NIST , S-Box, منصة SensLAB.

Mots clés: Réseau de capteurs sans fil, cryptographie, chaos, Tests du NIST, S-Box, SensLab.

Key-words: Wireless sensor network, cryptography, chaos, NIST tests, S-Box, SensLab.