République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université de Batna 2
Faculté de mathématiques et
D'informatique
Département d'informatique

Thèse

En vue de l'obtention du diplôme de

Doctorat en Informatique

Neuronal crypto-système basé sur un attracteur chaotique

Présentée Par

Merzoug Assia

Soutenue le: 20 / 02 / 2019

Membres du jury :

Président:	***Noui Lemnouar***	***Professeur***	***Université de Batna 2***
Rapporteur:	***Ali pacha Adda***	***Professeur***	***Université USTO Oran***
Examinateurs:	***Bilami Azeddine***	***Professeur***	***Université de Batna 2***
	Seghir Rachid	***MCA***	***Université de Batna 2***
	Guenda Kenza	***MCA***	***Université USTHB Alger***

ملخص

تم تطوير الشبكات العصبية لأول مرة لحل مشاكل المراقبة، التعرف على الأشكال أو الكلمات، أخذ قرار، للتخزين كبديل للذكاء الاصطناعي.

الشبكات العصبية الاصطناعية هي نماذج حاسوبية لشبكات الأتوماتا التي يتم نسخ بنيتها وسلوكها الى الخلايا العصبية الحقيقية. على طريقة الدماغ، يمكنهم التعرف على الأشكال، إعادة ترتيب البيانات وأخذها بطريقة أكثر أهمية. نجدها في مختلف المجالات مثل: الطيران، السيارات، الدفاع، الإلكترونيات، المالية، الطبية، الاتصالات. ونجدها أيضا مرتبطة لحل مشكلة تحليل الشفرات.

نقترح في هذا العمل ربطها بقيم أخرى مشتقة من جاذبات فوضوية لتأمين البيانات، اذن هدف الأطروحة متكون من إنشاء نظام تشفير مبني على شبكات عصبية اصطناعية وهو النظام الذي هو في الأساس مستوحى من المخططات لعمل الخلايا العصبية البيولوجية. استخدمنا جاذبا فوضويا لعمل نظامنا المسمى "نظام التشفير العصبي".

في سياق دراستنا، والتي تستند إلى دراسة تشكيل قيم عشوائية من سلسلة فيبوناتشي المشكلة من جهة أو خرائط فوضوية محددة لبناء تسلسلات تدريجية متزايدة، قمنا بتطوير العديد من أنظمة التشفير.

كلمات البحث: الشبكات العصبية، فيبوناتشي، التشفير، سلاسل تدريجية متزايدة، مشكل حقيبة الظهر، الفوضى، الجاذب

RESUME :

Les réseaux de neurones ont d'abord été développés pour résoudre des problèmes de contrôles, de reconnaissance de formes ou de mots, de décision, de mémorisation comme une alternative à l'intelligence artificielle.

Les réseaux de neurones artificiels sont des modèles informatiques de réseaux d'automates dont la structure et le comportement sont copiés sur ceux des neurones réels. A la façon de cerveau, ils peuvent reconnaître des formes, réorganiser des données et de façon plus intéressante à apprendre des données. On les trouve des dans divers domaines comme : Aérospatial, Automobile, Défense, Electronique, Finance, Secteur médical, Télécommunications, on les trouve aussi associé pour résoudre un problème de cryptanalyse.

On propose, dans ce travail de les associer à d'autres valeurs issues d'attracteur chaotique pour **sécuriser les données**, donc l'objectif de thèse consiste à réaliser un **crypto-système** basé sur les réseaux de neurones artificiels qui est un système dont la conception est à l'origine schématiquement inspirée du fonctionnement des neurones biologiques. Nous avons utilisé un attracteur chaotique pour réaliser notre système baptisé "Neuronal crypto-system".

Au cours, de notre démarche qui repose sur l'étude d'une part de la génération des valeurs aléatoires issus des suites de Fibonacci généralisées ou des cartes chaotiques spécifiques pour construire des suites super-croissantes nous avons développés et réalisé plusieurs systèmes cryptographiques.

Mots-clés : Réseaux de neurones, Fibonacci, cryptographie, suite super croissante, problème de sac à dos, chaos, attracteur.

ABSTRACT:

Neural networks were, first developed to solve problems of control, recognition of shapes or words, decision, memorization as an alternative to artificial intelligence.

Artificial neural networks are computer models of automata networks whose structure and behavior are copied to those of real neurons. In the brain way, they can recognize shapes, rearrange data and more interestingly learn. They are found in various fields such as Aerospace, Automotive, Defense, Electronics, Finance, Medical, and Telecommunications. They are, also found associated to solve a problem of cryptanalysis.

In this work, we propose to associate them with other values derived from chaotic attractors to secure the data, so the thesis objective consists in creating a crypto-system based on artificial neural networks, which is a system whose conception is originally schematically inspired by the functioning of biological neurons. We used a chaotic attractor to realize our system called "Neuronal crypto-system".

In the course of our study, which is based on the study of the generation of random values from generalized Fibonacci sequences or specific chaotic maps to build super-incremental sequences, we have developed and realized several cryptographic systems.

Keywords: Neural networks, Fibonacci, cryptography, increasing sequence, Knapsack problem, chaos, attractor.

Remerciement

" Quand un travail nécessite l'aide et la collaboration de plusieurs personnes, il est évident d'exprimer sa reconnaissance envers tous ceux la ".

Tous d'abord, je remercie le tout puissant de m'avoir donné patience et volonté pour arriver à ce stade " Dieu merci ".

En deuxième lieu je tiens à remercier mon encadreur Prof. Mr : Ali-Pacha Adda pour la confiance qui m'a accordée en acceptant d'encadrer ce travail doctoral, pour ses précieux conseils, son aide et sa patience; encore une fois merci Monsieur.

Je tiens à exprimer ma gratitude aussi à mon enseignant Prof. Mr : Noui Lemnouar de m'avoir aidé à la réussite de ce travail, ses énormes efforts à l'ouverture de notre nouvelle spécialité " Cryptographie et Sécurité " dans l'université Mostefa Ben Boulaïd -Batna 2." Mercie encore une fois Monsieur " .

Mes remerciements vont également aux membres de jury Prof. Mr :Noui Lemnouar, Prof. Mr :Bilami Azeddine, Dr. Mr : Seghir Rachid, Dr. Mme :Guenda Kenza. Qui ont accepté d'examiner ce travail, et leurs précieux conseils." Encore une fois Mercie Messieurs "

Je souhaiterais aussi adresser ma gratitude à tous ceux qu'ont participé de près ou de loin à la réussite de ce travail.

ASSIA

Dédicace

Je dédie cet humble travail :

A mes chers parents qui m'ont couvert de tendresse et l'amour et m'ont offert la précieuse éducation, qui m'a permet d'arriver à ce stade, " Que Dieu les récompense ".

A mon époux , qui m'a encourager à poursuivre le chemin de la science, sa longue patience, son aide moral et son soutient, " Que Dieu le garde pour moi".

A mon très cher bébé, tu es ma joie, ma raison de suivre ce chemin et te laisse une trace à suivre, maman te dédie cette thèse doctoral comme souvenir. Je t'aime mon bébé et je te souhaite tous le bonheur du monde.

A mes chers frères et sœurs qui sont mon honneur de vivre, et mon appui.

A mes beaux-frères et belles sœurs, je leurs doit tous respects son oublier mes neveux et nièces.

A ma belle-mère, je vous dois un grand respects et Mercie de m'avoir accueillis permet vous.

A mes collègues, de mon passage d'études doctoral.

" Que le tout puissant, accepte de moi ce humble travail, pour la jeunesse qui viendra après nous ".

ASSIA

Sommaire

Tableau des figures

Tableau des tableaux

Travaux réalisés dans le cadre de cette thèse

1. **Articles :**
    1. **Auteurs: Assia Merzoug,** Adda Ali-Pacha and Naima Hadj-Said
       Titre: New Approach of the Playfair's Cipher with a Numerical Value of the Keyword
       Journal: Indonesian Journal of Electrical Engineering and Computer Science Vol. 6, No. 3, June 2017, pp. 695 ~ 703

    2. **Auteurs: Assia Merzoug,** Adda Ali-Pacha , Naima Hadj-Said, Mustafa Mamat and Mohamad Afendee Mohamed
       Titre: Generating a random sequence based on PWLCM map: applicated to RC4 algorithm
       Journal: International Journal of Engineering & Technology, Vol. 7, No. 2, may 2018, pp. 182 ~185

    3. **Auteurs: Assia Merzoug,** Adda Ali-Pacha , Naima Hadj-Said
       Titre: New chaotic cryptosystemfor the image encryption
       Journal: Int. J. Information and Computer Security, Vol. X, No. Y, 2019

2. **Conférences :**
    1. International Workshop on Cryptography and its Applications - IWCA'16 - Université des Sciences et de la Technologie d'Oran - Mohamed BOUDIAF. 26 & 27 Avril 2016.
       Auteurs: Assia Merzoug, Adda Ali-Pacha and Naima Hadj-Said.
       Titre: Construction d'une Suite aléatoire par le biais de la Carte PWLCM : Application au RC4

    2. Colloque Tassili SCCIBOV'2015, Université Djillali Liabès de Sidi Bel Abbès, Faculté de Technologie, 02-03 Decembre 2015.
       Auteurs: Assia Merzoug, Adda Ali-Pacha and Naima Hadj-Said
       Titre: Construction d'une Suite Super-Croissante par le biais de la Carte Logistique : Application au Chiffre de Merkle-Helman

3. International Conference on Coding and Cryptography: ICCC2015, University of Science and Technology Houari Boumediene, USTHB, Alger 2 to 5 November 2015.
   Auteurs: Assia Merzoug, Adda Ali-Pacha and Naima Hadj-Said
   Titre: New Crypto-System based on a Concatenation of Two Chaotic Attractors for the Image Encryption.

4. 1er Journées Doctorales sur les Technologies de l'Information et de la Communication (JDTIC'14), Université Hadj Lakhdar, Batna, 04 au 05 Juin 2014
   Auteurs: Assia Merzoug, Adda Ali-Pacha and Naima Hadj-Said
   Titre: Le Problème du Sac à Dos dans le Chiffrement de Merkle-Hellman

5. International Conf. on Advances in Applied Mathematics and Mathematical Physics, Yildiz Technical University of Istanbul, Turkey, 19-21 Aug. 2014
   Auteurs: Adda Ali-Pacha**, Assia Merzoug,** and Naima Hadj-Said
   Titre: Modified Fibonacci's Sequence Applied to the Encryption of Data

3. Articles soumis à des revues :

1. Neuronal Crypto System based on Logistics Map and the Super-Increasing Sequence
   Auteurs: Assia Merzoug, Adda Ali-Pacha**,** and Naima Hadj-Said
   Soumis au journal : Int. J. Information and Computer Security

2. Adaption of the Generalized Fibonaci Sequence to Real Coefficients to Produce a Super-Increasing Sequence: Application to the Merkle-Helman cipher
   Auteurs: Assia Merzoug, Adda Ali-Pacha**,** and Naima Hadj-Said
   Soumis au journal : Information Security Journal: A Global Perspective

INTRODUCTION GÉNÉRALE

Introduction générale

Depuis l'antiquité, le problème pour dissimuler l'information et de communiquer secrètement a été l'un des intérêts principaux de l'humanité. Il a fourni à travers des époques successives, des efforts autant physiques qu'intellectuels pour pouvoir trouver une technique de dissimulation efficace et appropriée.

En effet, les modes de télécommunications sont en évolution continue avec la recherche permanente de meilleurs débits, de faciliter l'utilisation, de mobilité et surtout de confidentialité élevée.

La cryptologie est la science du secret, elle est un art ancien et une science nouvelle, elle est liée aux autres disciplines comme la théorie des nombres, l'algèbre ou encore la théorie de l'information. Elle est composée de deux branches : la cryptographie et la cryptanalyse.

La cryptographie traditionnelle utilisait différents outils pour dissimuler une information. Certains remplacent des mots par des nombres, d'autre mélangent, décalent ou permutent les lettres, pour rendre la lecture difficile voire impossible.

La cryptographie actuelle utilise des outils mathématiques pour construire des algorithmes qui permettent de transférer un texte en claire à un texte chiffré, cet algorithme dit algorithme de chiffrement, l'inversement d'algorithme qui permet de transférer un texte chiffré à un texte en clair est dit algorithme de déchiffrement. Ces algorithmes cryptographiques dépendent d'un paramètre qui est la clé (clé de chiffrement et clé de déchiffrement), on distingue deux types de la cryptographie à base de cette clé la cryptographie symétrique où la clé de chiffrement est identique à celle de déchiffrement, et la cryptographie asymétrique où la clé de chiffrement est différente à celle de déchiffrement.

L'inversement à la cryptographie, la cryptanalyse est la science qui utilise des procédés cryptographiques cherchant à décrypter des textes chiffrés sans la connaissance de la clé.

Donc la résistance des algorithmes cryptographiques aux attaques et à la cryptanalyse dépendent de la clé, plus la clé est longue et aléatoire plus la cryptanalyse est difficile, et vice versa. Alors on cherche un phénomène d'apparence aléatoire mais qui est déterministe à l'originale pour masquer l'information.

Il existe plusieurs systèmes présentant ce comportement, mais dans notre cas on utilise les systèmes chaotiques, ils sont caractérisés par la sensibilité aux conditions initiales et le déterministe. Le chaos est toujours associé à l'incompréhension des choses.

La cryptographie chaotique est appliquée dans les télécommunications et systèmes de transaction, l'idée est d'intégrer un message avec un signal chaotique pour perturber les attaques. La transaction chaotique est un mode de communication à clé secrète, c'est-à-dire cette clé est connue par l'émetteur et le récepteur pour le chiffrement et le déchiffrement du message. On doit alors mettre un signal chaotique au niveau de récepteur identique à celle de l'émetteur pour pouvoir récupérer le message masqué, mais on doit au préalable régler le problème de synchronisation des signaux.

L'objectif de thèse consiste à réaliser un crypto-système basé sur les réseaux de neurones artificiels, ou réseau neuronal artificiel, est un système dont la conception est à l'origine schématiquement inspirée du fonctionnement des neurones biologiques, et qui par la suite s'est rapproché des méthodes statistiques. Nous avons utilisé la théorie du chaos et particulièrement, le chaos algorithmique pour réaliser notre système baptisé "neuronal crypto-système".

Mais au cours, de notre démarche qui repose sur l'étude d'une part de la génération des valeurs aléatoires issus des équations déterministes ou principalement dans notre cas des cartes chaotiques spécifiques et d'autre part sur la construction des suites super-croissante, nous avons développé et réalisés plusieurs systèmes cryptographiques.

Cette thèse est organisée de quatre chapitres comme suit :

Le premier chapitre : Généralité sur la cryptographie

Dans le premier chapitre nous avons commencé par une brève historique sur la cryptographie. Ensuite nous avons parlé des principes de base de la cryptographie "terminologie". Apres nous avons présenté les deux classes de chiffrement : le chiffrement à clé secrète et le chiffrement à clé publique, et enfin nous avons donné quelques notions sur le chiffrement basé sur le chaos.

Le deuxième chapitre : Chaos

Dans le second chapitre nous avons donné une définition du chaos, ensuite nous avons présenté quelques cartes chaotiques qui nous intéressent. Nous avons vu leur

utilisation dans la cryptographie telle que la génération des nombres aléatoires, construction d'un crypto-système basé sur la concaténation de deux cartes chaotiques, et l'amélioration du crypto-système Playfair.

Le troisième chapitre : Construction d'une suite super-croissante

Dans le troisième chapitre nous avons commencé par l'introduction. Ensuite nous avons présenté deux méthodes pour la génération de la suite super-croissante, la première en utilisant la carte chaotique et la deuxième en utilisant la suite de Fibonacci. Et enfin nous avons appliqué cette suite pour la confidentialité des données.

Le quatrième chapitre : Neuronal crypto-système

Dans le dernier chapitre nous avons introduit les réseaux de neurones. Ensuite nous avons présenté notre crypto-système : la méthode de chiffrement, la méthode de déchiffrement et une évaluation de la clé utilisée dans ce crypto-système, ensuite nous avons présenté les résultats des trois tests de validation d'un crypto-système qui sont l'histogramme, l'entropie, et la corrélation entre les pixels adjacents.

Cette thèse se termine par une conclusion générale et un ensemble des perspectives.

CHAPITRE 1

GÉNÉRALITÉ SUR LA CRYPTOGRAPHIE

Chapitre 1

Généralité sur la cryptographie

1.1 Introduction

La cryptographie est une science qui permet de sécuriser la communication. Même si un espion intercepte les messages échangés, il ne peut pas les comprendre.

Avec la venue des technologies et les moyens de communications (réseaux locaux, internet, téléphonie mobile, satellite, transaction bancaire,...) qui exige la sécurité des données échangées. Ce qui permet non seulement aux personnes légitimes d'avoir accès aux informations sensibles mais aussi des personnes non légitimes (ou espions). Donc il faut empêcher cet accès aux personnes non autorisés. Alors la protection de ces informations (identité des personnes communiquant, données échangés, monnaie, mot de passe, …) est nécessaire. Pour répondre à cette problématique la cryptographie a développé de nombreux mécanismes dans différents domaines.

De nombreux systèmes de chiffrement ont été inventés, on peut les classer en deux classes : système à clé publique et système à clé privée.

Dans ce chapitre nous présentons les différentes notions de base de la cryptographie.

1.2 Historique sur la cryptographie

1.2.1 Période classique : l'antiquité

- Vers 600 ans avant .J.-C, le roi de Babylone Nabuchodonosor écrivait le message qu'il souhaitait transmettre à ses généraux, sur le crâne préalablement rasé de ses esclaves. Il attendait que leurs cheveux repoussent avant de les envoyer chez ces généraux, qui rasaient de nouveau les cheveux des messagers pour lire le texte.

- Dans laX^{ème}et VII^{ème} siècle avant J.-C les Grecs ont utilisé le chiffrement de la scytale spartiate [1] c'est un exemple de chiffrement par transposition. Des lettres étaient écrites sur une longue et mince bande de cuir enveloppée autour d'un cylindre, pour déchiffrer ces lettres, il devait faire un cylindre d'un diamètre identique à celui utilisé pour le chiffrement, il lui suffit d'enrouler la scytale autour de ce cylindre pour obtenir les lettres en clair. Le diamètre du cylindre était la clé.

Figure.1.1 Scytale

- Dans 200 avant J.-C apparait les premiers systèmes de cryptographie, ce sont les chiffrements par substitution ; il existe 4 types de substitutions :
  - **Mono-alphabétique :** Remplace chaque lettre du message par une autre lettre de l'alphabet.
  - **Poly-alphabétique:** Utilise une suite de chiffres mono-alphabétiques "la clé" réutilisée périodiquement.
  - **Homophonique :** Fait correspondre à chaque lettre du message en clair un ensemble possible d'autres caractères.
  - **Polygrammes:** Substitue un groupe de caractères dans le message par un autre groupe de caractères.
- Dans le1^{er} siècle avant J.-C lorsque Jules César envoyait des messages à ses généraux, il ne faisait pas confiance à ses messagers. Il remplaçait donc tous les A contenus dans ses messages par des D, les B par des E, et ainsi de suite pour tout l'alphabet. Seule la personne connaissant la règle du "décalage par trois" pouvait déchiffrer ses messages. Et voilà comment tout a commencé.

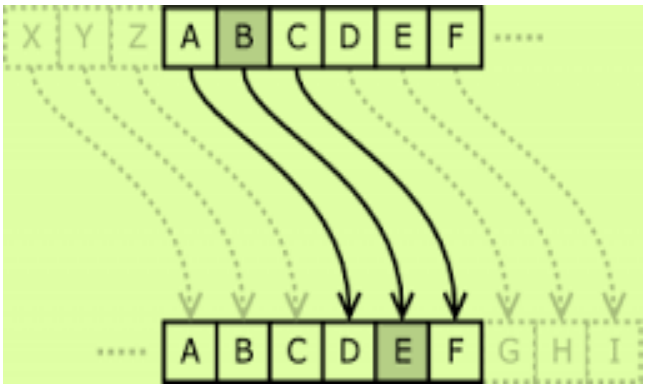


Figure.1.2 Chiffrement de César

1.2.2 Période de l'antiquité à la guerre

- En 1412, l'italien Leone Battista Alberti expose pour la première fois le chiffrement par substitution "poly-alphabétique" qu'il applique à l'aide d'un disque à chiffrer (Figure.1.3) et d'un mot clé. Ce dernier est constitué de deux disques composés de deux alphabets. Le petit disque étant mobile, et le second fixe.

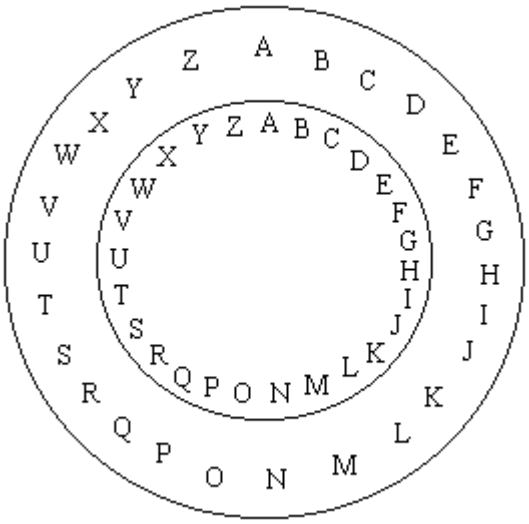


Figure.1.3 Disque à chiffrer

- En 1586, le français Blaise de Vigenère élabore un système de substitution poly-alphabétique, connue sous le nom "chiffre de Vigenère".Le procédé consiste à remplacer une lettre par une autre qui n'est pas toujours la même. C'est un système bien plus solide que le code de César car elle nécessite une clé de décryptage. En effet, pour pouvoir chiffrer un texte, nous avons besoin

d'une clé, dont les caractères sont utilisés pour effectuer la substitution. Évidemment, plus la clé sera longue et variée le texte sera mieux chiffré.

1.2.3 Période de la deuxième guerre mondiale et l'utilisation d'enigma

- En 1918, l'allemand Arthur Scherbius fit breveter sa machine à crypter, appelée Enigma. C'est une machine électromécanique portable intégrait d'une méthode de chiffrement par substitution poly-alphabétique. La machine se compose de multiples rotors comportant les 26 lettres de l'alphabet, un dispositif appelé "brouilleur", ainsi que d'un pupitre de connexion qui effectue les conversions mono-alphabétiques. C'est cette combinaison qui forme la clé du chiffrement. Une fois le brouilleur configuré, le texte clair est saisi par l'intermédiaire d'un clavier, puis passé à travers le brouilleur pour enfin apparaître crypté sur un tableau lumineux. Pour chaque lettre saisie sur le clavier, le brouilleur tourne d'un cran, changeant ainsi la clé de cryptage à chaque nouvelle frappe. Elle utilise les mêmes clés au chiffrement et au déchiffrement, ce qui facilite les deux processus.
- L'ingénieur américain Philip Johnston eut l'idée d'utiliser la langue navajo comme procédé cryptographique. La méconnaissance quasi totale de cette langue ainsi que sa construction grammaticale très particulière, la rendant impénétrable aux étrangers, décidèrent de son utilisation, lors de la campagne du pacifique pendant la seconde guerre mondiale.

1.2.4 Utilisation moderne de la cryptographie

- Sécurisation des réseaux Informatiques
- Cryptage des communications militaires
- Signature électronique
- Commerce électronique

1.3 Définition de la cryptographie

Le mot cryptographie vient des deux mots grecs kryptós "secret" et gráphein "écrire", C'est-à-dire écrire secrètement.

Le mot cryptographie est un terme générique désignant l'ensemble des méthodes utilisées pour cacher l'information, c'est-à-dire la rendant incompréhensible sans aucun secret.

La cryptographie est l'art de cacher une information pour la rendre inintelligible [2][3]à toute personne ne connaissant pas un certain secret. Autrement dit, est l'ensemble des processus de verrouillage visant à protéger l'accès à certaines données afin de les rendre incompréhensible aux personnes non autorisées [4].

La cryptographie est la science du secret, en premier lieu elle a été réservée uniquement aux relations diplomatique ou militaire, ensuite elle s'est généralisée. Son objectif est l'étude des méthodes permettant d'assurer les services d'intégrité, d'authentification, de confidentialité, et du non répudiation dans les systèmes d'information et de communication.

- **Confidentialité :** C'est la propriété qui assure que seuls les utilisateurs autorisés, dans des conditions prédéfinie ont accès aux informations sensibles. C'est-à-dire garder les informations secrètes sauf pour les personnes auxquels elles sont destinées. C'est l'un des moyens pour garantir la confidentialité des données; c'est la cryptographie où bien le chiffrement des données.
- **Authentification :** C'est la propriété qui assure la vérification et la confirmation de l'identité des entités qui s'échangent des informations, c'est-à-dire de garantir à chacun des correspondants que son partenaire est bien celui qu'il croit être. L'un des moyens pour garantir l'authentification est l'utilisation des signatures numériques.
- **Intégrité :** C'est la propriété qui assure que les données ne sont pas corrompues ni modifiées de façon non autorisée. L'un des moyens pour garantir l'intégrité est l'utilisation des empreintes digitales.
- **Non répudiation :** C'est la propriété qui assure que l'auteur d'un acte ne peut ensuite nier l'avoir fait, il en assume la responsabilité.

1.4 Terminologie [5][6]

Emetteur : C'est la personne qui possède la clé de chiffrement pour chiffrer un **message** et l'envoie au récepteur.

Récepteur : C'est la personne qui possède la clé de déchiffrement pour déchiffrer un message reçu de l'émetteur.

Alphabet : Ensemble fini de symboles utilisés pour écrire les messages "textes en clair".

Texte en clair : C'est une suite de caractères écrite à partir de l'**alphabet** pour former un message à chiffrer.

Texte chiffré : Appelé cryptogramme, c'est une suite de caractères incompréhensible, autrement dit, c'est le résultat du chiffrement du texte en clair.

Clé : C'est un paramètre utilisé lors de chiffrement et de déchiffrement, on distingue deux types de clé : une clé de chiffrement et une clé de déchiffrement. Si la clé de chiffrement et la clé de déchiffrement sont identiques on parle alors de chiffrement symétrique, Si la clé de chiffrement et la clé de déchiffrement sont différents on parle alors de chiffrement asymétrique.

Chiffrement : C'est l'opération qui permet de transférer un texte en clair à un texte chiffré c'est-à-dire le rendre incompréhensible aux personnes qui ne possèdent pas une clé de déchiffrement.

Déchiffrement : C'est l'opération inverse de chiffrement, elle permet de retrouver le texte en clair à partir du texte chiffré à l'aide d'une clé de déchiffrement c'est-à-dire le texte sera compréhensible.

Cryptologie : C'est une science mathématique qui étudié le secret. Elle est composée de deux branches : la cryptographie et la cryptanalyse.

Cryptographie : C'est la branche qui regroupe les méthodes de chiffrement et de déchiffrement des textes en clair à fin de les rendre inintelligible aux personnes qui ne possèdent pas la clé de déchiffrement.

Cryptanalyse : Contrairement à la cryptographie la cryptanalyse c'est la branche qui cherche des faiblesses des systèmes pour décrypter les textes chiffrés, elle consiste à retrouver le texte en clair, sans la connaissance de la clé de déchiffrement. Autre mot dit c'est l'art de casser des crypto-système.

Décrypter : c'est l'opération de retrouvé le texte en clair à partir du texte chiffré sans la connaissance de la clé de chiffrement, donc ce mot ne devrait être employé que dans le contexte de la cryptanalyse.

Crypter et cryptage : Ce sont des mots anglicismes dérivés du verbe "to crypt", ce sont souvent employés à la place des mots "chiffrer" et "déchiffrer" qui est faux car ces termes n'ont pas de sens à la cryptographie.

Crypto-système : C'est l'ensemble des méthodes de chiffrement et de déchiffrement, il existe trois algorithmes pour un crypto-système : un algorithme pour la génération de clés, le deuxième pour le chiffrement, et le troisième pour le déchiffrement.

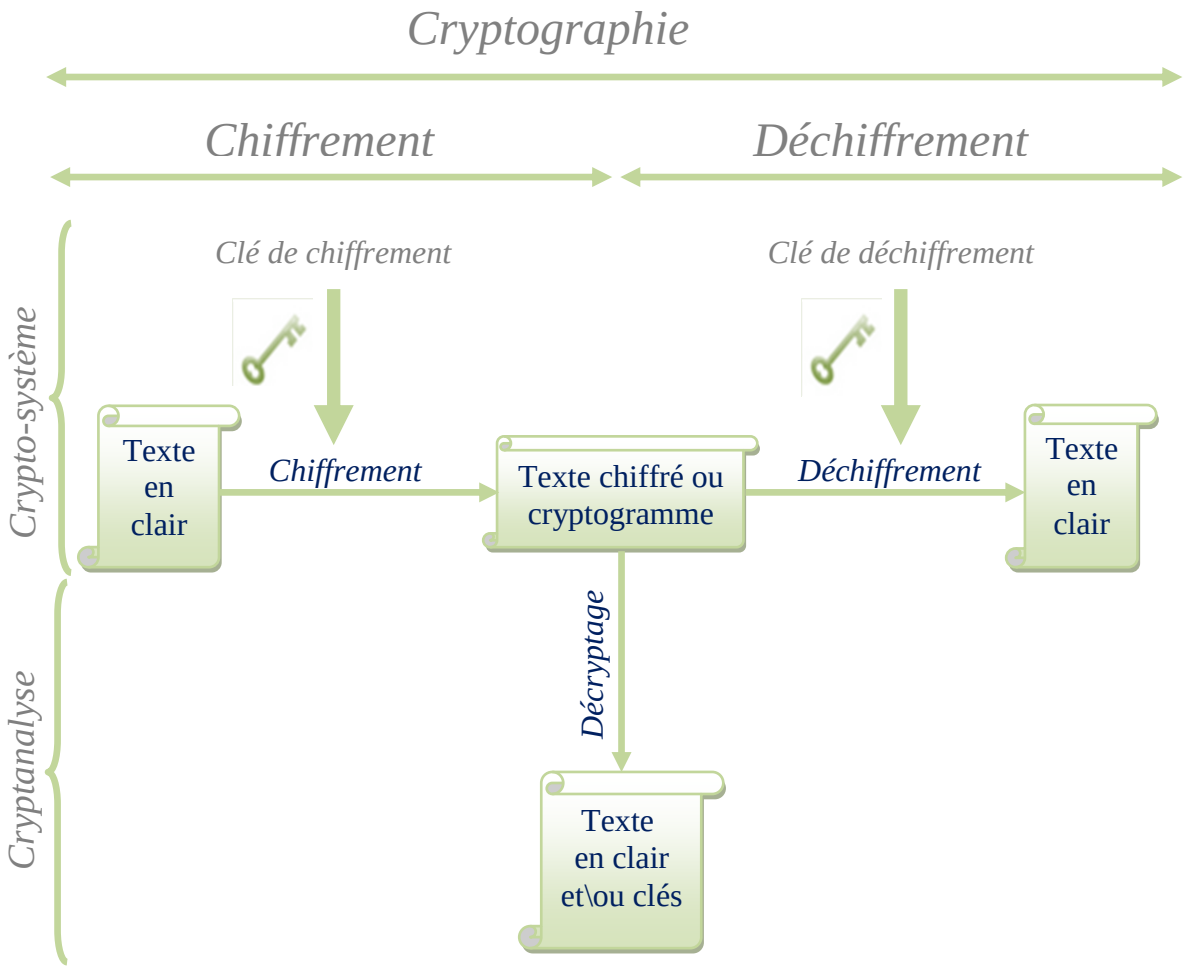


Figure.1.4 Schéma explicatif de différentes terminologies de la cryptographie

1.5 Algorithme de la cryptographie

Pour parler secrètement, c'est-à-dire parler confidentiellement on doit chiffrer les textes échangés. Il existe deux grandes familles de chiffrement [7] à base de clés : le chiffrement symétrique "à clé secrète", et le chiffrement asymétrique "à clé publique".

1.5.1 Chiffrement symétrique

On dit qu'un chiffrement est symétrique si la clé de chiffrement et la clé de déchiffrement sont identiques (on utilise la même clé lors de chiffrement et de déchiffrement).La clé de chiffrement peut être calculée à partir de la clé de déchiffrement et vice versa. Donc il faut que les communicants se mettent d'accord au préalable sur une clé, et cette clé doit être gardée secrète car la sécurité de la communication repose sur elle.[8][9].

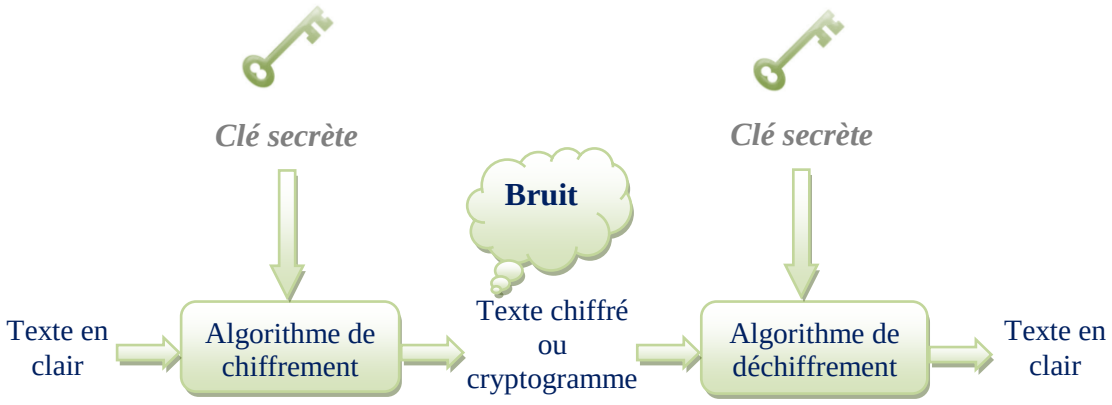


Figure.1.5 Schéma explicatif de chiffrement symétrique

On distingue deux types de chiffrement dans cette famille : le chiffrement par blocs et le chiffrement par flot.

- **Chiffrement par blocs :** traite le message en clair par groupes de bits appelés bloc, chaque bloc est chiffré l'un après l'autre.
- **Chiffrement par flot :** appelé aussi chiffrement continu, traite l'information bit à bit.

Quelque exemple de systèmes cryptographiques qui utilise le chiffrement symétrique : DES (***D**ata **E**ncryption**S**tandard*) [10], 3DES (***Triple-d**ata*

E**ncryption**S**tandard*), RC4 (R**ivest**C**ipher 4*)[11], RC5, AES [12](***A**dvanced**E**ncryption**S**tandard*), chiffre césar.

1.5.2 Chiffrement asymétrique

Le chiffrement asymétrique utilise deux clés une clé pour le chiffrement et une deuxième clé différente pour le déchiffrement. Ce chiffrement est appelé aussi chiffrement à clé publique. La clé secrète ne peut pas être déduite facilement à partir de la clé publique, donc il faut garder la clé privée secrètement mais la clé publique on peut la diffusé même sur des canaux pas sûr [8] [13].

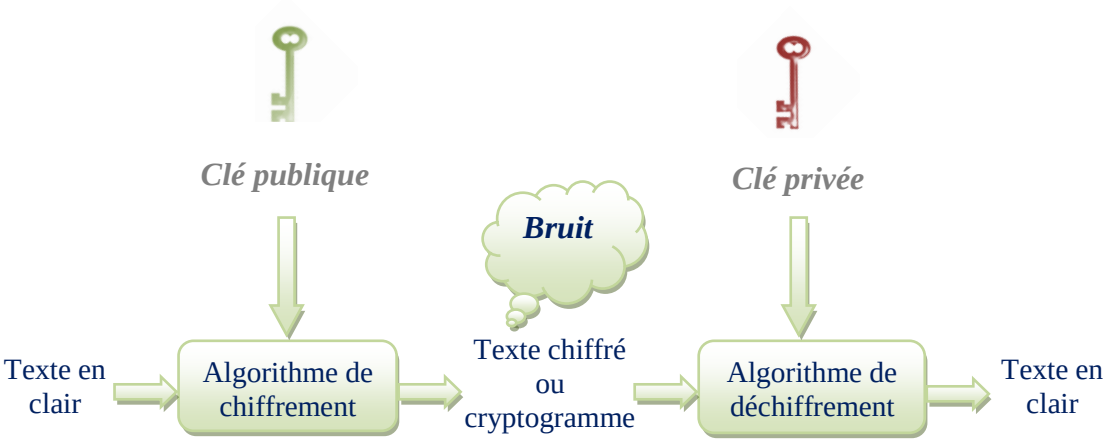


Figure.1.6 Schéma explicatif de chiffrement asymétrique

Ce concept de chiffrement a été inventé par Whitfield Diffie et Martin Hellman en 1976[14]. Elle a pour but de résoudre le problème posé dans le chiffrement symétrique ce qui concerne la distribution de la clé de chiffrement.

Quelque exemple de systèmes cryptographiques qui utilise le chiffrement asymétrique : RSA[15]: (***R**ivest**S**hamir **A**dleman*) c'est un algorithme utilisé pour chiffrer les données ou pour les signé, Diffie-Hellman[14] : c'est un protocole d'échange des clés. **DSA**[16] :(***D**igital **S**ignature **A**lgorithm*) c'est un algorithme de signature. ElGamal[17] : c'est un algorithme utilisé à la fois pour le chiffrement et pour la signature.

1.6 Chiffrement basé sur le chaos

Le plus simple système cryptographique est celui du chiffrement utilisant opération mathématique XOR (ou-exclusif) qui consiste à xorer le message en clair avec une suite binaire pseudo-aléatoire construite à partir d'un registre à décalage par exemple. L'idée du chiffrement basé sur le chaos, est de remplacer cette suite binaire par une discrétisation numérique d'une fonction (générateur) chaotique dont ses caractéristiques permettant d'attendre des performances supérieures. C'est-à-dire le chiffrement basé sur le chaos qui consiste à mélanger le texte en clair avec une suite généré par un système chaotique pour former le texte chiffré qui va transmettre à travers un canal non sécurisé. Lors de réception de ce texte chiffré un procédé inverse est appliqué pour retrouver le texte en clair original, le récepteur doit utiliser le même système chaotique pour former la même suite utilisé lors de chiffrement.

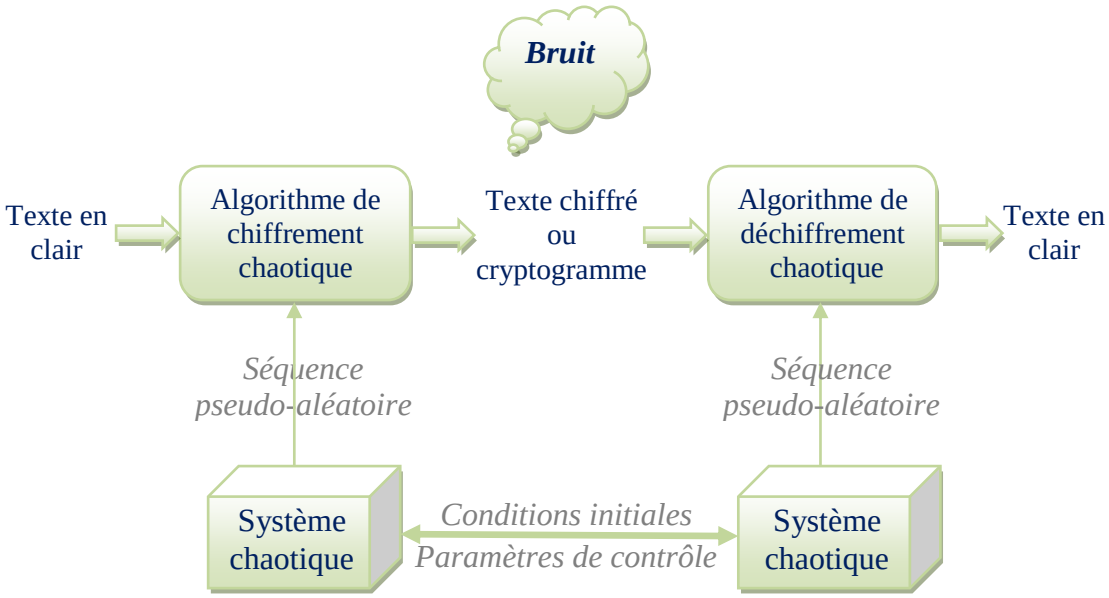


Figure.1.7 Schéma explicatif de chiffrement basé sur le chaos

On dit un système est chaotique s'il est non-linéaire, imprévisible et sensible aux conditions initiales, c'est-à-dire une simple modification dans l'état initial provoque une divergence exponentielle aux états futurs. La clé secrète de ces systèmes est formée de conditions initiales et de paramètres d'un générateur chaotique.

1.7 Conclusion

Les besoins de sécurité restent toujours en augmentation, nécessitant des systèmes robustes pour la protection de l'information. Dans ce chapitre nous avons présenté l'un des moyens de sécuriser des données par la cryptographie. Tous d'abord nous avons introduit une brève historique, des définitions et des concepts de base "terminologie". Puis nous avons vu les deux grandes classes de la cryptographie, le chiffrement symétrique dit aussi à clé secrète et le chiffrement asymétrique dit aussi à clé publique. Enfin nous avons parlé sur le chiffrement basé sur le chaos.

CHAPITRE 2
CHAOS

Chapitre 2

Chaos

2.1 Introduction

Depuis longtemps, le chaos était synonyme de désordre et de confusion, cette science était caractérisée par le déterminisme, la prévisibilité et la réversibilité. Il découvrit la notion de sensibilité aux conditions initiales.

Le terme "chaos" définit un état particulier d'un système dont le comportement ne se répète jamais, il est très sensible aux conditions initiales, est imprédictible à long terme. Autrement dit le chaos est défini par un comportement lié à l'instabilité et à la non-linéarité dans des systèmes dynamiques déterministes. La relation entre l'instabilité et la chaoticité est alors que le système manifeste une très haute sensibilité aux changements de conditions.

Nous nous intéresserons dans ce chapitre à quelques systèmes chaotiques afin de les exploiter pour construire de nouveaux crypto-systèmes.

2.2 Chaos

Il n'existe pas de définition rigoureuse du chaos, mais par chaos, il faut admettre la notion de "phénomène imprévisible et erratique" [18] [19] [20]. Cependant, depuis une trentaine d'années, on attribue le terme chaos à des "comportements erratiques qui sont liés à des systèmes simples pouvant être régis par un petit nombre de variables entre lesquelles les relations décrivant leur évolution peuvent être écrites. Ces systèmes sont donc déterministes bien qu'imprévisibles". Divers auteurs précisent que le chaos est "un comportement effectivement imprévisible à long terme survenant dans un système dynamique à cause d'une sensibilité aux conditions initiales (S.C.I)".

2.2.1 Sensibilité aux conditions initiales

La sensibilité aux conditions initiales (S.C.I) est une caractéristique fondamentale des systèmes dynamiques. Il faut entendre ici qu'un système réagira de façon totalement différente selon la condition initiale. Ceci a notamment comme conséquence le fait

qu'un système chaotique, même si toutes ses composantes sont déterminées, et totalement imprévisible car sensible à d'infimes perturbations initiales.

2.2.2Attracteur

L'attracteur est une limite vers laquelle semblent convergé les orbites du système. On peut définir un attracteur comme un ensemble compact de l'espace d'état vers lequel toutes les trajectoires environnantes convergent, c'est à dire que l'attracteur décrit en fait une situation de régime telle qu'elle peut apparaître après disparition des phénomènes transitoires. Les attracteurs étranges constituent ce que l'on appelle le chaos. Les attracteurs étranges sont également appelés attracteurs chaotiques; un système chaotique.

Enfin, les trajectoires dans l'attracteur étranges ne doivent pas se couper. Cette dernière propriété est très intéressante et sera exploitée dans le cadre de la cryptographie.

Parmi les attracteurs les plus connus dans le cas discret on trouve la carte logistique [19] et l'attracteur de Hénon [21], sur l'étude de la dynamique des étoiles se déplaçant dans des galaxies. Hénon découvrit que l'attracteur étrange des orbites stellaires est de la forme d'une banane (figure.2.1).

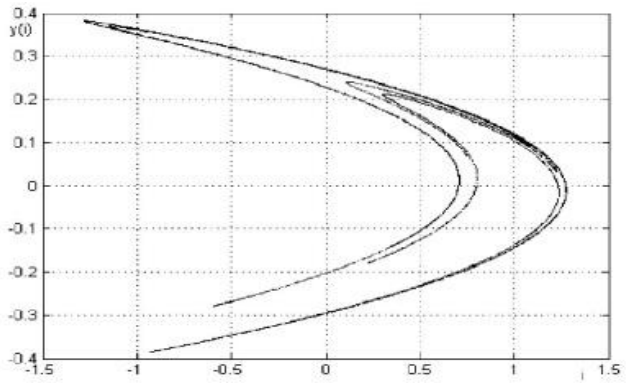


Figure.2.1 Attracteur de Hénon

2.3 Carte logistique et ses variantes

La carte logistique [18] [22] [23] est l'une des dynamiques très connues dans la théorie des systèmes non-linéaires et qui est définit par l'équation suivante :

$$y_{k+1} = r.x_k(1 - x_k)$$

Elle nous donne une explication parfaite pour un comportement d'un système dynamique. Ce système a été développé par le Pr. Pierre François Verhulst (1845) pour mesurer l'évolution de population dans un environnement limité, utilisé plus tard en 1976 par le biologiste Robert May pour l'étude d'évolution de population des insectes ou :

y_{k+1} : La génération à la venir qui est proportionnel a x_k.

x_k : La génération précédente.

r : Constante positive incorpore tous les facteurs reliés au reproductif, succès à la survie hivernale des œufs par exemple, etc.

Afin d'étudier ce système dynamique ainsi que certains modes asymptotiques particuliers, la première chose à faire est de tracer le graphe parabolique $y = r.x(1 - x)$, et le diagonale $y = x$. L'opération que nous allons suivre pour tracer la forme itérative y_{k+1} en fonction x_k se résume simplement comme suite :

À partir d'une valeur initiale x_0 de l'axe des abscisses, nous rejoignons la fonction par une verticale; la fonction prend la valeur $y_1 = r.x_0(1 - x_0)$,

Par l'horizontale $y_1 = r.x_0(1 - x_0)$ issue du point précédent, nous rejoignons la droite $y = x$,

Nous représentons l'abscisse de ce point d'intersection par la droite verticale $x = x_0$, nous avons bien $y_1 = x_1$.

A partir de la valeur x_1 de l'axe des abscisses, nous rejoignons la fonction par une verticale, la fonction prend la valeur $y_2 = r.x_1(1 - x_1)$, et ainsi de suite.

On prenant r = 3.9 et, $x_0 = 0.01$ pour la carte logistique, les opérations précédentes se retrouvent pour 100 itérations sur les graphiques des figures (figure.2.2.1, figure.2.2.2, figure.2.2.3).

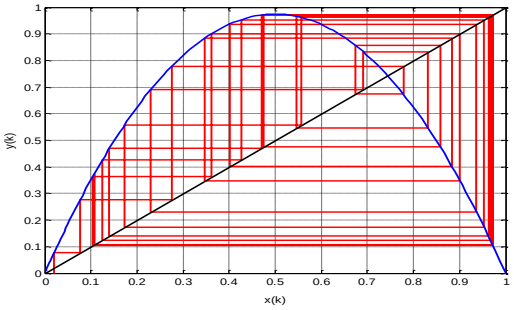


Figure.2.2.1 Évolution de y_k en fonction de x_k

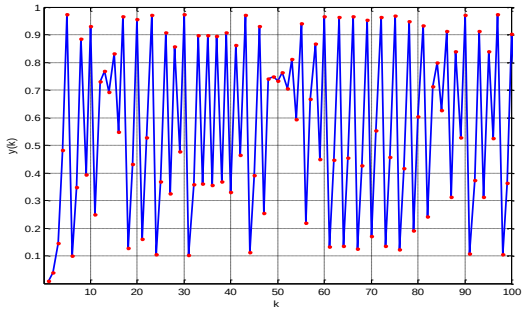


Figure.2.2.2 Régime chaotique en fonction k

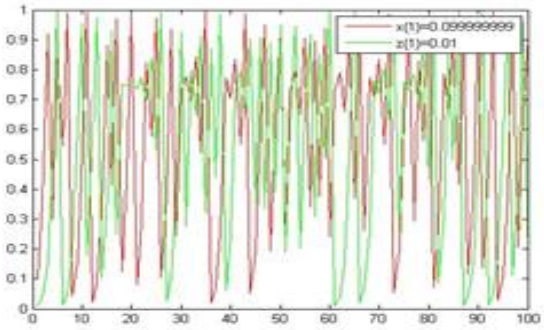


Figure.2.2.3 Sensibilité aux conditions initiales

2.4 Utilisation de la carte logistique dans la cryptographie

Le chiffrement par flot consiste à utiliser un **XOR** entre les données en clair et une suite aléatoire. En cryptographie asymétrique, il est nécessaire de produire de grands nombres aléatoires avec des contraintes supplémentaires (premier, premier entre eux, etc.). De plus, un texte chiffré doit s'approcher le plus possible d'un fichier au contenu aléatoire pour limiter les fuites d'information.

La carte logistique est souvent employée pour générer des données pseudo-aléatoires qui sont employées sur des ordinateurs, dans diverses tâches comme la méthode de Monte-Carlo, la simulation ou les applications cryptographiques.

Au cours des dernières années, il y a eu une explosion de recherche universitaire dans le domaine de la cryptographie en utilisant la carte logistique [24] [25] [26] [27] [28] [29] [30] [31] [32] [33].

Dans ce qui suit nous vous proposons quelques variantes de la carte logistique susceptibles pour nous donner une meilleure appréciation si nous l'utilisons en cryptographie, mais tout d'abord on doit définir comment on génère les valeurs pseudo-aléatoires de ces cartes chaotiques et comment on teste leur aléas.

2.4.1 Génération des nombres aléatoires

Un générateur de nombres aléatoires, ***R**andom **N**umber **G**enerator* (**RNG**) en anglais, est un dispositif capable de produire une séquence de nombres dont on ne peut pas « facilement » tirer des propriétés déterministes [34], de façon que cette séquence puisse être appelée « suite de nombres aléatoires ».

La nécessité d'obtenir des données aléatoires est présente dans bien d'autres domaines. Certains domaines peuvent se contenter de données pseudo-aléatoires et utilisent des générateurs qui s'approchent plus ou moins d'un aléa parfait. On les trouve dans, les jeux de hasard, la simulation, l'analyse, l'échantillonnage, la prise de décision et, la sécurité informatique (cryptographie).

Un générateur pseudo-aléatoire est en général construit à partir d'une fonction mathématique déterministe, qui calcule par récurrence une suite d'états. La valeur initiale de l'état est appelée le germe. Si un ennemi ignore la valeur du germe, il doit lui être impossible en pratique, à partir des premiers termes d'une suite pseudo-aléatoire, de prévoir le terme suivant.

Un cas typique est celui où on dispose d'une fonction f et d'une fonction à sens unique F. À partir du germe s_0 on calcule une suite cachée d'états: $s_k = f(s_k - 1)$, et la suite pseudo-aléatoire $x_k = F(s_k)$. C'est ce principe qui est utilisé pour l'extraction d'une suite de nombres à comportement aléatoire de la carte logistique.

Un système chaotique, même si toutes ses composantes sont déterminées, est totalement imprévisible car sensible à d'infimes perturbations initiales. Cette sensibilité est exploitée pour les algorithmes cryptographiques (figure 2.2.3).

2.4.2 Tests de l'aléa

Produire des nombres aléatoires pose une double difficulté : la production en elle-même bien sûr, mais surtout savoir caractériser le hasard [35]. En effet, le caractère aléatoire est une notion difficile à appréhender, mais des travaux récents ont permis de comprendre comment caractériser une série véritablement aléatoire.

Pour trouver la suite de nombres aléatoires distribuée uniformément dans l'intervalle [0,1[, il faut prendre des :

$u_i = x_i/M$, pour i=1, ..., n (1)

Avec, M la plus grande valeur de x_i, et dans le cas idéal, on doit trouver les trois valeurs théoriques de tests :

Type de test	**Equation**	**valeur**
Moyenne	$\overline{U} = \frac{1}{n}\sum_{i=1}^{n} U_i = \frac{1}{2}$	0.5
Facteur d'autocorrélation	$E(U_i\, U_{i+1}) = \frac{1}{n}\sum_{i=1}^{n-1} U_i U_{i+1} = \frac{1}{4}$	0.25
Variance	$V = \frac{1}{n}\sum_{i=1}^{n} U_i^2 - (\overline{U})^2 = \frac{1}{12}$	0.08

2.4.3 Résultats et interprétations

Au cours des dernières années, il y a eu énormément d'article de recherche sur la justification de l'utilisation de la carte logistique dans le domaine de la cryptographie et la possibilité diversifier des variantes de cette dernière [23] [36] [37] [38].

Une variante connue de la carte logistique, il s'agit de l'équation de récurrence discrète proposée par Marotto [38] :

$$x_{n+1} = r.x_n{}^2(1 - x_n) \qquad (2)$$

$$x_n > 0, \qquad 0 < r < \frac{27}{4}$$

Une autre variante était proposé à l'International Workshop on Chaos-Fractals Theories and Applications par Sun, Y. and G.Y. Wang [37].

$$x_{n+1} = x_n.(4 - \frac{x_n}{\mu}) \qquad (3)$$

$$\mu \neq 0, = 256, 1000$$

Nous proposons dans ce qui suit une nouvelle variante double de la carte logistique :

$$x_n = \alpha.x_n.(1 - \frac{x_n}{\mu}) \qquad (4)$$

Avec α= 4 ou bien, $\alpha = (4 - \frac{1}{\mu})$

Afin d'évaluer l'aléa de la suite de nombres extraite de la carte logistique et de ses différentes variantes, on les tests, en calculant leur moyenne, leur facteur d'auto-corrélation et leur variance, il faut les adapter à l'équation 1. Dans ce cas, et pour une valeur donnée de x_0, on calcule tous les x_i, puis on transforme cette suite pour qu'elle prenne les valeurs de 0 à 255, comme suit :

$$x(n) = \text{mod}(\text{floor}(\text{abs}(x(n))*\text{sacl}), 256) \qquad (5)$$

Scal=10000 (cette valeur peut être changé pour donner un autre champ de la clé de chiffrement), les x(n) sont strictement inférieures à 256, en prend M=256.

2.4.3.1 Fonction de Marotto

L'équation proposée par Marotto est :

$$x_{n+1} = r.x_n{}^2(1 - x_n)$$

On prend N =15000 valeurs.

Valeurs initiaux		Moyenne	Variance	Auto-corrélation
x_0	r			
0.1	6.21	5.0850e-06	2.3086e-07	7.0738e-08
0.5	6.21	0.6678	0.0397	0.4262
0.83	6.21	0.6680	0.0397	0.4265
0.1	6.4	5.3442e-06	2.4827e-07	8.0198e-08
0.5	6.4	0.6637	0.0468	0.4316
0.83	6.4	0.6612	0.0474	0.4284

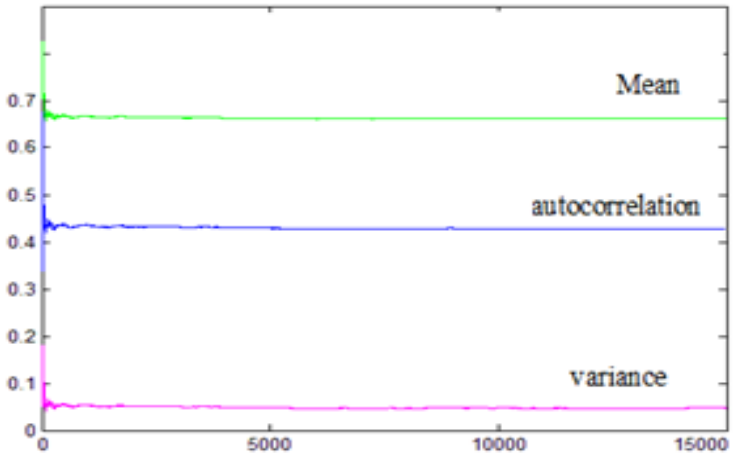


Figure.2.3 Comportement de la fonction de Marotto avec x_0=0.83, et r=6.4.

Cette variante peut être utilisée dans des simulations, mais elle ne satisfait pas le cas théorique des données aléatoires de l'équation 1.

2.4.3.2 Carte logistique

L'évaluation est faite pour l'équation de la carte logistique :

$$x_{n+1} = \lambda.\ x_n.(1 - x_n)$$

Nous prenons comme valeurs initiales :

Scal=10000

N=15000 nombre d'échantillons

$x_0 = 0.1$ et $x_0 = 0.100000000001$

λ=3.998

	x(1)	
	0.1	0.100000000001
Moyenne	0.4951	0.4846
Autocorrélation	0.2483	0.2378
Variance	0.0857	0.0880

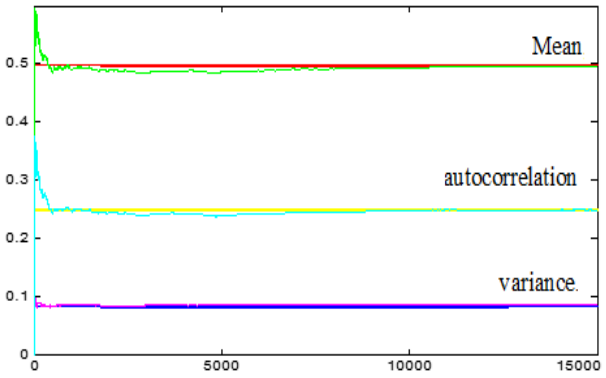


Figure.2.4 Comportement de la carte logistique

Les tests montrent que la génération des données de la carte logistique a un comportement aléatoire.

2.4.3.3 Variante de Sun and Wang

L'équation proposée par Sun et Wang

$$x_{n+1} = x_n.(4 - \frac{x_n}{\mu})$$

On prend

- μ=1000, μ=1000 ; N=15000

	x(1)	
	0.1	0.100000000001
Moyenne	0.4953	0.5031
Autocorrélation	0.2451	0.2527
Variance	0.0829	0.0842

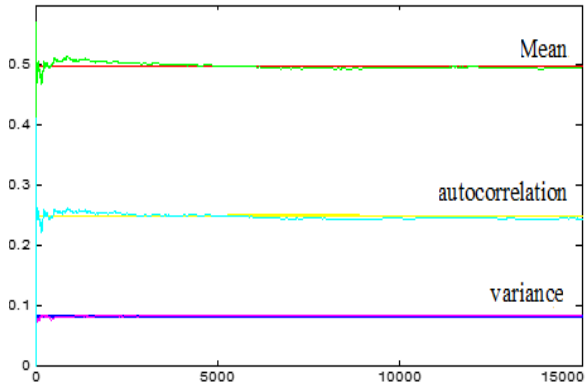


Figure.2.5 Comportement de la variante de Sun et wang.

Les tests montrent que la génération des données de la variante de la carte logistique proposée par Sun et Wang a un comportement aléatoire.

2.4.3.4 Nouvelles variantes de la carte logistique

On essaie de tester les deux variantes proposées :

$$x_n = 4.x_n.(1-\frac{x_n}{\mu})$$

$$x_n = (4-\frac{1}{\mu}).x_n.(1-\frac{x_n}{\mu})$$

On prend comme valeurs initiales :

μ=1000

N=15000

x_0=0.1 et x_0=0.100000000001

$$x_n = 4.x_n.(1-\frac{x_n}{\mu})$$

	x(1)	
	0.1	0.100000000001
Moyenne	**0.4951**	0.4968
Auto-corrélation	**0.2440**	0.2469
Variance	**0.0839**	0.0826

Graphes des tests dans la figure2.6.1.

1) $x_n = (4 - \frac{1}{\mu}).x_n.(1 - \frac{x_n}{\mu})$

	x(1)	
	0.1	0.100000000001
Moyenne	0.4995	0.4949
Autocorrélation	0.2505	0.2455
Variance	0.0832	0.0836

Graphes des tests dans la figure.2.6.2.

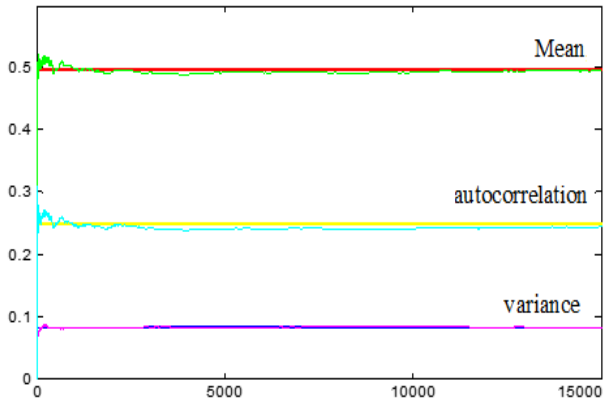


Figure.2.6.1 Comportement de notre première variante

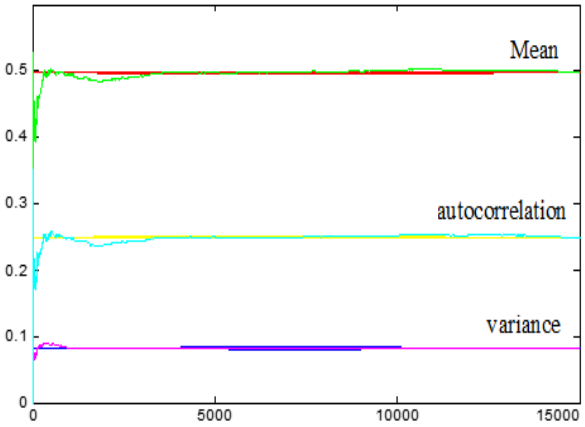


Figure.2.6.2Comportement de notre deuxième variante

On remarque que :

- A chaque fois qu'on augmente μ de 4, 10, 100, 200, 1000 dans les deux équations. Le test de la moyenne calculé s'approche de la moyenne de la série idéale.
- A chaque fois que le nombre de réalisations augmentent le test de la moyenne calculé converge vers 0.5

- A chaque fois qu'on augmente de μ =8, 9, 10, …N=1000 on a une variance proche de 1/12.
- A chaque fois qu'on augmente de μ =100 et 200 le facteur de corrélation est presque1/4.
- Donc pour un μ=1000 et N=15000, la génération des données des variantes de la carte logistique que nous avons proposée satisfait les tests de la moyenne de la variance et du facteur d'auto-corrélation pour des données aléatoires.
- La génération des données avec les deux variantes que nous avons proposées, et celle de Sun et Wang sont équivalentes en efficacité cryptographie avec la génération des données de la carte logistique. Le choix de l'équation génératrice revient à l'utilisateur. Aussi, il est question de rajouter dans le champ de la clé de chiffrement le coefficient d'amplification (Scal) utilisé dans l'équation 5 pour renforcer le cryptage.

2.4.4 Conclusion

Après avoir ajusté les différents paramètres des équations qui donnent la génération des nombres aléatoires. Lorsque *N* « nombre des réalisations » tendent vers une valeur égale à 1000, les valeurs qu'on trouve pour la moyenne, la variance, et le facteur d'auto-corrélation sont à peu près proche aux valeurs théoriques, et lorsqu'on augmente le nombre des réalisations *N* jusqu'à 15000, on trouve des valeurs plus proche pour la moyenne, la variance, et le facteur d'auto-corrélation aux valeurs théoriques :

Donc pour avoir une meilleure appréciation aléatoire il faut augmenter le nombre des réalisations *N*.

La génération de nombre aléatoire construit à partir d'une fonction mathématique déterministe, produit une séquence de nombres dont on ne peut pas "facilement" tirer des propriétés déterministes. Mais, elle peut toutefois souffrir (et souffre généralement) de biais. Actuellement, les meilleures méthodes, censées produire des suites véritablement aléatoires, sont des méthodes physiques qui profitent du caractère aléatoire des phénomènes quantiques.

2.5 L'attracteur Hénon

L'attracteur chaotique de Hénon du nom de l'astronome français Michel Hénon a été présenté pour la première fois en 1976 [21] dépendant de deux paramètres a et b, et est basé sur les équations suivantes :

$$X_{n+1} = (Y_n + 1) - (a * X_n^2) \qquad (6)$$
$$Y_{n+1} = b * X_n$$

Les conditions initiales sont $(x_0 = 1, y_0 = 1)$avec a = 1.4 et b = 0.3, ces deux valeurs montre le comportement chaotique de l'attracteur de Hénon.

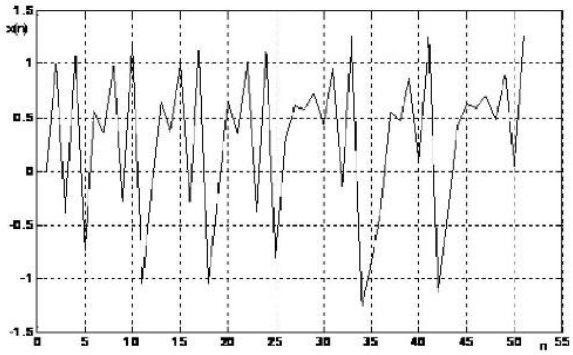


Figure.2.7.1 l'attracteur par rapport à x

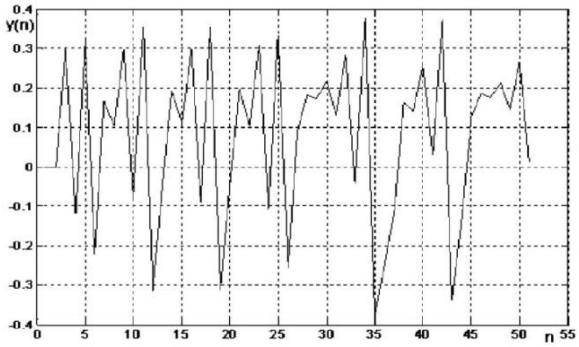


Figure.2.7.2 l'attracteur par rapport à y

2.6 La carte PWLCM

La carte PWLCM (Piecewise Linear Chaotic Maps) est une autre carte chaotique linéaire par morceaux [39] [40], décrite par l'équation suivante :

PWLCM: $(0,1) \rightarrow (0,1)$

$$x_{n+1}=\text{PWLCM}(p,x_n)=\begin{cases}\frac{x_n}{p}, & 0\le x_n<p\\ \frac{x_n-p}{0.5-p}, & p\le x_n<0.5\\ \text{PWLCM}(1-x_n,\ p), & 0.5\le x_n<1\end{cases}$$

Où : $p \in [0,1]$: C'est le paramètre du contrôle.

$x_n \in [0,1]$: C'est la valeur initiale.

Grâce à ses bonnes caractéristiques cryptographiques, comparées à celles de la carte Logistique, la carte PWLCM est très utilisée dans le chiffrement des données.

2.7 Utilisations des cartes chaotiques

Lors de notre recherche pour trouver la meilleure façon d'utiliser le chaos dans nos simulations, nous avons pu développer quelques résultats intéressants, que nous allons présenter sous forme de contributions.

2.7.1 Contribution 1 : Concaténation de cartes chaotiques pour le chiffrement des images

La sécurisation de l'information est aujourd'hui, essentiellement fondée sur les algorithmes de calcul dont la confidentialité dépend du nombre de bits nécessaires à la définition d'une clé cryptographique. Si ce type de système a fait ses preuves, la puissance croissante des moyens de calcul menace la confidentialité de ces méthodes cryptographiques classiques [8]. Les ordinateurs puissants sont certes capables de chiffrer et de déchiffrer rapidement l'information, mais leur vitesse de calcul autorise parallèlement la cryptanalyse, qui a pour objectif de "casser" un code en découvrant la clé, par exemple en testant toutes les clés possibles.

Les recherches récentes sur les algorithmes de chiffrage d'image ont été de plus en plus basées sur des systèmes chaotiques que l'on peut trouver par exemple: R. Ranjith Kumar and M. Bala Kumar (2014)[41], Long Bao, Yicong Zhou, C. L. Philip Chen, Hongli Liu (2012) [42], G.A.Sathishkumar, .K.Bhoopathybagan and N.Sriraam (2011) [43], Mintu Philip, AshaDas (2011) [44], K. Sakthidasan@Sankaran and B. V. Santhosh Krishna (2011) [45], Noura, H. El Assad, S. Vladeanu (2010) [46], Ai-hongZhu, Lia Li (2010) [47], Shubo Liu, Jing Sun, Zhengquan Xu (2009) [48], Xiping He Qionghua Zhang (2008) [49], Xin Zhang, Weibin Chen (2008) [50].

Dans cette contribution, nous proposons un nouveau crypto-système à clé secrète sur la base de la manipulation spécifique des cartes chaotiques. Nous associons à l'attracteur de Hénon, la carte logistique, pour la construction de ce crypto-système. Nous générons des valeurs par le biais de la carte logistique qui sera ajouté aux pixels de l'image en clair. Ce résultat modulo 256, sera permuté à une autre position de l'image cryptée. Le calcul de cette permutation est déduit de l'attracteur de Hénon, qui est à 2 dimensions.

2.7.1.1 Crypto-système proposé

La construction du nouveau crypto-système à clé secrète ce fait en deux étapes jumelées : A partir de la concaténation des cartes chaotiques, on réalise une permutation de l'image en clair plus, une production d'une matrice masque VAL de même dimension que celle de l'image en clair. On XOR ce masque avec l'image originale permutée pour trouver l'image chiffrée.

On propose l'organigramme de la figure 2.8. Avec les conditions initiales de l'attracteur de Hénon on prend (x_0=0.1 et y_0=0.1), et a=1.4 et b=0.3 et, pour la carte logistique selon l'équation z_n, on prend z_0=0.01, r=3.9.

Les valeurs de départ N1 | N2 respectivement pour l'attracteur de Hénon et la carte logistique sont 17 et 53.

Pour k = 1 à 256
Pour h = 1 à 256
Ligne i= mod(fix (x_n*10^7), 256)
Colonne j= mod(fix (y_n*10^7), 256)
VAL(i,j)= mod(fix(z_n*10^7), 256)
C(i, j)= M(k, h) + VAL(i,j)
End, end.

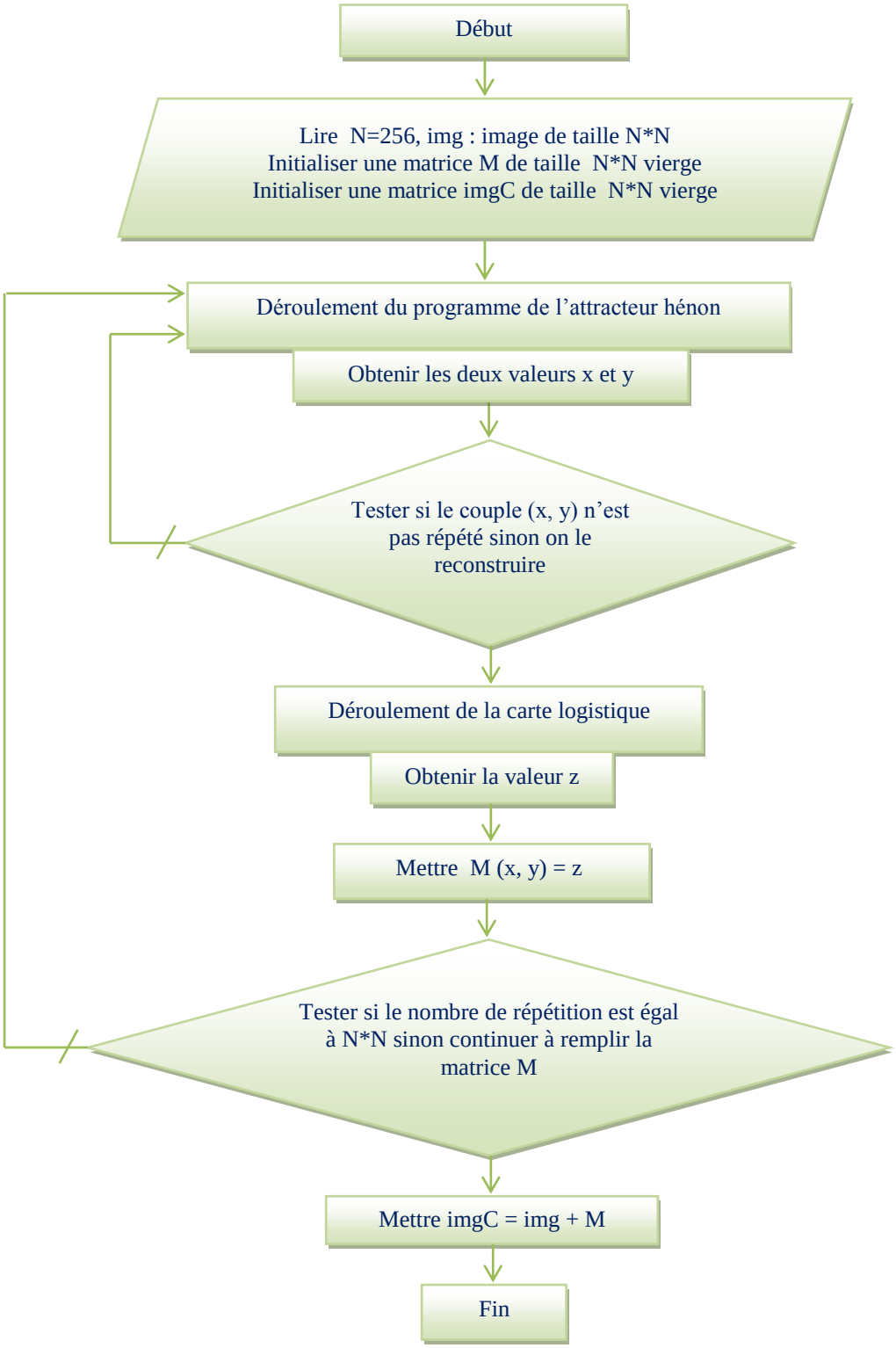


Figure.2.8 Organigramme du crypto-système proposé

Encryption: C= M_pe +VAL mod (256)

Avec C : matrice chiffrée,

VAL : matrice du Masque,

M_per : matrice de l'image en clair permutée au rythme de la matrice VAL.

La taille de la clé de chiffrement de l'espace est le nombre total de différentes valeurs qui peuvent être utilisés dans le procédé de chiffrement. Dans l'algorithme proposé, le champ de clé secrète est fixé comme suit :

$$ST = \{x_0, y_0, z_0, N1, N2\}.$$

Où x_0, y_0, z_0, sont des nombres à doubles précision. N1, N2 sont des constantes entières. Si la précision de calcul de x_0, y_0, z_0, est 10^{-16}, et $N1 \mid N2 \in [1, 1000]$.

Par conséquent, l'espace de clé est plus grand que 10^{16} x 10^{16} x 10^{16} x 1000 x 1000, (avec $10^3 \approx 2^{10}$) dans ce cas on aura un champ de clé de l'ordre de 2^{180}est c'est énorme.

Donc, l'algorithme de chiffrement a un très grand espace de clé pour résister à toutes sortes d'attaques par force brute.

2.7.1.2 Résultats et interprétations

On travaille avec des images BMP,
L'application a été créée depuis un PC :
HP 6830.
Mémoire : 2.00 Go,
Processeur : intel® Core™ 2Duo CPU T5870 @ 2.00GHZ 2.00GHz
Système d'exploitation : Windows 7 édition intégrale 32 bits,
Carte Graphique : ATI Mobility Radeon HD 6370.

On prend comme clé de chiffrement les valeurs fixes : a = 1.4, b = 0.3, r = 3.9. Les nouvelles valeurs variables qui constituent la clé de chiffrement sont : x_0=0.95, y_0=0.56, z_0= 0.02, avec les valeurs de départ N1 | N2 respectivement pour l'attracteur de Hénon et la carte logistique sont 19 et 19. On regarde les performances du crypto-système proposé.

2.7.1.2.1 Histogramme des images

Pour une image monochrome, c'est-à-dire à une seule composante, l'histogramme est défini comme une fonction discrète qui associe à chaque valeur d'intensité le nombre de pixels prenant cette valeur. La détermination de l'histogramme est donc réalisée en comptant le nombre de pixel pour chaque intensité de l'image. L'histogramme peut alors être vu comme une densité de probabilité [51].

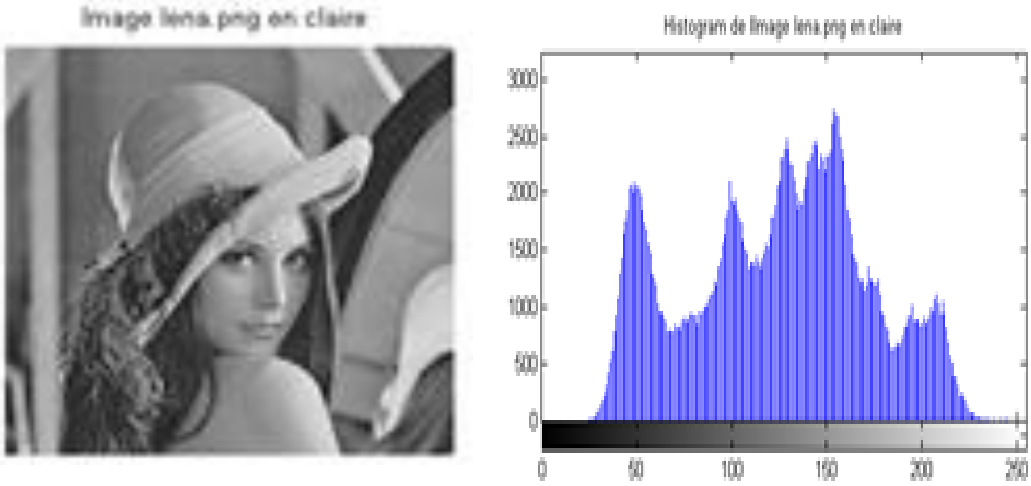
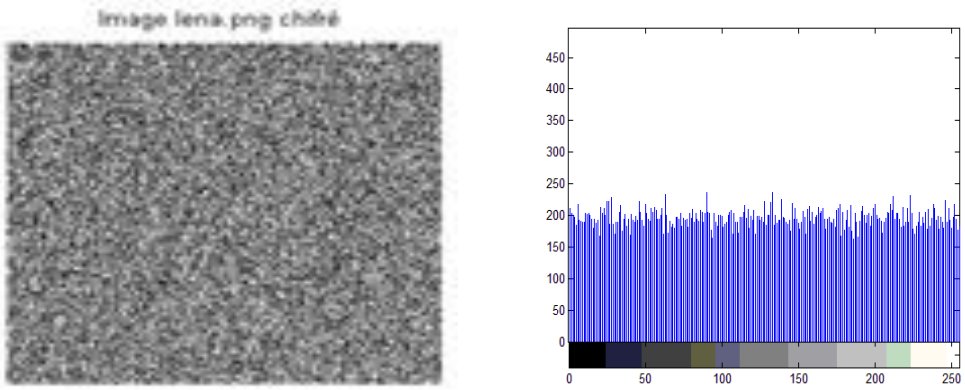


Figure.2.9.1 Image lena.bmp en claire et son histogramme



Figur.2.9.2 Image lena.bmp chiffré et son histogramme

Se référant aux résultats obtenus, nous pouvons clairement voir que l'image en claire diffère sensiblement de celui correspondant cryptée. Par ailleurs, l'histogramme de l'image cryptée est assez uniforme ce qui rend difficile d'extraire les pixels nature statistique de l'image en claire.

Les histogrammes des images claire et chiffrée de Lena montrant ainsi que le crypto-système proposé fonctionne de façon correcte.

On constate que :

- Le chiffrage change la fréquence des pixels avec une distribution équiprobable pour toute l'image
- Les pixels sont très corrélés dans l'image en claire et que le chiffrage annule toute corrélation entre eux dans l'image chiffrée.
- L'image après chiffrage est devenue parasité et ne contient aucune information visible qui se voit sur l'histogramme des deux images qui ne contient aucune information sur l'image en claire.

2.7.1.2.2 Corrélation entre deux pixels adjacents

Pour tester la corrélation entre deux pixels adjacents horizontalement, verticalement et diagonalement de l'image on calcule le coefficient de corrélation pour une séquence de pixels adjacents en utilisant les équations suivantes [51] :

$$E(x) = \frac{1}{N}\sum_{i=0}^{N-1} x_i$$

$$D(x) = \frac{1}{N}\sum_{i=0}^{N-1} \left(x_i - E(x)\right)^2$$

$$cov(x,y) = \frac{1}{N}\sum_{i=0}^{N-1} \left(x_i - E(x)\right)(y_i - E(y)$$

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

Ou x et y sont les pixels adjacents.

On va étudier la distribution de 1000 pixels adjacents qui sont choisis aléatoirement dans l'image en claire et chiffré grâce à la fonction « randsrc ». La fonction « randsrc » randsrc(m,n,[symboles]) génère une matrice de taille MxN de symboles équiprobables, et on obtient une distribution des pixels adjacents.

Picture	Coefficient de Corrélation de l'image en claire	Coefficient de corrélation de l'image chiffrée
lena.bmp	0.9311	0.0117
cameraman.bmp	0.9603	-0.0544
peppers.bmp	0.9525	0.0124
coins.bmp	0.9673	0.0333
football.bmp	0.9857	-0.0237
rice.bmp	0.9813	-0.0470
mandrill.bmp	0.8130	0.0045
house.bmp	0.9832	-0.0485
clown.bmp	0.9802	-0.0232
barbara.bmp	0.9218	0.0409
boat.bmp	0.8834	0.0101

Tableau.2.1 Comparaison des coefficients de corrélation entre les images en clair et chiffrée

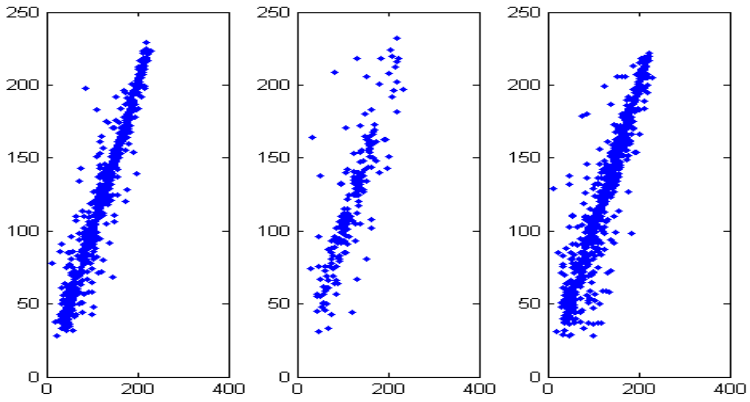


Figure.2.10.1 Distribution des pixels adjacents de l'image lena.bmp en claire

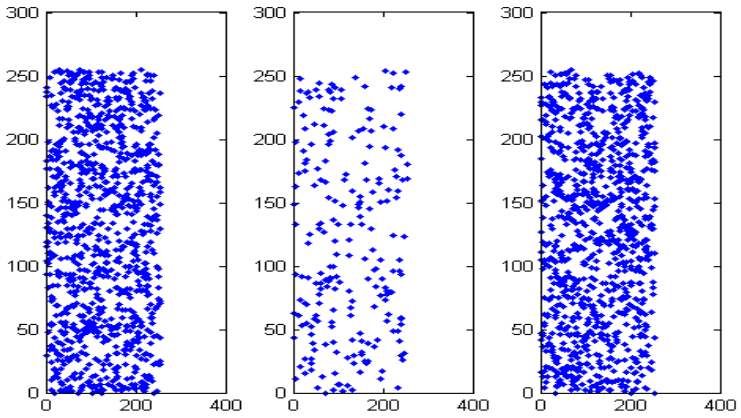


Figure2.10.2 Distribution des pixels adjacents de l'image lena.bmp chiffrée

Les figures (figure.2.10.1 et figure.2.10.2) qui sont en nombre de trois de gauche à droite, comme suit : Corrélation des pixels horizontaux, Corrélation des pixels verticaux, Corrélation des pixels diagonaux.

On constate d'après le tableau 2.8 et les figure.2.10.1 et figure.2.10.2 que :

- Les pixels adjacents sont très corrélés pour les images chiffrées, donc le cryptage créé un désordre très important.
- Les coefficients d'auto-corrélation sont proches de 1 pour les images en claire et ils s'annulent pour les images chiffrées qui prouve le bon fonctionnement de notre système.

2.7.1.2.3 Calcul de l'entropie

La quantité d'information moyenne [51] associée à chaque symbole de la source sans mémoire est définie comme l'espérance mathématique (notée E{.}) de l'information propre fournie par l'observation de chacun des symboles possibles $\{S_1,\ldots,S_n\}$:

$$H(s) = -\sum_{i=0}^{n} p_i \log_2(p_i)$$

Il s'agit de l'information qu'on obtiendrait en moyenne en observant en parallèle les symboles émis par de très grand nombre de source sans mémoire identiques. Comme la source est stationnaire et sans mémoire, il s'agit également de

l'information moyenne par symbole, qu'on obtiendrait en observant une suite de symbole très longue émise par une seule source.

Picture	Entropie de l'image en claire	Entropie de l'image chiffrée
lena.bmp	7.4651	7.9965
Cameraman.bmp	7.0917	7.9970
peppers.bmp	7.5854	7.9966
coins.bmp	6.3106	7.9971
football.bmp	7.1375	7.9967
rice.bmp	7.0279	7.9969
mandrill.bmp	7.2990	7.9966
house.bmp	6.3877	7.9966
Clown.bmp	7.4009	7.9974
barbara.bmp	7.4189	7.9962
boat.bmp	7.1890	7.9957

Tableau.2.2 Comparaison des Entropies entre les images en claire et chiffrée

On constate que l'entropie des images augmente jusqu'à presque atteindre 8 ce qui prouve que le cryptage crée un grand niveau de désordre.

2.7.1.3 Conclusion

Dans cette contribution, nous avons présenté un nouveau système de chiffrement à clé secrète des images.

La construction de ce crypto-système a été faite en deux étapes jumelées. L'association de deux cartes chaotiques, l'attracteur Hénnon et la carte logistique. Par l'attracteur Hénnon qui est de dimension 2, on produit d'une part une permutation de l'image en clair, et d'autre part une matrice masque de dimension égale à celle de l'image originale, où on prend comme valeur des éléments de cette matrice, une des valeurs de la carte logistique. On XOR ce masque avec l'image originale permutée pour trouver l'image chiffrée.

Pour tester l'efficacité de ce crypto-système, on a calculé les trois indicateurs qui permettent d'estimer cet algorithme, à savoir :

1. Histogramme des Images,
2. Corrélation entre deux pixels adjacents,
3. Entropie.

Tous les trois tests ont donné une bonne appréciation de ce crypto-système. De plus, ce crypto-système possède un très grand espace de clé (180 bits) pour résister à toutes sortes d'attaques par force brute.

Ce crypto-système est rapide et simple a implémenté et ne consomme que peu de ressource.

Complexité de calcul de notre algorithme proposé : On peut dire que notre crypto-système a vérifié les deux propriétés établies par Claude Shannon, à savoir la diffusion d'une part et la confusion d'autre part:

1. L'idée de confusion est de mettre en relation le texte en clair et la clé de chiffrement, afin que le texte chiffré soit aussi difficile que possible à établir. Principe souvent présenté comme atteinte d'une substitution pour masquer toute relation linéaire entre le texte chiffré et le texte en clair.
2. Dans notre crypto-système (partie de substitution), nous utilisons une substitution poly-alphabétique du type Vigener, où la clé de chiffrement Vigener est produite par la carte wlogistique (valeurs aléatoires) et dont la taille est identique à celle du texte en clair à chiffrer, autrement dit, elle remplace le type du masque jetable (OTP). (OTP) est une technique de chiffrement qui ne peut pas être craquée, mais nécessite l'utilisation d'une clé pré-partagée unique de la même taille ou plus longue que le message envoyé. Dans cette technique, un texte en clair est oxydé avec une clé secrète aléatoire. C'est donc un cryptage idéal.
3. L'idée de la diffusion est de s'assurer que chaque symbole en clair est "diffusé" dans tout le texte crypté (même une modification mineure du texte en clair doit entraîner une modification très significative du texte crypté). Le principe souvent présenté comme atteinte d'un chiffrement par permutation

pour masquer la redondance en répartissant l'influence d'un bit de clé sur tout le texte chiffré.

Dans notre crypto-système (partie permutation), nous utilisons une simple permutation où chaque pixel est échangé dans une autre position. Pour une taille d'image BMP 256x256 = 65536 pixels, le premier pixel a des possibilités (65536-1) échangeables, le deuxième pixel a une possibilité de permutation (65536-2) et le troisième pixel a une possibilité de permutation (65536-3). Il y a 65536! (Factorial) différentes manières d'échanger ce genre d'image, qui est gigantesque, c'est donc un chiffrement par permutation idéal.

En conséquence de ces deux parties de crypto-système (partie de substitution et partie de permutation), nous pouvons affirmer que notre proposition de ce crypto-système présente toutes les caractéristiques d'un cryptage robuste de toute attaque.

2.7.2 Contribution 2 : Nouvelle Approche du Chiffrement Playfair

Le progrès technologique notamment dans les domaines de la télécommunication et ainsi de l'informatique a fait naissance à une nouvelle forme de sécurité de l'information : Cryptographie. Donc, c'est un moyen de sauvegarder le caractère confidentiel des informations.

Cette deuxième contribution est basée sur le chiffre Playfair [8], qui consiste à chiffrer des paires de lettres (des digrammes), plutôt que des lettres seules comme dans les chiffrements par substitutions poly-alphabétiques tels que le chiffre de Vigenère, plus répandus à l'époque. Durant ces dernières années, il y a eu une explosion de recherche académique publique en cryptographie concernant ce chiffre [52][53][54][55][56][57][58][59][60][61][62][63][64].

Dans ce travail on propose une nouvelle approche du chiffre de Playfair, avec l'associant des concepts de la théorie du chaos. L'idée est d'associer deux cartes chaotiques, l'attracteur Hénon et la carte logistique. Par l'attracteur Hénon qui est de dimension 2, on construit une matrice de Playfair, où on prend comme valeur des éléments de cette matrice, une des valeurs de la carte logistique. Cette valeur correspond

à un caractère unique de la matrice de Playfair et de l'alphabet utilisé. Le mot-clé secret est formé par les conditions initiaux des attracteurs chaotiques.

2.7.2.1 Chiffre Playfair

Le Chiffre de Playfair ou Carré de Playfair est une méthode manuelle de chiffrement symétrique qui fut la première technique utilisable en pratique de chiffrement par substitution polygamique. Il fut imaginé en 1854 par Charles Wheatstone (1802-1875), un des pionniers du télégraphe électrique, mais porte le nom de Lord Playfair qui popularisa son utilisation. Il a été utilisé par les forces britanniques durant la Seconde Guerre des Boers et la Première Guerre mondiale et aussi par les Australiens pendant la Seconde Guerre mondiale.

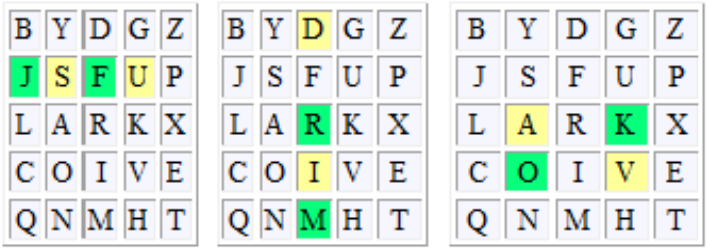
2.7.2.1.1 Méthode de chiffrement

Le chiffre de Playfair utilise un tableau de 5x5 lettres, contenant un mot clé ou une phrase. La mémorisation du mot clé et de 4 règles à suivre suffisent pour utiliser ce chiffrement.

Remplir le tableau avec les lettres du mot clé (en ignorant les doublons), puis le compléter avec les autres lettres de l'alphabet dans l'ordre (soit en omettant la lettre Q, soit en occupant une même case pour les lettres I et J suivant les versions).

Le mot clé peut être écrit en ligne, en colonne ou même en spirale. Pour former les grilles de chiffrement, on utilise un mot-clé secret pour créer un alphabet désordonné avec lequel on remplissait la grille ligne par ligne.

Pour chiffrer un message, il faut prendre les lettres 2 par 2 (des bigrammes) et appliquer les règles suivantes en fonction de la position des lettres dans la table :



1. si les 2 lettres sont identiques (ou s'il n'en reste qu'une) mettre un 'X' après la première lettre. Chiffrer la nouvelle paire ainsi constituée et continuer avec la suivante. Dans certaines variantes, on utilise 'Q' au lieu du 'X', mais n'importe quelle lettre peut faire l'affaire, entre les deux lettres pour éliminer ce doublon
2. si les lettres se trouvent sur la même ligne de la table, il faut les remplacer par celles se trouvant immédiatement à leur droite (en bouclant sur la gauche si le bord est atteint), FJ sera remplacé par US, VE par EC (règle 1).
3. si les lettres apparaissent sur la même colonne, les remplacer par celles qui sont juste en dessous (en bouclant par le haut si le bas de la table est atteint), BJ sera remplacé par JL, RM par ID (règle 2).
4. sinon, remplacer les lettres par celles se trouvant sur la même ligne, mais dans le coin opposé du rectangle défini par la paire originale. Exemple OK devient VA, BI devient DC, GO devient YV. La première des deux lettres chiffrées est sur la même ligne que la première lettre claire (règle 3).

2.7.2.1.2. Exemple de chiffre Playfair

2.7.2.1.2.1 Chiffrement

En supposant que la clé soit « exemple Playfair », le tableau doit alors être rempli comme suit :

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

Chiffrement du message « Cache l'or dans la souche de l'arbre ».
Texte en clair : CA CH EL OR DA NS LA SO UC HE DE LA RB RE
Cryptogramme : BY DB XE QI BF JU ER VJ TD BL BM ER AH AL

2.7.2.1.2.2 Déchiffrement

Pour déchiffrer, utiliser la méthode inverse en ignorant les 'X' ou les 'Q' qui n'ont pas leur place dans le message final, c'est-à-dire en prenant les lettres à gauche

dans le cas d'une même ligne, vers le haut dans le cas d'une même colonne, et toujours les coins dans le cas d'un rectangle.

2.7.2.1.2.3 Cryptanalyse

Ce chiffrement est significativement plus dur à casser car les attaques par analyse fréquentielle habituellement utilisées sur les chiffrements par substitutions simples sont peu efficaces sur lui. L'analyse de fréquence des digrammes reste toujours possible, mais appliquée à $25^2 = 625$ digrammes possibles au lieu des 26 lettres de l'alphabet, elle est considérablement plus difficile et exige un texte chiffré beaucoup plus long pour espérer être efficace.

Mais, comme la plupart des chiffrements anciens, le Chiffre de Playfair peut être cassé si l'on dispose de suffisamment d'échantillons. Obtenir la clé est relativement rapide si l'on a connaissance à la fois du texte chiffré et du texte clair (attaque à texte clair connu). L'utilisation du chaos compliquera les choses.

2.7.2.2 Crypto-système proposé

En premier lieu, il faut savoir qu'on travaille dans un ensemble de caractères qui contient l'alphabet augmenté par deux caractères (#,@), des symboles de ponctuations, les chiffres de base décimaux ainsi que des opérateurs de calcul.

- ✓ L'alphabet fondamental — 26 +2

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

L'alphabet est augmenté par deux caractères :

1. le signe typographique « # » : le croisillon Il est souvent confondu avec le dièse « ♯ ».
2. le caractère typographique @ : l'arobase, a commercial.

- ✓ Symboles spéciaux— 5 caractères : le point, la virgule, le point d'interrogation ainsi que les parenthèses ouverte et fermée.

.	,	?	(	)

- ✓ Chiffres — 10 caractères de 0 à 9

0	1	2	3	4
5	6	7	8	9

- ✓ Opérateurs— 6 caractères : l'addition, la soustraction, la division, la multiplication représentée par *, l'égalité et le signe de pourcentage :

+	-	/	*	=	%

- ✓ On a (26+2+5+10+6 =) 49 caractères, on les mets dans une matrice de dimension 7x7, comme suit :

	0	1	2	3	4	5	6
0	*A*	*B*	*C*	*D*	*E*	*F*	*G*
1	*H*	*I*	*J*	*K*	*L*	*M*	*N*
2	*O*	*P*	*Q*	*R*	*S*	*T*	*U*
3	*V*	*W*	*X*	*Y*	*Z*	*#*	*@*
4	*.*	*,*	*?*	*(*	*)*	*0*	*1*
5	*2*	*3*	*4*	*5*	*6*	*7*	*8*
6	*9*	*+*	*-*	*/*	***	*=*	*%*

Tableau.2.3 Matrice de base de Playfair

Il y'a une biunivoque entre la matrice de base de Playfair, le modulo (49) et le caractère proprement dit. N'importe quel caractère est représenté par sa valeur dans la matrice de base de Playfair comme suit :

Un caractère → valeur (ligne *7 + colonne)

L'idée est de remplir la matrice de chiffrement de Playfair de dimension (7 X 7), lignes et colonnes comme suit :

$$Lignei = mod(fix\ (x_n * 10^7), 7)$$

Colonne j= mod(fix (y_n*10^7), 7)

Valeur (i,j)= mod(fix (z_n*10^7), 49)

x_n \ y_n	0	1	2	3	4	5	6
0							
1				$y_n \approx j$ ↓			
2			$x_n \approx i$ →	Z_n			
3							
4							
5							
6							

On propose l'organigramme 2.11 :

1. Avec les conditions initiales par exemple (x_0=0.1 et y_0=0.1), et a=1.4 et b=0.3 : on calcule les positions i et j selon le principe de l'attracteur de Hénon :

   Lignei= mod(fix (x_n*10^7), 7),

   Colonne j= mod(fix (y_n*10^7), 7)
2. Calculer la valeur z de la carte logistique selon l'équation :

   z_n = mod(fix(z_n*10^7), 49)

   Avec z_0=0.01, r=3.9
3. Remplir la matrice de chiffrement de playfair: k(i, j)=z_n (z_n la valeur construite par la carte logistique), et on garde le couple (i, j) pour qu'on remplit le pixel de la matrice k une seul fois. Cette valeur est aussi déduite par rapport à notre matrice de base de Playfair.
4. On refaire les étapes 1, 2, 3 et 4 jusqu'on termine le remplissage de la matrice k.

On chiffre le texte en clair avec cette matrice k selon les mêmes principes de base du chiffre de Playfair. Les deux caractères (#,@) peuvent être utilisés pour éliminer les doublons de lettres, ou s'il n'en reste qu'une seule lettre vers la fin du texte en clair. Le choix de ce caractère est arbitraire.

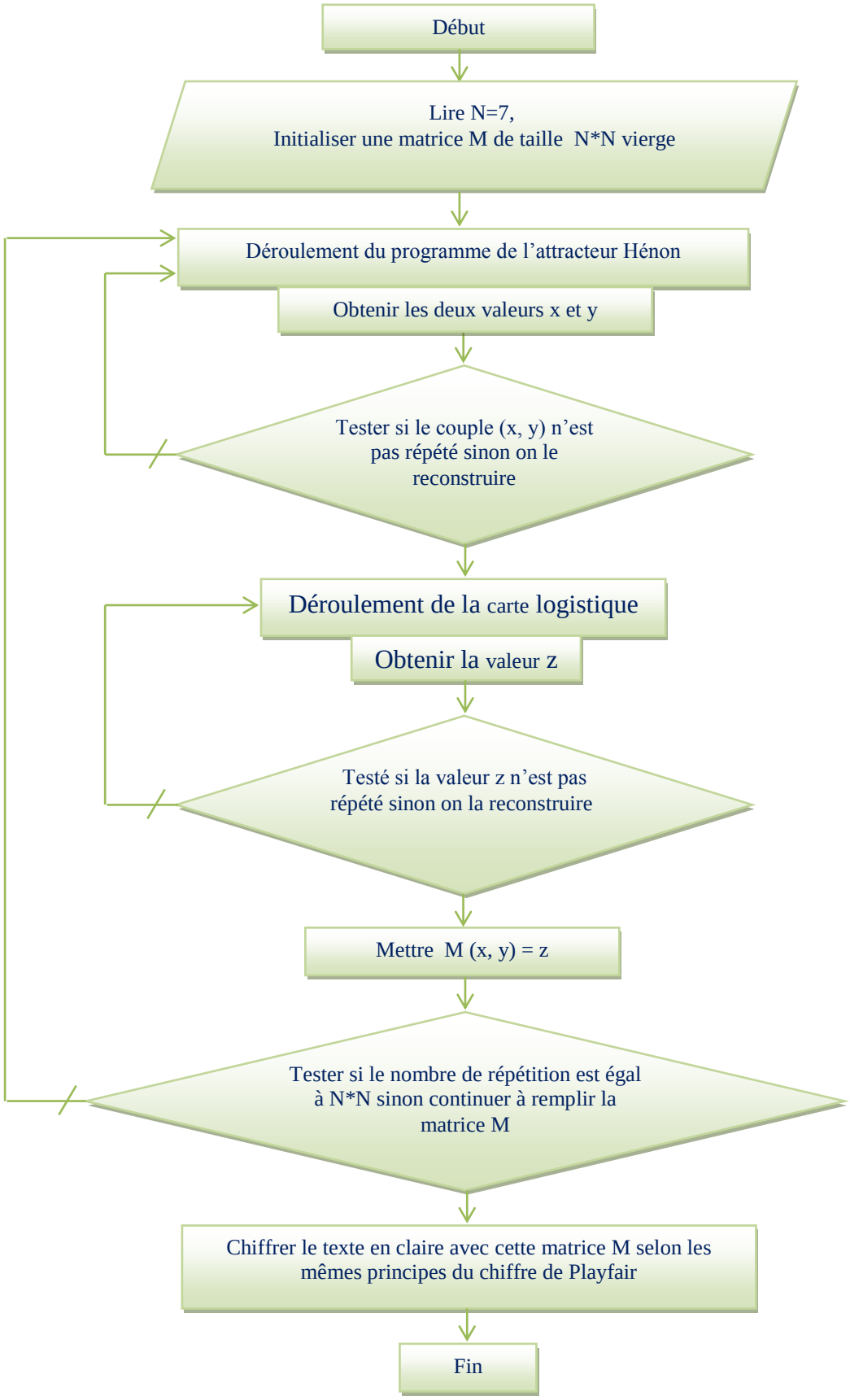


Figure.2.11 Organigramme de la nouvelle approche de chiffrement Playfair

2.7.2.3 Résultats et interprétations

La taille de la clé de chiffrement de l'espace est le nombre total de différentes valeurs qui peuvent être utilisés dans le procédé de cryptage.

Dans l'algorithme proposé, le champ de clé secrète est fixé comme suit : ST = {x_0, y_0, a, b, z_o, r, N_1, N_2}. Où x_0, y_0,a, b,z_o,r sont des nombres à doubles précision. N_1, N_2 sont des constantes entières. Si la précision de calcul de x_0, y_0, a, b,z_o, r est 10^{-16}, et $N_1| N_2 \in [1, 1000]$.

Par conséquent, l'espace de clé est plus grand que 10^{16} x 10^{16} x 10^{16} x 10^{16} x 10^{16} x 10^{16}x 1000 x 1000, qui est beaucoup plus grand que : 2^{330}.

Donc, l'algorithme de chiffrement a un très grand espace clé pour résister à toutes sortes d'attaques par force brute. Pour des raisons pratiques, on fixe les valeurs a, b, et r la clé de chiffrement sera constituée par STN = {x_0, y_0, z_o, N_1, N_2}, dans ce cas on aura un champ de clé de l'ordre de 2^{180} est c'est énorme.

On donne quelques matrices de chiffrement de Playfair pour différentes clés de chiffrement. On pend comme valeurs fixes : a = 1.4, b = 0.3, r = 3.9.

1) Les nouvelles valeurs variables qui constituent la clé de chiffrement sont :

x_0=0.95, y_0=0.56, z_0= 0.02, N_1=N_2=19

	0	**1**	**2**	**3**	**4**	**5**	**6**
0	*A*	*?*	*I*	*-*	*H*	*T*	*C*
1	*0*	*=*	*N*	*.*	*K*	*V*	*)*
2	*#*	*U*	*@*	*2*	*M*	*4*	*+*
3	*(*	*L*	*3*	*8*	*J*	*Y*	*D*
4	*/*	*P*	*X*	*E*	*G*	*F*	*,*
5	*S*	*%*	*9*	*5*	*W*	***	*R*
6	*6*	*O*	*7*	*1*	*Z*	*B*	*Q*

Tableau.2.4 Matrice du chiffrement N°1

2) Les nouvelles valeurs variables qui constituent la clé de chiffrement sont :

x_0=0.9500000001, y_0=0.56, z_0= 0.02, N_1=N_2=19

	0	1	2	3	4	5	6
0	W	(	T	6	N	1	C
1	#	F	X	7	D	J	/
2	P	R	+	.	@	G	H
3	?	2	M	8	Y	)	O
4	0	-	L	S	=	I	%
5	U	S	V	3	B	Z	E
6	K	4	9	*	,	Q	A

Tableau 2.5 Matrice du chiffrement N°2

3) Les nouvelles valeurs variables qui constituent la clé de chiffrement sont :

x_0=0.95, y_0=0.5600000001, z_0= 0.02, N_1=N_2=19

	0	1	2	3	4	5	6
0	E	5	O	T	B	8	N
1	9	Q	L	0	U	#	W
2	P	D	C	-	M	F	X
3	.	,	6	S	?	K	%
4	Y	2	=	4	@	1	G
5	7	)	3	H	R	*	V
6	A	+	/	I	Z	(	J

Tableau 2.6 Matrice du chiffrement N°3

4) Les nouvelles valeurs variables qui constituent la clé de chiffrement sont :

x_0=0.95, y_0=0.56, z_0= 0.0200000001, N_1=N_2=19

	0	1	2	3	4	5	6
0	R	A	.	4	3	9	6
1	F	B	M	V	?	0	J
2	E	K	Y	Z	D	N	+
3	)	H	G	C	5	X	#
4	7	*	(	,	@	1	I
5	T	%	2	U	W	S	8
6	Q	O	L	-	=	/	P

Tableau 2.7 Matrice du chiffrement N°4

5) Les nouvelles valeurs variables qui constituent la clé de chiffrement sont :

x_0=0.95, y_0=0.56, z_0= 0.02, N_1= 18, N_2=19

	0	1	2	3	4	5	6
0	*A*	*?*	*I*	*-*	*H*	*T*	*C*
1	*0*	*=*	*N*	*.*	*K*	*V*	*)*
2	*#*	*U*	*@*	*2*	*M*	*4*	*+*
3	*(*	*L*	*3*	*8*	*J*	*Y*	*D*
4	*/*	*P*	*X*	*E*	*G*	*F*	*,*
5	*S*	*%*	*9*	*5*	*W*	***	*R*
6	*6*	*O*	*7*	*1*	*Z*	*B*	*Q*

Tableau 2.8 Matrice du chiffrement N°5

6) Les nouvelles valeurs variables qui constituent la clé de chiffrement sont :

x_0=0.95, y_0=0.56, z_0= 0.02, N_1=19, N_2=18

	0	1	2	3	4	5	6
0	*3*	*?*	*I*	*L*	*A*	*T*	*C*
1	*@*	*K*	*V*	*.*	*N*	*M*	*)*
2	*-*	*=*	*5*	*Z*	*F*	*4*	*D*
3	*(*	*Q*	*#*	*8*	*J*	***	*2*
4	*/*	*P*	*+*	*E*	*G*	*Y*	*,*
5	*S*	*%*	*9*	*H*	*W*	*0*	*R*
6	*6*	*X*	*7*	*O*	*U*	*B*	*1*

Tableau 2.9 Matrice du chiffrement N°6

7) Les nouvelles valeurs variables qui constituent la clé de chiffrement sont :

x_0=0.95, y_0=0.56, z_0= 0.02, N_1=N_2=19, Scalaire : $x=10^7+1$, $y=10^7+1$, $z=10^7+1$

	0	1	2	3	4	5	6
0	*R*	*W*	*A*	*K*	*S*	*5*	***
1	*3*	*Z*	*1*	*L*	*#*	*J*	*0*
2	*P*	*E*	*U*	*?*	*,*	*@*	*M*
3	*V*	*2*	*B*	*-*	*)*	*H*	*9*
4	*/*	*(*	*C*	*4*	*=*	*F*	*X*
5	*N*	*Q*	*6*	*D*	*8*	*O*	*7*
6	*T*	*Y*	*.*	*I*	*G*	*+*	*%*

Tableau2.10 Matrice du chiffrement N°7

8) Les nouvelles valeurs variables qui constituent la clé de chiffrement sont :

x_0=0.95, y_0=0.56, z_0= 0.02, N_1=N_2=19, Scalaire : x=10^7, y=10^7, z=10^7+1

	0	**1**	**2**	**3**	**4**	**5**	**6**
0	*B*	*A*	*9*	*/*	*I*	*F*	*Y*
1	*1*	*@*	*O*	*?*	*K*	*W*	*0*
2	*=*	*U*	*.*	*#*	*M*	*8*	*D*
3	*(*	*L*	*3*	*N*	*T*	*Z*	*2*
4	*7*	*)*	*-*	*J*	*E*	*G*	*,*
5	*R*	*%*	*6*	*5*	*+*	***	*S*
6	*4*	*X*	*V*	*P*	*A*	*C*	*Q*

Tableau 2.11 Matrice du chiffrement N°8

9) Les nouvelles valeurs variables qui constituent la clé de chiffrement sont :

x_0=0.95, y_0=0.56, z_0= 0.02, N_1=N_2=19, Scalaire : x=10^7+1, y=10^7, z=10^7

	0	**1**	**2**	**3**	**4**	**5**	**6**
0	*D*	*F*	*Z*	*N*	*5*	*U*	*M*
1	*3*	*#*	*0*	*J*	*2*	*8*	*L*
2	*-*	*7*	*=*	*)*	*K*	*Y*	*,*
3	*6*	*+*	*A*	*1*	*G*	*4*	*T*
4	*S*	*W*	*?*	*.*	*R*	*B*	*Q*
5	*/*	*%*	*(*	*O*	*X*	*V*	*P*
6	*9*	*E*	*I*	*@*	*H*	***	*C*

Tableau.2.12 Matrice du chiffrement N°9

10) Les nouvelles valeurs variables qui constituent la clé de chiffrement sont :

x_0=0.95, y_0=0.56, z_0= 0.02, N_1=N_2=19, Scalaire : x=10^7, y=10^7+1, z=10^7

	0	**1**	**2**	**3**	**4**	**5**	**6**
0	*A*	*U*	*4*	*L*	*=*	*H*	*6*
1	*+*	*0*	*N*	*P*	*K*	*V*	*.*
2	*-*	*Z*	*@*	*D*	*E*	*M*	*)*
3	*#*	*Q*	*3*	*7*	*(*	*T*	*Y*
4	*C*	*X*	*?*	*%*	*,*	*F*	*S*
5	*R*	*W*	*/*	*5*	*9*	***	*B*
6	*I*	*1*	*G*	*J*	*O*	*2*	*8*

Tableau.2.13 Matrice du chiffrement N°10

On remarque à travers les matrices de chiffrement N°2, N°3 et N°4 par rapport à la matrice de chiffrement N°1, qu'elles sont totalement différentes malgré la différence minime des champs de la clé. Ce qui prouve, que ce que nous avons proposé donne des résultats très satisfaisants.

La même remarque on peut la faire en comparaissant les matrices de chiffrement restantes à la matrice de chiffrement N°1.

2.7.2.4 Conclusion

Dans ce travail, nous avons présenté un nouveau système de chiffrement de Playfair. Nous avons gardé le même principe de chiffrent, mais en modifiant la dimension du tableau de 5x5 caractères à une dimension de 7x7 caractères, et nous avons enlevé le mot clé ou la phrase à retenir, et nous l'avons remplacer par une clé de 180 bits.

Le nouveau chiffré ainsi représenté a donné de très bon résultat, et il y'a un très grand espace clé pour résister à toutes sortes d'attaques par force brute.

2.7.3 Contribution 3 : Construction d'une Suite aléatoire par le biais de la Carte PWLCM : Application au RC4

Le chiffrement par flot [2] [3] consiste à utiliser un XOR entre les données en clair et une suite aléatoire. En cryptographie asymétrique, il est nécessaire de produire de grands nombres aléatoires avec des contraintes supplémentaires (premier, premier entre eux, etc.). De plus, un texte chiffré doit s'approcher le plus possible d'un fichier au contenu aléatoire pour limiter les fuites d'information.

La carte PWLCM est souvent employée pour générer des données aléatoires qui sont employées dans les applications cryptographiques. Dans cette troisième contribution, nous proposons une variante de la carte PWLCM susceptible de nous donner des meilleures valeurs que nous les utilisons au flux RC4.

2.7.3.1 RC4 (RivestCipher 4)

RC4 est l'un des chiffrements de flux, c'est un algorithme symétrique, c'est-à-dire on utilise le même principe pour chiffrer ou déchiffrer un texte. [11]

C’est un logiciel parmi les logiciels les plus largement utilisés. En plus d'être utilisé dans les protocoles réseau tels que SSL, TLS, WEP et WPA, le chiffre trouve des applications dans Microsoft Windows, Lotus Notes, Apple AOCE, Oracle SQL sécurisé… etc. Malgré diverses autres chiffrements de flux ont été proposés après RC4, il est encore le flux le plus populaire utilisé en raison de sa simplicité, de facilité de mise en œuvre, la vitesse et l'efficacité. [65] Il été conçu en 1987 par Ronald Rivest, l'un des inventeurs du RSA, pour les Laboratoires RSA.

Cet algorithme est un générateur pseudo-aléatoire qui génère une suite d’octets (appelée keystream). Ces octets sont ensuite combinés au texte à chiffrer par un OU exclusif (XOR). L'algorithme peut être décomposé en deux phases : l'algorithme de calcul de clé (KSA) et l’algorithme de génération pseudo-aléatoire (RPAG), la KSA tourne la clé secrète K en une permutation aléatoire de S 0, 1, . . ., N- 1 et PRGA utilise cette permutation pour générer des flux (octets) de clefs pseudo-aléatoires. L'octet de sortie de keystream est XOR-er avec l'octet de message pour générer l'octet de texte chiffré à la fin de l'expéditeur. Encore une fois, z est XOR-er avec l'octet chiffré pour récupérer l'octet de message à la fin du récepteur.

2.7.3.1.1 Algorithme de planification de clé (Key Scheduling AlgorithmKSA)

```
# Initialisation de la permutation identité
pour  i = 0 à 255 alors
        S[i] := i
        T[i] = K[i mod keylen] #c’est la clé d’entrée
fin pour
j := 0
# Mélange de S dépendant de K
pour i = 0 à 255 alors
        j := (j + S[i] + T[i]) % 256
        Changer(S[i], S[j])
fin pour
i = j = 0
tant que (vrai)
        i = (i + 1) mod 256
        j = (j + S[i]) mod 256
```

Changer (S[i], S[j])

t = (S[i] + S[j]) mod 256

$C_i = M_i$ XOR S[t]

fin

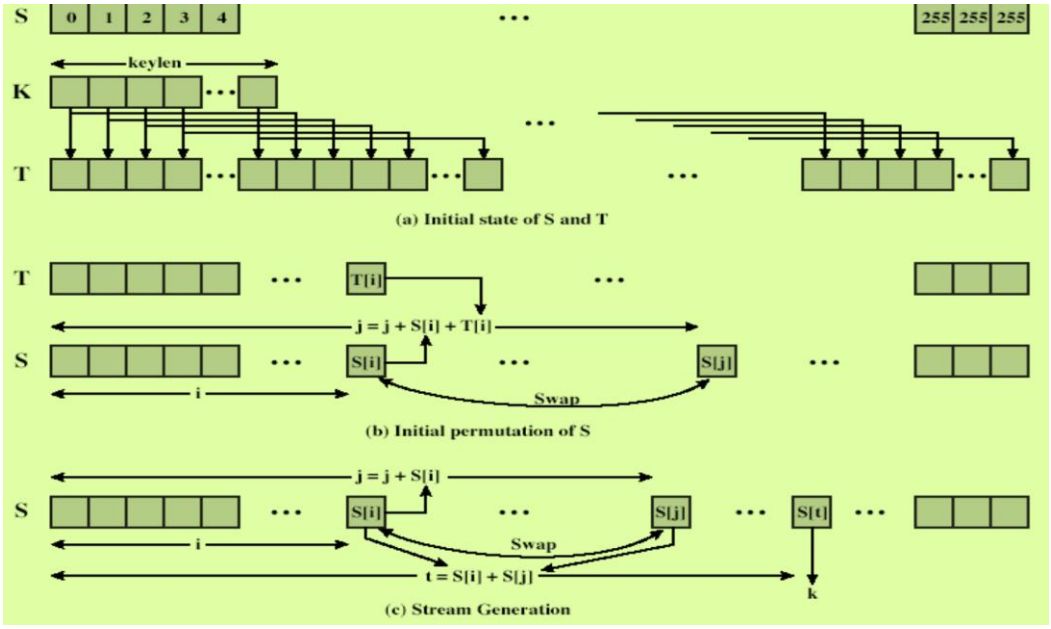


Figure.2.12 Schéma représente la première phase du RC4 (KSA)

2.7.3.1.2 Algorithme de générateur pseudo aléatoire (Pseudo RandomGeneratorAlgorithm PRGA)

i = j = 0

tant que (vrai)

i = (i + 1) mod 256

j = (j + S[i]) mod 256

Changer (S[i], S[j])

t = (S[i] + S[j]) mod 256

$C_i = M_i$ XOR S[t]

fin

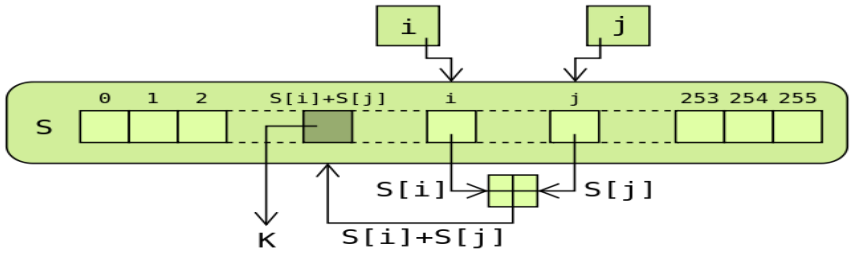


Figure.2.13 Schéma représente la deuxième phase du RC4 (PRGA)

2.7.3.1.3 Exemples numérique

Considérons le chiffrement de flux RC4, mais au lieu de la totalité des 256 octets, nous allons utiliser 8 x 3 bits. Autrement dit, le vecteur d'état S est de 8 x 3 bits. Nous allons fonctionner sur 3 bits de texte en clair à un moment puisque S peut prendre les valeurs de 0 à 7, qui peuvent être représentés que 3 bits.

1)

- Supposons que nous utilisons une clé 4 x 3 bits de K = [1 2 3 6]. Et un texte en clair P = [1 2 2 2].
- La première étape consiste à générer le flux.
- Initialiser le vecteur d'état S et le vecteur temporaire T. S est initialisé ainsi le S [i] = i, et T est initialisé donc par la clé K (répétée si nécessaire).
- S = [0 1 2 3 4 5 6 7], T = [1 2 3 6 1 2 3 6]
- Maintenant effectuer la permutation initiale sur S.

j = 0;

pour i = 0 à 7 alors

j = (j + S[i] + T[i]) mod 8

Changer(S[i],S[j]);

fin

- pour i=0, j=1, S= [1 0 2 3 4 5 6 7]

  pour i=1, j=3, S= [1 3 2 0 4 5 6 7]

  pour i=2, j=0, S= [2 3 1 0 4 5 6 7]

  pour i=3, j=6, S= [2 3 1 6 4 5 0 7]

  pour i=4, j=3, S= [2 3 1 4 6 5 0 7]

  pour i=5, j=2, S= [2 3 5 4 6 1 0 7]

  pour i=6, j=5, S= [2 3 5 4 0 1 6 7]

  pour i=7, j=2, S= [2 3 7 4 0 1 6 5]

- Par conséquent, notre permutation initiale de S= [2 3 7 4 0 1 6 5]
- Maintenant, nous générons 3-bits à la fois, k, que nous XOR avec chacun 3 bits de texte clair pour produire le texte chiffré. Les 3-k bits est générée par:

```
i=j = 0;
tant que (vrai)
        i = (i + 1) mod 8;
          j = (j + S[i]) mod 8;
        Changer (S[i], S[j]);
        t = (S[i] + S[j]) mod 8;
        k = S[t];
fin
```

- i=1, j=3, S= [2 4 7 3 6 0 1 5], t=6, k=1.
  i=2, j=2, S= [2 4 7 3 6 0 1 5], t=5, k=0.
  i=3, j=5, S= [2 4 7 0 6 3 1 5], t=2, k=7.
  i=4, j=3, S= [2 4 7 6 0 3 1 5], t=5, k=3.
- Donc, pour chiffrer le flux de texte en clair P = [1 2 2 2] avec la clé K = [1 2 3 6] en utilisant notre simplifié flux RC4 nous obtenons C = [0 2 5 1].

2)

- Supposons que nous utilisons une clé 4 x 3 bits de K = [1 5 6 3]. Et un texte en clair P = [1 2 3 4].
- La première étape consiste à générer le flux.
- Initialiser le vecteur d'état S et le vecteur temporaire T. S est initialisé ainsi le S [i] = i, et T est initialisé donc par la clé K (répétée si nécessaire).
- S = [0 1 2 3 4 5 6 7], T = [1 5 6 3 1 5 6 3]
- Maintenant effectuer la permutation initiale sur S.

```
j = 0;
pour i = 0 à 7 alors
          j = (j + S[i] + T[i]) mod 8
        Changer(S[i],S[j]);
fin
```

- pour i=0, j=1, S= [1 0 2 3 4 5 6 7]
  pour i=1, j=6, S= [1 6 2 3 4 5 0 7]
  pour i=2, j=6, S= [1 6 0 3 4 5 2 7]

pour i=3, j=4, S= [1 6 0 4 3 5 2 7]

pour i=4, j=0, S= [3 6 0 4 1 5 2 7]

pour i=5, j=2, S= [3 6 5 4 1 0 2 7]

pour i=6, j=2, S= [3 6 2 4 1 0 5 7]

pour i=7, j=4, S= [3 6 2 4 7 0 5 1]

- Par conséquent, notre permutation initiale de S= [3 6 2 4 7 0 5 1]
- Maintenant, nous générons 3-bits à la fois, k, que nous XOR avec chacun 3 bits de texte clair pour produire le texte chiffré. Les 3-k bits est générée par:

```
i=j = 0;
tant que (vrai)
        i = (i + 1) mod 8;
        j = (j + S[i]) mod 8;
        Changer (S[i], S[j]);
        t = (S[i] + S[j]) mod 8;
        k = S[t];
fin
```

- i=1, j=6, S= [3 5 2 4 7 0 6 1], t=2, k=2.
  i=2, j=0, S= [2 5 3 4 7 0 6 1], t=4, k=7.
  i=3, j=4, S= [2 5 3 7 4 0 6 1], t=2, k=3.
  i=4, j=0, S= [4 5 3 7 2 0 6 1], t=5, k=0.
- Donc, pour chiffrer le flux de texte en clair P = [1 2 3 4] avec la clé K = [1 5 6 3] en utilisant notre simplifié flux RC4 nous obtenons C = [3 5 0 4].

2.7.3.1.4 Résumé

Certaines des caractéristiques de l'algorithme RC4 peuvent être résumées comme suit:

1- symétrique chiffrement de flux

2- longueur de clé variable.

3- Très rapide dans les logiciels

4- Utilisé pour les communications sécurisées comme dans le chiffrage du trafic vers et depuis les sites Web en utilisant le protocole SSL.

2.7.3.2 Algorithme proposé

Nous avons essayé de remplacer la suite des s-box utilisé dans le principe de RC4 par des valeurs aléatoire retournées par la carte PWLCM, on aura donc le nouveau principe de l'RC4 comme suit :

 # Initialisation de la permutation identité

x=suite aléatoire retourné par la carte PWLCM de longueur 256

pour i = 0 to 255 alors

S[i] := $[x[i] * 10^{14} \%256]$

T[i] = T[i] = K [i mod keylen] *# c'est la clé d'entrée*

fin pour

j := 0

 # Mélange de S dépendant de K

pour i = 0 à 255 alors

j := (j + S[i] + T[i]) % 256

Changer(S[i], S[j])

fin pour

i = j = 0

tant que (vrai)

i = (i + 1) mod 256

j = (j + S[i]) mod 256

Changer (S[i], S[j])

t = (S[i] + S[j]) mod 256

$C_i = M_i$ XOR S[t]

fin tant que

Exemples numérique:

On utilise les mêmes exemples c'est-à-dire :

Considérons le chiffrement de flux RC4, mais au lieu de la totalité des 256 octets, nous allons utiliser 8 x 3 bits. Autrement dit, le vecteur d'état S est de 8 x 3 bits. Nous allons fonctionner sur 3 bits de texte en clair à un moment puisque S peut prendre les valeurs de 0 à 7, qui peuvent être représentés que par 3 bits.

1)

- Supposons que nous utilisons une clé 4 x 3 bits de K = [1 2 3 6]. Et un texte en clair P = [1 2 2 2].
- La première étape consiste à générer la suite aléatoire par la carte PWLCM. Avec les valeurs initiales de la carte PWLCM (x_0=0.001, p=0.37, n=8) on obtient ces valeurs : 0.0027 0.0073 0.0197 0.0534 0.1442 0.3898 0.0043 0.0115
- Initialiser le vecteur S et le vecteur temporaire T. T est initialisé par la clé K (répétée si nécessaire), et S est initialisé par les valeurs retournées par la carte PWLCM. Pour rendre ces valeurs entières on va les multiplier par 10^{14} et on garde la partie entière du reste de division sur 256 ainsi on obtient :
- S=[6 7 0 6 6 0 2 4], T=[1 2 3 6 1 2 3 6]
- Maintenant effectuer la permutation initiale sur S.

```
j = 0;
pour i = 0 à 7 alors
        j = (j + S[i] + T[i]) mod 8
        Changer(S[i],S[j]);
fin
```

- pour i=0, j=7, S= [4 7 0 6 6 0 2 6]
  pour i=1, j=0, S= [7 4 0 6 6 0 2 6]
  pour i=2, j=3, S= [7 4 6 0 6 0 2 6]
  pour i=3, j=1, S= [7 0 6 4 6 0 2 6]
  pour i=4, j=0, S= [6 0 6 47 0 2 6]
  pour i=5, j=2, S= [6 0 0 4 7 6 2 6]
  pour i=6, j=7, S= [6 0 0 4 7 6 6 2]
  pour i=7, j=7, S= [6 0 0 4 7 6 6 2]
- Par conséquent, notre permutation initiale de S= [6 0 0 4 7 6 6 2]
- Maintenant, nous générons 3-bits à la fois, que nous XOR avec chacun 3 bits de texte clair pour produire le texte chiffré. Les 3-k bits est générée par:

  i=j = 0;

tant que (vrai)

i = (i + 1) mod 8;

j = (j + S[i]) mod 8;

Changer (S[i], S[j]);

t = (S[i] + S[j]) mod 8;

k = S[t];

fin

- S= [6 0 0 4 7 6 6 2], t=5, k=6.
  S= [6 0 0 4 7 6 6 2], t=5, k=6.
  S= [6 0 0 6 7 4 6 2], t=4, k=7.
  S= [6 0 0 6 7 4 6 2], t=4, k=7.
- Donc, pour chiffrer le flux de texte en clair P = [1 2 2 2] avec les valeurs retournée par la carte PWLCM [6 7 0 6 6 0 2 4] et la clé K = [1 2 3 6] on utilise le même principe de flux RC4 ainsi nous obtenons le message chiffré C = [7 4 5 5].

2)

- Supposons que nous utilisons une clé 4 x 3 bits de K = [1 5 6 3], et un texte en clair P = [1 2 3 4].
- La première étape consiste à générer la suite aléatoire par la carte PWLCM.
- Avec les valeurs initiales de la carte PWLCM (x_0=0.001, p=0.37, n=8) on obtient ces valeurs : 0.0027 0.0073 0.0197 0.0534 0.1442 0.3898 0.0043 0.0115
- Initialiser le vecteur S et le vecteur temporaire T. T est initialisé par la clé K (répétée si nécessaire), et S est initialisé par les valeurs retournées par la carte PWLCM. Pour rendre ces valeurs entières on va les multiplier par 10^{14} et on garde la partie entière du reste de division sur 256 donc on obtient :
- S=[6 7 0 6 6 0 2 4], T=[1 5 6 3 1 5 6 3]
- Maintenant effectuer la permutation initiale sur S.

j = 0;

pour i = 0 à 7 alors

j = (j + S[i] + T[i]) mod 8

Changer(S[i],S[j]);

fin

- pour i=0, j=7, S= [4 7 0 6 6 0 2 6]

  pour i=1, j=0, S= [7 4 0 6 6 0 2 6]

  pour i=2, j=3, S= [7 4 6 0 6 0 2 6]

  pour i=3, j=7, S= [7 4 6 6 6 0 2 0]

  pour i=4, j=6, S= [7 4 6 6 2 0 6 0]

  pour i=5, j=0, S= [0 4 6 6 2 7 6 0]

  pour i=6, j=1, S= [0 6 6 6 2 7 4 0]

  pour i=7, j=5, S= [0 6 6 6 2 0 4 7]

- Par conséquent, notre permutation initiale de S= [0 6 6 6 2 0 4 7]
- Maintenant, nous générons 3-bits à la fois, que nous XOR avec chacun 3 bits de texte clair pour produire le texte chiffré. Les 3-k bits est générée par:

I=j = 0;

tant que (vrai)

i = (i + 1) mod 8;

j = (j + S[i]) mod 8;

Changer (S[i], S[j]);

t = (S[i] + S[j]) mod 8;

k = S[t];

fin

- S= [0 7 6 6 2 0 4 6], t=2, k=6.

  S= [0 7 0 6 2 6 4 6], t=1, k=7.

  S= [0 7 0 6 2 6 4 6], t=5, k=6.

  S= [0 7 0 6 6 2 4 6], t=3, k=6.

- Donc, pour chiffrer le flux de texte en clair P = [1 2 3 4] avec les valeurs retournée par la carte PWLCM [6 4 5 1 2 3 5 6] et la clé K = [1 2 3 6] en utilisant le même principe de flux RC4 nous obtenons le message chiffré C = [7 2 0 5].

RC4 est composé de deux phases, la première consiste à initialiser notre suite comme suit :

$$S_0 = 0, S_1 = 1, S_2 = 2, \dots\dots, S_{255} = 255$$

En appliquant une permutation sur cette suite à l'aide de la clé donc on obtient une nouvelle suite.

La deuxième phase consiste à modifier à chaque fois deux éléments de la suite, et on chiffre le message à l'aide de cette suite.

Notre idée est de remplacer les s –box par des valeurs calculées à l'aide d'un attracteur chaotique (PWLCM), cet attracteur utilise des valeurs initiales pour produire des valeurs aléatoires; c'est-à-dire on ajoute une deuxième clé au flux RC4 (utilisé par l'attracteur chaotique) ce qui augmentera la sécurité du chiffrement.

2.7.3.3 Conclusion

La nouveauté de cet algorithme est de faire une complexité de l'initialisation des s-box c'est-à-dire au lieu d'initialisé $S_i = i$ on va initialiser avec $S_i = PWLCM(i)$ sachant que PWLCM c'est une valeur retournée par la carte avec une valeur entière.

Cet algorithme RC4 amélioré par cette carte PWLCM par les mêmes principes de fonctionnement que le RC4, mais sa clé de chiffrement est augmenté avec les valeurs initiaux de la carte chaotique.

Le chaos a trouvé de nombreuses applications dans des différents domaines tel que : physique, biologique, chimique ou économique. Dans ce chapitre nous avons consacré sur la cryptographie chaotique. Au premier lieu nous avons présenté une définition sur le chaos. Puis nous avons parlé sur la Carte Logistique et ses variantes et nous avons vu l'utilisation de cette carte dans la cryptographie tel que la génération des nombres aléatoire. Ensuite nous parlerons aussi sur l'attracteur Hénon et la carte PWLCM. En fin nous avons concaténé les deux systèmes chaotiques la Carte Logistique et l'attracteur Hénon pour former un crypto-système pour le chiffrement des images, et pour voir une nouvelle approche du chiffre Playfair, et nous avons utilisé la Carte PWLCM pour construire une suite aléatoire et l'appliquer au flux RC4.

CHAPITRE 3

CONSTRUCTION D'UNE SUITE SUPER-CROISSANTE

Chapitre 3

Construction d'une suite super-croissante

3.1 Introduction

En 1978, Ralph Merkle et Martin Hellman proposèrent un crypto-système à clé publique basé sur un problème célèbre : le problème du sac à dos (Knapsack problem) [66].

Le problème du sac à dos est l'un des 21 problèmes NP-complets de Richard Karp, exposés dans son article de 1972 [70] [71]. La formulation du problème est fort simple, mais sa résolution est plus complexe. Cependant, la structure singulière du problème, et le fait qu'il soit présent en tant que sous-problème dans d'autres problèmes plus généraux, en font un sujet de choix pour la recherche. Il consiste à empiler des objets dans un sac, de manière à atteindre (si possible) un poids total fixé. Plus formellement, étant donnés des poids entiers $P_1, \dots, P_n$ et un but T, il s'agit de trouver $b_1, \dots, b_n$, valant 0 ou 1, tels que

$$T = b_1P_1 + b_2P_2 + \cdots + b_nP_n$$

Si la suite des poids P_k est super-croissante (chaque poids est strictement supérieur à la somme de tous les précédents), alors il existe une méthode de résolution simple (algorithme glouton) :

Algorithme glouton

Pour i = n à 1 faire
  Si T ≥ Pi alors
    T = T - Pi
    bi = 1
  sinon
    bi = 0
Si T = 0 alors {b_1, ..., b_n} est solution sinon il n'y a pas de solution.

Vérifiez qu'avec cette suite super-croissante P_1=2, P_2=3, P_3=6, P_4=12 et T=15 on obtient la solution b_1=0, b_2=1, b_3=0, b_4=1.

Au contraire, si la suite des poids n'est pas super-croissante, le seul algorithme connu consiste à essayer successivement toutes les solutions $(b_1, b_2, \dots, b_n)$ possibles. Si la suite est suffisamment longue, c'est un algorithme impraticable.

Le problème du sac à dos fournit un autre exemple de fonction à sens unique (pour x fixé, le calcul de f(x) est très facile, mais l'inverse est impossible). Ce problème est encore connu sous le nom de knapsack problem.

Il est aussi utilisé en cryptographie comme une base pour différents schémas de chiffrement. Il faut cependant noter que la plupart de ces schémas de chiffrement ne sont plus actuellement considérés comme sûrs.

Les travaux de recherche récents qui traitent des crypto-systèmes [68] [69] utilisent systématique des suites super-croissantes quelconque sans aucune précision de leur provenance.

Dans ce chapitre, on essaie de proposer des méthodes basées soit sur la carte logistique pour la construction d'une suite super-croissante soit sur la séquence **de Fibonacci Généralisée à Coefficients réels**, et de l'utiliser dans le système à clé publique de Merkle-Helman.

3.2 Crypto-système Merkle-Helman

En 1978 [66], Ralph Merkle et Martin Hellman proposèrent un crypto-système à clé publique basé sur ce problème célèbre: le problème du sac à dos (Knapsack problem) [66]. Qui est sous la forme suivante : Imaginons une collection de cailloux de poids $\{a_1, a_2, \dots, a_n\}$ connus. Supposons que l'on place certains de ces cailloux dans un sac à dos et que l'on pèse le tout.

Est-il possible, connaissant ce poids total, de savoir quels sont les cailloux qui sont dans le sac ? La réponse est très difficile, mais c'est possible dans le cas où les cailloux forment une suite super-croissante.

Alors, Merkle et Hellman modélisent une situation analogue au remplissage d'un sac à dos, ne pouvant supporter plus d'un certain poids, avec tout ou partie d'un ensemble donné d'objets ayant chacun un poids et une valeur. Les objets

mis dans le sac à dos doivent maximiser la valeur totale, sans dépasser le poids maximum.

Le crypto-système de Merkle et Hellman [2][3][67] utilise le problème de sac à dos de la manière suivante :

3.2.1 Génération des clés

- Un entier positif *n* suffisamment grand (Merkle et Hellman recommandaient de prendre *n* de l'ordre de 100).
- Choisir une suite $\{b_1, b_2, \dots, b_n\}$ d'entiers positifs vérifiant la propriété suivante :

$$\forall\, i \in [2, n], b_i > \sum_{j=1}^{i-1} b_j$$

- Choisir un entier M, appelé *module*, tel que :

$$M > \sum_{i=1}^{n} b_i$$

- Choisir un entier $W \in [1, M-1]$ premier avec *M*, i.e. tel que le $pgcd(W, M) = 1$.
- Calculer :

  $ai' = W \times b_i \bmod M$ pour $i \in [1, n]$
- C'est-à-dire calculer le produit de *W* et b_i et ne conserver que le reste de la division euclidienne du résultat par *M*, l'entier obtenu est donc compris entre *0* et *M-1*).
- Calculer la permutation π de *{1, 2,..., n}* telle que $\{a_{\pi(1)}', a_{\pi(2)}', \dots, a_{\pi(n)}'\}$soit une suite croissante.
- La clé publique est :

$$(a_1, a_2, \dots, a_n) = (a_{\pi(1)}', a_{\pi(2)}', \dots, a_{\pi(n)}')$$

Cette clé peut-être librement diffusée à tous ses correspondants potentiels ou stockée dans un annuaire comparable à ceux utilisés pour les numéros de téléphone.

La clé privée est composée de *M*, *W*, $(b_1, b_2, \dots, b_n)$ ainsi que de la permutation π, cette clé privée doit impérativement être gardée confidentielle et ne doit être fournie à personne car elle n'est pas nécessaire pour pouvoir chiffrer un

message. Elle est par contre indispensable (si le système est sûr) afin de pouvoir déchiffrer un message.

Au contraire, si la suite des poids n'est pas super-croissante, le seul algorithme connu consiste à essayer successivement toutes les solutions $(b_1, b_2, \dots, b_n)$ possibles. Si la suite est suffisamment longue, c'est un algorithme impraticable.

3.2.2 Chiffrement d'un message

- Le message à chiffrer est écrit en binaire sous la forme $m_1\ m_2 \dots\ m_n$, avec $m_i \in \{0,1\}$ (si le message est trop long, il est coupé en blocs de *n* bits au plus)
- Calculer le chiffré C tel que :

$$C = \sum_{i=1}^{n} m_i . a_i$$

- Transmettre C

3.2.3 Déchiffrement d'un message chiffré

- Calculer $d = W^{-1}(c \bmod M)$ en utilisant l'algorithme d'Euclide étendu
- Calculer $\varepsilon_1, \varepsilon_2, \dots, \varepsilon_n$ tels que *d* est donné par :

$$d = \sum_{i=1}^{n} \varepsilon_i . b_i$$

- Il s'agit de résoudre un problème de sac à dos très simple grâce à la propriété b_i :

$$b_i > \sum_{j=1}^{i-1} b_j$$

- Calculer $m_i = \varepsilon_{\pi(i)}$ pour $i \in [1, n]$
- Le message déchiffré s'écrit, en binaire, sous la forme $m_1, m_2, \dots, m_n$.

3.2.4 Exemples numérique

3.2.4.1 Exemple 1

- Pour un *n* artificiellement petit, *n = 10*

- Choix des b_i : 4, 9, 30, 70, 185, 451, 1306, 3534, 6517, 17486
- Choix de M : 50349($> \sum_{i=1}^{n} b_i$)
- Choix de W : 36334 (premier avec M)
- Calcul des a_i : 44638, 24912, 32691, 25930, 25373, 23209, 23446, 14406, 47680, 32642
- Calcul de la permutation π : π(1)=8, π(2)=6, π(3)=7, π(4)=5, π(5)=2, π(6)=4, π(7)=10, π(8)=3, π(9)=1, π(10)=9
- La clé publique est : (14406, 23206, 23446, 25373, 24912, 25930, 32642, 32691, 44638, 47680)
- La clé privée est composée de M, W, $(b_1, b_2, \dots, b_{10})$ et de la permutation π
- Chiffrement de message (1,0,0,1,0,0,0,1,1,0) : c=14406+25373+32691+44638=117108
- Déchiffrement de c : application de l'algorithme d'Euclide étendu à W et M : 7864× W-5675× M=1 donc W^{-1}=7864 *mod* M, calcul de $d=W^{-1}\times c \bmod M$=7865× 117108 *mod* 50349=3753
- Résolution du problème de sac à dos avec les b_i et la valeur cible 3753 : 3534+185+ 30+4=3753=b_1+b_3+b_5+b_8, soit : (ε_1,ε_2,ε_3, ε_4, ε_5, ε_6, ε_7, ε_8, ε_9, ε_{10})= (1,0,1,0,1,0,0,1,0,0)
- Calcule de m_1=$\varepsilon_{\pi(1)}$= ε_8=1, m_2=$\varepsilon_{\pi(2)}$= ε_6=0, m_3=ε_7=0, m_4=ε_5=1, m_5=0, m_6=0, m_7=0, m_8=1, m_9=1, m_{10}=0
- On retrouve bien le message initial (1,0,0,1,0,0,0,1,1,0).

3.2.4.2 Exemple 2

1ère étape : Pour un n =9 *[5]*, Alice choisit S= (1, 3, 5, 11, 25, 53, 101, 205, 512), m = 960 et w = 143. L'inverse d de 143 *mod* 960 est 47.

2ème étape : Pour chaque élément a_i de S, Alice calcule $b_i = a_i \cdot e \bmod m$, ce qui donne (143, 429, 715, 613, 695, 859, 43, 515, 256). En ordonnant b_i, elle obtient la clé publique 'knapsack' S' = (43, 143, 256, 429, 515, 613, 695, 715, 859).

3ème étape : Pour exprimer le message "*RAS*" en code binaire, Bernard peut, par exemple, utiliser le code ASCII à 8 bits. *R* correspond à 01010010, *A* à 01000001 et *S* à 01010011. Le message à coder est *RAS* = 0101001 0010000 0101010 011.

Il le décompose en blocs de longueur L convenue (7 par exemple) et chiffre chacun des blocs :

0101001 se code 43 + 429 + 613 = 1085, 0010000 se code 515, 0101010 se code 143 + 429 + 613 = 1185, 011 → 0110000 et se code 515 + 613 = 1128.

Il transmet à Alice le message 108-515-1185-1128

4ème étape : Alice va déchiffrer ce message élément par élément, en calculant *M * d mod m* et en déterminant la solution du problème du sac à dos.

- ✓ 1085 * 47 *mod* 960 = 115 = 101+11+3 correspond à 0000001 + 0100000 + 0001000 = 0101001
- ✓ 515 * 47 *mod* 960 = 205 = 205 correspond à 0010000
- ✓ 1185 * 47 *mod* 960 = 15 = 11+3+1 correspond à 0100000 + 0001000 + 0000010 = 0101010
- ✓ 1128 * 47 *mod* 960 = 216 = 205+11 correspond à 001000 + 0100000 = 0110000

Alice retrouve le message: 0101001001000001010100110000: *RAS*.

3.2.4.3 Exemple 3 (chiffrement d'un texte)

1. *n = 9*
2. Choisissez une suite [6] super-croissante *S* contenant au moins neuf éléments (séparez vos nombres par des virgules) : 2, 5, 9, 21, 45, 103, 215, 450, 946
3. $\sum_i 1^n a_i = 1796$
4. Choisissez un nombre *M* supérieur à $(\sum_i 1^n a_i)$ et un nombre *W* premier avec *M* :

   M = 2003, W = 1289

5. Votre clé publique est :{436, 569, 575, 721, 1030, 1183, 1570, 1586, 1921}.
6. L'inverse de *W mod m* est : 317
7. Choisissez la longueur *L* des blocs de chiffrement, L inférieure ou égale à 9

   L = 5, L = 8

Texte en clair :*"Le premier crypto-système à clef publique, qui fut proposé par Ralph Merkle et Martin Hellman en 1978, est basé sur le problème du sac à dos (Knapsackproblem en anglais). Il n'est plus utilisé actuellement puisque ce chiffre,*

ainsi que de nombreuses variantes, a été cassé au début des années 80 par Adi Shamir. "

Texte Chiffré avec L= 5 :"1157,1466,1599,1599,0,
2326,1005,1599,1296,2041,2174,2174,1599,2187,1726,1599,575,436,1466,2610,575,2895,1726,1030
,1865,1466,2610,2610,1144,2895,1726,2035,1865,2035,1605,1144,2320,2187,1157,0,2326,0,1030,11
44,1144,2756,1005,1011,1296,1751,1030,1580,0,2762,1726,569,1732,1466,1605,2610,569,2762,172
6,1011,1011,1030,1030,1580,569,2762,1726,1157,575,436,2035,1580,1290,2762,436,0,1865,436,218
7,1144,2895,2326,1005,2301,1865,2301,1605,1599,0,2326,1005,436,1865,721,1030,1011,575,1751,1
726,1296,1865,436,1605,569,0,2035,1726,1011,1865,1157,2041,2174,1751,2187,1157,0,1296,2041,2
320,569,0,2035,1726,436,1865,1157,2320,1144,1599,2756,1466,0,1157,436,1599,2174,1751,2756,10
05,1732,1296,1011,2610,569,0,2187,1726,1865,575,0,1751,2035,1599,1732,2187,1751,1011,1030,10
30,1144,1290,2326,2756,1605,575,436,1466,1144,569,2326,3331,1157,575,436,2187,2610,1290,232
6,1466,0,1732,1466,1599,1599,0,2326,1005,1599,1732,2762,1466,1144,2187,2320,1005,1732,1296,1
605,1030,1144,721,2762,1157,0,1865,1732,1030,2174,1144,721,1580,0,575,436,1599,1144,2895,232
6,2187,0,1011,436,1011,2174,2326,1751,1726,1030,1865,1732,1030,2174,1144,2320,2187,0,1865,43
6,2187,1144,2895,1751,2035,1296,1296,2041,2174,1599,0,2187,1726,1865,575,436,1030,2174,2326,
2187,2756,1296,1296,1011,1605,2610,1144,1290,1157,1865,575,436,575,2174,1751,721,1005,1865,
575,2762,1599,2610,1144,2762,436,0,1865,436,2174,1580,1290,2326,2187,0,1865,2041,2320,1144,1
599,2756,1005,1157,1865,2301,1605,1599,0,1751,1726,1005,1865,1466,2320,2174,1290,2756,1005,
1296,1296,2041,2174,2174,1290,2756,2035,1605,575,436,1751,1580,1290,2320,1726,2035,1865,101
1,2320,2174,1290,721,1005,1005,1296,1605,1030,1144,1144,2320,1005,1157,1296,2187,2035,1580,
575,2187,1157,1296,575,436,1030,2174,1599,2756,2035,2035,1732,575,1030,1580,569,2762,1726,1
011,575,436,1599,1144,1290,721,1005,1865,1732,2762,2174,2174,575,2326,2035,1011,1865,2041,2
187,2174,1290,2326,2187,0,1865,2187,1030,2610,575,2320,1726,436,1732,2187,2320,1144,1290,23
26,2187,1296,575,436,1030,1599,436,2320,1726,1605,2762,575,1030,1144,1144,1751,1726,2035,18
65,2301,1605,1599,0,1751,1726,2041,575,436,1599,1865,1599,1751,2035,2041,1865,1030,1030,114
4,721,2187,1726,2035,575,436,1030,2174,2326,2756,2610,1157,1296,2041,2187,1599,0,1865,436,10
30,575,436,1751,1144,569,2326,1466,0,721,1011,1599,1144,1599,721,569,2035,1732,436,1030,2174
,2320,2320,1726,1599,1011,175. ".

Texte Chiffré avec L= 8 :"2866,3764,1183,3783,
4352,3764,4485,3910,3764,4352,1183,3758,4352,4940,3783,4358,5054,4788,4940,4788,4358,5060,4
485,3764,1183,4339,1183,3758,4049,3764,3897,1183,3783,4794,3322,4049,3910,4219,4794,3764,24
79,1183,4219,4794,3910,1183,3897,4794,4358,1183,3783,4352,5054,3783,5054,4788,5496,1183,378
3,3189,4352,1183,3169,3189,4049,3783,3474,1183,3302,3764,4352,4479,4049,3764,1183,3764,4358
,1183,3302,3189,4352,4358,3910,4618,1183,2291,3764,4049,4049,4485,3189,4618,1183,3764,4618,
1183,2649,3370,3793,2934,2479,1183,3764,4788,4358,1183,3322,3189,4788,5496,1183,4788,4794,4
352,1183,4049,3764,1183,3783,4352,5054,3322,4049,5060,4485,3764,1183,3328,4794,1183,4788,31
89,3758,1183,4339,1183,3328,5054,4788,1183,1904,3296,4618,3189,3783,4788,3189,3758,4479,118
3,3783,4352,5054,3322,4049,3764,4485,1183,3764,4618,1183,3189,4618,4333,4049,3189,3910,4788

,2340,3048,1183,2727,4049,1183,4618,2763,3764,4788,4358,1183,3783,4049,4794,4788,1183,4794,4358,3910,4049,3910,4788,5496,1183,3189,3758,4358,4794,3764,4049,4049,3764,4485,3764,4618,4358,1183,3783,4794,3910,4788,4219,4794,3764,1183,3758,3764,1183,3758,3474,3910,3897,3897,4352,3764,2479,1183,3189,3910,4618,4788,3910,1183,4219,4794,3764,1183,3328,3764,1183,4618,5054,4485,3322,4352,3764,4794,4788,3764,4788,1183,4927,3189,4352,3910,3189,4618,4358,3764,4788,2479,1183,3189,1183,5496,4358,5496,1183,3758,3189,4788,4788,5496,1183,3189,4794,1183,3328,5496,3322,4794,4358,1183,3328,3764,4788,1183,3189,4618,4618,5496,3764,4788,1183,2934,2213,1183,3783,3189,4352,1183,2006,3328,3910,1183,3605,3474,3189,4485,3910,4352,3048 ".

Plusieurs remarques s'imposent :

1. Si la longueur des blocs de chiffrement est égale à celle des caractères en code ASCII à 8 bits, chaque lettre sera codée par le même nombre. Le système est alors vulnérable à une attaque à l'aide d'une analyse de fréquence. Il convient donc de choisir des blocs de chiffrement de longueur inférieure à celle de la clé.
2. Seule la connaissance de la clé publique est nécessaire pour chiffrer un message. Par contre, il faut nécessairement disposer de l'ensemble des éléments de la clé privée pour pouvoir déchiffrer en utilisant l'algorithme proposé. Nous sommes donc bien dans un scénario de chiffrement à clé publique.
3. Retrouver le message clair associé à un chiffré nécessite la résolution d'un problème de sac à dos comme décrit précédemment, la sécurité du crypto-système repose donc en grande partie sur l'hypothèse que ce problème est impossible à résoudre sans connaître la clé privée.
4. Connaissant la clé privée, il est facile de recalculer la clé publique. Par contre, retrouver la clé privée, et notamment les b_i, à partir de la clé publique.

Dans ce chapitre, on essaie d'étudier les suites super croissantes sur la base du système de chiffrement à clé publique de Merkle-Helman, et de proposer des méthodes nouvelles pour la construction de telle suite.

3.3 Contribution 4 : Construction d'une Suite Super-Croissante par le biais de la Carte Logistique

La construction d'une suite super croissante d'un nombre N (par exemple $N=8$) d'éléments passe par les étapes suivantes :

1. Déroulement de la carte logistique pour avoir un paquet de *N (=8)* valeurs, il faut les leurs transformer en valeurs entières :

$$x_i = \left(x_{i_{LogisticMap}} * \text{scalaire}\right) \bmod(256)$$

2. Ordonner ces valeurs dans un ordre donné (Croissant/Décroissant) :

$$X_{ordonnée} = \{x_1,\ x_2, x_3, x_4, x_5,\ x_6,\ x_7, x_8\}$$

3. Calculer le vecteur S de la suite super croissante :

$$S\ = \{s_1,\ s_2, s_3, s_4, s_5,\ s_6,\ s_7, s_8\}$$

Tel que :

$$s_i = \sum_{j=8}^{i} s_j$$

Soient $x_0 = 0.01$ et r =3.9 des valeurs initiales, le déroulement de la carte logistique nous donne :

$X_{Logistic\ Map} = \{x_1,\ x_2, x_3, x_4, x_5,\ x_6,\ x_7, x_8\}$ = { 0.03861, 0.144765144, 0.482851968, 0.973853185, 0.09930632, 0.348833841, 0.88588029, 0.394275967 }

La transformation de ces valeurs, en valeurs entières avec le scalaire 10000 nous donne : 130 – 167 – 220 – 10 – 225 – 160 – 154 - 102

Le trie de ces valeurs et l'affichage suivant un ordre croissant par exemple, nous donne la suite suivante : 10 - 102 - 130 - 154 - 160 - 167 - 220 – 225

3.3.1 Première Méthode de génération proposée

On calcule la suite super croissante SC de la façon suivante :

La première valeur S de la suite SC prend la première valeur de la suite des x ordonnée traitée. La deuxième valeur S de la suite SC prend la somme des deux premières valeurs des x ordonnée traitée. La troisième valeur S de la suite SC prend la somme des trois premières valeurs des x ordonnée traitée, et ainsi de suite jusqu'à la dernière valeur S de la suite SC prend la somme de toutes les valeurs de la suite des x ordonnée traitée.

Mais chaque fois qu'on incrémente pour passer à une nouvelle valeur S de la suite SC, on doit la tester, pour qu'elle soit supérieure à la somme cumulée des précédentes valeurs S de la suite SC.

Pour satisfaire cette condition on continue à ajouter à cette valeur, sa valeur jumelée de la suite des x ordonnée traitée.

3.3.2 Application numérique

I) Méthode à valeurs des x décroissantes de départ :

$X_{ordonnée} = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8\}$= {225, 220, 167, 160, 154, 130, 102, 10}

1. La première valeur S de la suite SC, S1= 225.
2. La deuxième valeur S de la suite SC, S2=225+220, S2=445.
3. La troisième valeur S de la suite SC, S3=S2+167=612, on a : S1+S2=670 et S3=612, donc on ajoute à S3 la valeur 167 ce que nous donne S3=S2+ 167*2, S3= 779> S1+S2 donc on passe au calcul du S4.
4. La quatrième valeur S de la suite SC, S4=S3+160=939, on a : S1+S2+S3=1449, qui est supérieur à S4 donc on ajoute à S4 la valeur 160 jusqu'à obtenir une valeur supérieur à 1449 donc : S4=S3+ 160*5, S4= 1579 > S1+S2+S3, donc on passe au calcul de S5.
5. La cinquième valeur S de la suite SC, S5=S4+154=1733, on a : S1+S2+S3+S4= 3028, qui est supérieur à S5 donc on ajoute à S5 la valeur 154 jusqu'à obtenir une valeur supérieur à 3028, donc : S5=S4+154*10, S5= 3119> S1+S2+S3+S4, donc on passe au calcul du S6.
6. La sixième valeur S de la suite SC, S6=S5+130=1249, on a : S1+S2+S3+ S4+ S5= 6147, qui est supérieur à S6 donc on ajoute à S6 la valeur 130 jusqu'à obtenir une valeur supérieur à 6147, donc: S6=S5+130*24, S6= 6239> S1+S2+S3+S4+S5, donc on passe au calcul du S7.
7. La septième valeur S de la suite SC, S7=S6+102=6341, on a : S1+S2+S3+ S4+S5+S6= 12386, qui est supérieur à S7 donc on ajoute à S7 la valeur 102 jusqu'à obtenir une valeur supérieur à 12386, donc: S7=S6+102*61, S7= 12461 > S1+S2+S3+S4+S5 +S6, donc on passe au calcul du S8.
8. La huitième valeur S de la suite SC, S8= S7+ 10= 12471, on a : S1+S2+ S3+S4+S5+ S6+S7= 24847, qui est supérieur à S8 donc on ajoute à S8 la valeur 10 jusqu'on obtient une valeur supérieur à 24847, donc: S8=S7+10*1239, S8= 24851> S1+S2+S3+ S4+S5+S6+S7.

Alors la première suite super-croissante est :

225-445 - 779 - 1579 - 3119 - 6239 - 12461-24851

II) Méthode à valeurs des x croissantes de départ :

$X_{ordonnée} = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8\}$= {10, 102, 130, 154, 160, 167, 220, 225}

1. La première valeur S de la suite SC, S1= 10.
2. La deuxième valeur S de la suite SC, S2=10+102, S2=112.
3. La troisième valeur S de la suite SC, S3=112+ 130=242, on a : S1+S2=122 et S3=242, donc S3= 242> S1+S2 donc on passe au calcul du S4.
4. La quatrième valeur S de la suite SC, S4=242+154, S4=396, on a : S1+S2+S3=10+112+242= 364, qui est inférieur à S4 donc on passe au calcul de S5.
5. S5=396+160 =556, on a : S1+S2+S3+S4= 760, qui est supérieur à S5 donc on ajoute à S5 la valeur 160 jusqu'à obtenir une valeur supérieur à 760, donc : S5= 556 + 160*2= 876 > S1+S2+S3+S4 donc on passe à calculer S6.
6. La sixième valeur S de la suite SC, S6=876+167=1043, on a : S1+S2+S3+ S4+ S5= 1636, qui est supérieur à S6 donc on ajoute à S6 la valeur 167 jusqu'à obtenir une valeur supérieur à 1636, donc : S6=876+167*5, S6= 1711> S1+S2+S3+S4+S5, donc on passe au calcul du S7.
7. La septième valeur S de la suite SC, S7=1711+220=1931, on a : S1+S2+S3+ S4+S5+S6= 3347, qui est supérieur à S7 donc on ajoute à S7 la valeur 220 jusqu'à obtenir une valeur supérieur à 3347, donc : S7=1711+220*8, S7= 3471> S1+S2+S3+S4+S5+S6, donc on passe au calcul du S8.
8. La huitième valeur S de la suite SC, S8=3471+225=3696, on a : S1+S2+ S3 + S4+S5+ S6+S7= 6818, qui est supérieur à S8 donc on ajoute à S8 la valeur 225 jusqu'on obtient une valeur supérieur à 6818, donc : S8=3471+225*15, S8 = 6846 > S1+ S2+ S3+ S4+S5+S6+S7.

Alors la deuxième super-croissante est :

10 - 112 - 242 - 396 - 876 - 1711 - 3471 – 6846

Nous pouvons conclure par le biais de ces deux exemples de suite super-croissantes, que le trie de la suite des x suivant un ordre précis joue un rôle important à la construction d'une suite super-croissante.

Aussi, on prévoir d'utiliser la suite des valeurs issues de la carte logistique, telles qu'elles arrivent.

III) Méthode à valeurs des x telles qu'elle arrive de départ :

$X_{traitée} = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8\}$= {130, 167, 220, 10, 225, 160, 154, 102}

1. La première valeur S de la suite SC, S1= 130.
2. La deuxième valeur S de la suite SC, S2=130+167, S2=297.
3. La troisième valeur S de la suite SC, S3=297+ 220=517, on a : S1+S2=297 et S3=517, donc S3= 517> S1+S2 donc on passe au calcul du S4.
4. La quatrième valeur S de la suite SC, S4=517+10=527, on a : S1+S2+S3=130+297+517= 944, qui est supérieur à S4=527, on ajoute à S4 la valeur 10 jusqu'à obtenir une valeur supérieur à 944, S4=517+10*43 =947, donc : S4= 947 > S1+S2+S3+S4 donc on passe à calculer S5.
5. S5=947+225 =1172, on a : S1+S2+S3+S4= 130+297+396+947=1770, qui est supérieur à S5 donc on ajoute à S5 la valeur 225 jusqu'à obtenir une valeur supérieur à 1770, donc : S5= 1172 + 225*3= 1847 > S1+S2+S3+S4 donc on passe à calculer S6.
6. La sixième valeur S de la suite SC, S6=1847+160=2007, on a : S1+S2+S3+ S4+ S5= 1770+1847=3617, qui est supérieur à S6 donc on ajoute à S6 la valeur 160 jusqu'à obtenir une valeur supérieur à 3617, donc : S6=2007+160*11, S6= 3767> S1+S2+S3+S4+S5, donc on passe au calcul du S7.
7. La septième valeur S de la suite SC, S7=3767+154=3921, on a : S1+S2+S3+ S4+S5+S6= 3617+3767=7384, qui est supérieur à S7 donc on ajoute à S7 la valeur 154 jusqu'à obtenir une valeur supérieur à 7384, donc : S7=3921+154*23, S7= 7463> S1+S2+S3+S4+S5+S6, donc on passe au calcul du S8.
8. La huitième valeur S de la suite SC, S8=7463+102=7565, on a : S1+S2+ S3 + S4+S5+ S6+S7= 7384+7463=14847, qui est supérieur à S8 donc on ajoute à S8 la valeur 102 jusqu'on obtient une valeur supérieur à 14847, donc : S8=7565+102*72, S8 = 14909 > S1+ S2+ S3+ S4+S5+S6+S7.

Alors la troisième super-croissante est :

130-297-517-947-1847-3767-7463-14909

IV) Méthode à valeurs des x telles qu'elle arrive de départ, avec l'ajout de la valeur moyenne :

$X_{traitée} = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8\}$= {130, 167, 220, 10, 225, 160, 154, 102}

$$X_{moy} = \frac{1}{8} * \sum_{i=1}^{8} x_i = 146$$

1. La première valeur S de la suite SC, S1= 130.
2. La deuxième valeur S de la suite SC, S2=130+167, S2=297.
3. La troisième valeur S de la suite SC, S3=297+ 220=517, on a : S1+S2=297 et S3=517, donc S3= 517> S1+S2 donc on passe au calcul du S4.
4. La quatrième valeur S de la suite SC, S4=517+10=527, on a : S1+S2+S3=130+297+517= 944, qui est supérieur à S4=527, on ajoute à S4 la valeur 146 jusqu'à obtenir une valeur supérieur à 944, S4=527+146*3 =965, donc : S4= 965 > S1+S2+S3+S4 donc on passe à calculer S5.
5. S5=965+225 =1190, on a : S1+S2+S3+S4= 130+297+517+965=1909, qui est supérieur à S5 donc on ajoute à S5 la valeur 146 jusqu'à obtenir une valeur supérieur à 1788, donc : S5= 1190 + 146*5= 1920 > S1+S2+S3+S4 donc on passe à calculer S6.
6. La sixième valeur S de la suite SC, S6=1920+160=2080, on a : S1+S2+S3+ S4+ S5= 1909+1920=3829, qui est supérieur à S6 donc on ajoute à S6 la valeur 146 jusqu'à obtenir une valeur supérieur à 3829, donc : S6=2080+146*12, S6= 3832> S1+S2+S3+S4+S5, donc on passe au calcul du S7.
7. La septième valeur S de la suite SC, S7=3832+154=3986, on a : S1+S2+S3+ S4+S5+S6= 3829+3832=7661, qui est supérieur à S7 donc on ajoute à S7 la valeur 146 jusqu'à obtenir une valeur supérieur à 7661, donc : S7=3986+146*26, S7= 7782>S1+S2+S3+S4+S5+S6, donc on passe au calcul du S8.
8. La huitième valeur S de la suite SC, S8=7782+102=7884, on a : S1+S2+ S3 + S4+S5+ S6+S7= 7661+7782=15443, qui est supérieur à S8 donc on ajoute à S8 la valeur 146 jusqu'on obtient une valeur supérieur à 15443, donc : S8=7884+146*52, S8 = 15476 > S1+ S2+ S3+ S4+S5+S6+S7.

Alors la quatrième super-croissante est :

130-297-517-965-1920-3832-7782-15476

On peut modéliser ces résultats dans un tableau récapitulatif :

Ordre	Type de Méthode
00	Méthode à valeurs des x telles qu'elle arrive de départ, avec l'ajout de la valeur moyenne
01	Méthode à valeurs des x telles qu'elle arrive de départ, avec l'ajout de la valeur jumelée.
10	Méthode à valeurs des x décroissantes de départ, avec l'ajout de la valeur jumelée.
11	Méthode à valeurs des x croissantes de départ, avec l'ajout de la valeur jumelée.

Tableau.3.1 Type de Méthode

La méthode proposée, nous produit des éléments de la suite super-croissante d'une façon récurrente, et on ajoute la caractéristique aléatoire à ces derniers, on les utilise dans le système Merkle et Hellman précédemment décrit.

3.3.3 Conclusion

Nous avons transformé le crypto-système à clé publique de Merkle et Hellman en un algorithme à clé secrète, et à complexité équivalente.

La taille de la clé de chiffrement de l'espace est le nombre total de différentes valeurs qui peuvent être utilisés dans ce procédé de chiffrement. Dans l'algorithme proposé, le champ de clé secrète est fixé comme suit :

$$\mathrm{ST} = \{x_0, r, D, O, S, N\}.$$

Où x_0 et r, sont des nombres à doubles précision. D (valeur d'indice de départ de la carte logistique, dans les exemples précédent nous avons pris $D=1$) et S=scalaire et N=échantillon (c'est le nombre de la suite super-croissante) qui sont des constantes entières, O= ordre est un champ à deux bits (tableau 1). Si la précision de calcul de x_0, r, est 10^{16} et, $D \in [1, 1000]$, $S \in [1, 10000]$ et $N \in [1, 64]$.

Par conséquent, l'espace de clé est plus grand que 10^{16} x 10^{16} x 4 x 64 x 1000 x 10000, (avec $10^3 \approx 2^{10}$) dans ce cas on aura un champ de clé de l'ordre de 2^{140}est c'est énorme.

Donc, l'algorithme de chiffrement a un très grand espace clé pour résister à toutes sortes d'attaques par force brute.

3.4 Contribution 5 : Adaptation de la Suite de Fibonacci Généralisée à Coefficients Réels pour la Génération d'une Suite Super-Croissante

Dans ce travail, on essaie de proposer une méthode basée sur la suite de Fibonacci généralisée [8] [72] [73] avec des coefficients réels pour la construction d'une suite super-croissante, et de l'utiliser dans le système à clé publique de Merkle-Helman. La nouveauté dans cette approche est la transformation de cet algorithme, en un système à clé secrète, avec un gain en longueur de la clé de chiffrement.

3.4.1 Biographie de Fibonacci

Fibonacci est né à Pise en 1175, il a rejoigné très jeune son père à la colonie de Bejaia, en Algérie. Son vrai nom est Léonardo Pisano, ou Léonard de Pise. Fibonacci est un surnom qui vient de filius Bonacci qui veut dire fils de Bonacci (Bonacci signifie chanceux, de bonne fortune).

Bonacci est le premier grand mathématicien de l'ère chrétienne du monde occidental. C'est lui qui a introduit la numération décimale et l'écriture arabe des chiffres en Occident, en ramenant dans son livre Liber abaci, les connaissances acquises en Algérie (Bejaia) où travaillait son père.

3.4.2 Suite de Fibonacci

Enfermez un couple de lapins dans un enclos. Le premier mois de leur vie, ils n'ont pas d'enfants. Tous les mois suivants, ils enfantent un couple de lapins. Chaque couple né agit alors de la même façon : le mois suivant sa naissance, il ne donne pas d'enfants, mais chaque mois, ensuite, il enfante un couple. Et ainsi de suite. Le problème que le mathématicien italien pose est de savoir quel est le nombre de couples de lapins le n-ième mois? Ce nombre, que nous noterons un, conduit à la suite de Fibonacci. La suite de Fibonacci vérifie la relation de récurrence suivante :

$$U_{n+1} = U_n + U_{n-1}$$

En effet, le $(n+1)$-ème mois, tous les couples qui vivaient le mois précédent sont encore en vie, et les couples nés au moins deux mois avant (c'est-à-dire tous les couples vivant le mois $(n-1)$ enfantent un couple.

Cette relation de récurrence est initiée par les deux premiers termes, qui sont $u_0 = 0$, et $u_1 = 1$. Les premiers termes de la suite de Fibonacci sont :1,1,2,3,5,8,13,21,34,55,89,...Ce qui caractérise cette suite de nombres et la rend universelle, c'est le fait universel que, si on prend deux nombres consécutifs et qu'on divise le plus petit d'entre eux par le plus grand, on obtient toujours une valeur approchée du quotient 1/1.618, ou 0.618, c'est-à-dire le nombre d'or ($\varphi = \frac{1}{1.618} = 0.618$).

On rencontre parfois la définition de suites de Fibonacci généralisées. Ce sont les suites qui vérifient une relation de récurrence double du type :

$$U_{n+1} = a * U_n + b * U_{n-1}$$

3.4.2.1 Quotient de deux nombres successifs de Fibonacci

Si on calcule les valeurs des quotients $\frac{F_2}{F_1}$, $\frac{F_3}{F_2}$, $\frac{F_4}{F_3}$, $\frac{F_{10}}{F_9}$, .c'est-à-dire les quotients $\frac{F_{n+1}}{F_n}$, on remarque que l'on obtient des nombres de plus en plus proches les uns des autres (sans jamais être égaux !) et se rapprochent du nombre d'or.

On peut démontrer que la suite des quotients $\frac{F_{n+1}}{F_n}$ a pour limite le nombre d'or lorsque n tend vers l'infini.

En effet, la suite de Fibonacci définie par $F_1 = 1$, $F_2 = 1$ et $F_n = F_{n-1} + F_{n-2}$ pour n > 2, est une suite récurrente linéaire d'ordre 2.

Si on pose $F_n = a * q^n$, avec q>0 et a non-nul, et que l'on reporte dans l'égalité $F_n = F_{n-1} + F_{n-2}$

On obtient $a*q^n = a*q^{n-1} + a*q^{n-2}$ soit $q^2 = q + 1$ en simplifiant par $a*q^{n-2}$et q est la solution positive de l'équation (x^2 -x-1=0), c'est-à dire le nombre d'or $\varphi = \frac{1+\sqrt{5}}{2}$.

Une petite remarque : ceci ne dépend pas des premiers termes F_1 et F_2 de la suite.

3.4.2.2 Calcul général du nombre de Fibonacci de rang n

On a la formule suivante qui donne directement le $n^{ième}$ nombre de Fibonacci sans connaître les précédents. On y voit clairement apparaître le nombre d'or.

$$F_n = \frac{1}{\sqrt{5}}\left[\left(\frac{1+\sqrt{5}}{2}\right)^n - \left(\frac{1-\sqrt{5}}{2}\right)^n\right]$$

Remarque : le terme $\frac{1-\sqrt{5}}{2}$ est plus petit que 1 donc la partie $\left(\frac{1-\sqrt{5}}{2}\right)^n$ de la formule tend vers zéro quand n devient grand.Par conséquent, pour connaître F_n quand *n* est grand, il suffit de prendre la partie entière de $F_n \rightarrow \frac{1}{\sqrt{5}}\left(\frac{1+\sqrt{5}}{2}\right)^n$

3.4.2.3 Carré du nombre d'or

Pour calculer le carré du nombre d'or, il suffit de lui ajouter 1 :$\varphi^2=\varphi+1$

Pour calculer l'inverse du nombre d'or, il suffit de lui retrancher 1 :

$$\frac{1}{\varphi} = \varphi - 1$$

3.4.2.4 Puissances du nombre d'or

$\varphi^2= \varphi+1$

$\varphi^3=\varphi^2+\varphi= 2\varphi+1$

$\varphi^4=2\varphi^2+\varphi= 2\varphi+2+\varphi=3\varphi+2$

$\varphi^5=3\varphi^2+2\varphi=3\varphi+3+2\varphi=5\varphi+3$

$\varphi^6=5\varphi^2+3\varphi=5\varphi+5+3\varphi=8\varphi+5$

$\varphi^7= 13\varphi+8$

Que voit-on encore apparaître la suite de Fibonacci. Les puissances du nombre d'or s'expriment en fonction de phi et de 1 et les coefficients ne sont autres que les nombres de Fibonacci.

Pour obtenir une puissance du nombre d'or, il suffit de connaître les deux puissances précédentes et de les additionner, ce qui est exactement le procédé de construction de la suite de Fibonacci !

Le nom de suite de Fibonacci a été donné par l'arithméticien français Edouard Lucas en 1817, alors qu'il étudiait ce qu'on appelle aujourd'hui les "suites de Fibonacci généralisées" obtenues en changeant les deux premiers termes de la suite de Fibonacci et qui suivent le même procédé de construction.

On note (F_n) la suite de Fibonnacci définie par $F_n = F_{n-1} + F_{n-2}$, $F_0 = 1$ et $F_1 = 1$.
On note (L_n) la suite de Lucas définie par $L_n = L_{n-1} + L_{n-2}$; $L_0 = 1$ et $L_1 = 3$.La suite de ... Lucas ! (Elle commence par 1, 3, 4, 7, 11, 18, 29, 47, ...).Les suites de Fibonacci et de Lucas sont très liées.
On note (G_n) une suite de Fibonnacci "généralisée" définie par $G_n = G_{n-1} + G_{n-2}$sans préciser les valeurs de G_0et G_1.

3.4.3 Deuxième Méthode de Génération : Nouvelle construction de la suite super-croissante

L'utilisation d'un sac à dos dans un crypto-système pose un problème dans la génération de la clé de chiffrement [74] [75] [76] [77], vu le nombre important de données à utiliser.

Si A est une suite de 100 (N=100) caractères (nombres) ou plus, ce sera très difficile de générer cette suite (A) et d'entrer la clé car le calcul des deux données M et W dépend de la suite A. La clé se compose de la suite A, M, W et N.

Nous proposons une méthode pour générer une suite super-croissante (chaque élément est supérieur à la somme des éléments précédents) par le biais de la suite de Fibonacci généralisée.

Il ne faudra que 4 éléments pour pouvoir calculer la suite A pour n'importe quelle valeur de N, on prend deux valeurs initiaux de la suite A (A1, A2 valeurs de départ) avec deux autres facteurs réelles α, ß comme suit :

$$A_i = \lceil A_{i-2} * \alpha + A_{i-1} * ß \rceil \forall i \geq 3 \quad (7)$$

$\lceil . \rceil$ (Floor : partie entière de A_i)+1)

Condition :$A_1 < A_2$ Sinon la suite super-croissante ne serait pas possible.

Remarque :$\alpha \leq ß$

Cette modification nous a permis de gagner en champs de la clé : 4 valeurs pour A plus M, W et N, 7 entités au total par rapport à celui d'origine n+3 entités.

Exemples de la génération de la suite A

1) Pour ce premier exemple nous avons pris les 4 valeurs suivantes :

A_1	A_2	A	ß
1	3	1.1045	1.852

La première suite super-croissante est :

1 – 3 – 6- 14 – 32 – 74 – 172 – 400 – 930 - 2164

On peut vérifier facilement que cette suite est super-croissante.

2) Voici un second exemple tout en changent les deux coefficients départ

Pour cet exemple on a changé la valeur des deux coefficients α et ß

A_1	A_2	A	ß
1	3	1.0045	1.952

1 – 3 – 6 - 14 – 33 – 78 – 185 – 439 – 1042 – 2474.

On peut vérifier facilement que cette suite est super-croissante.

3) Nous allons maintenant faire un troisième exemple en mettant $\alpha > ß$

A_1	A_2	A	ß
1	3	1.952	1.0045

1 – 3 – 4 – 9 – 16 – 33 – 64 – 128 – 253 - 503

Cette suite n'est pas super-croissante car la remarque na pas était respecter $\alpha > ß$

4) Nous allons maintenant faire un quatrième exemple en mettant α =ß

A_1	A_2	A	ß
1	3	1.952	1.952

1 – 3 – 7 – 19 – 50 – 134 – 359 – 962 – 2578 – 6910

Cette suite est super-croissante pour ces valeurs.

5) Nous allons maintenant faire un cinquième exemple en mettant A1 > A2

A_1	A_2	A	ß
3	1	1.0045	1.952

3 – 1 – 4 – 8 – 19 – 45 – 106 – 252 – 598 – 1420

Cette suite n'est super –croissante au début, mais après quelques itérations elle vérifie la condition de super-croissante.

6) Dans cette sixième exemple nous allons parler du taux d'écart entre les deux facteurs α et ß pour que la suite A soit super –croissante

Après plusieurs essaies nous avons réussie à obtenir cette suite super-croissante

A_1	A_2	A	ß
1	3	1.0045	1.5

1 – 3 – 5 – 10 – 20 – 40 – 80 – 160 – 320 – 640

Taux d'écart T entre α et ß :

$$T = \frac{\alpha}{\beta} = \frac{1.0045}{1.5} = 0.669$$

Déduction :

Si on prend ß > α pour que la suite A de n=10 soit super-croissante, il faudra que l'équation suivante soit réalisable :

$ß = \frac{\alpha}{T}$ Avec T < 66.9% (8)

7) Dans ce septième exemple nous allons choisir α et T d'une façon d'obtenir ß (8) avec T=52% pour que la suite A soit super –croissante.

A_1	A_2	A	ß
1	3	1.03	1.980

1 – 3 – 6 – 14 – 33 – 79 – 190 – 457 – 1100 – 2648

8) Nous allons changer ka valeur de T=72% pour que la suite A soit super – croissante.

A_1	A_2	A	ß
1	3	1.03	1.43

1 – 3 – 5 – 10 – 19 – 37 – 72 – 141 – 275 – 538

Cette suite n'est pas super croissante T=72% > 66.9% ne vérifié pas l'équation 8.

La méthode proposée, nous produit des éléments de la suite super-croissante d'une façon récurrente et, ajouté la caractéristique aléatoire à ces derniers, on les utilise dans le système Merkle et Hellman précédemment décrit.

3.4.4 Conclusion

Nous avons transformé le crypto-système à clé publique de Merkle et Hellman en un algorithme à clé secrète et, à complexité équivalente.

Dans l'algorithme proposé, la taille de la clé secrète de chiffrement, ainsi que le nombre de différentes valeurs qui peuvent être utilisés dans le procédé de cryptage sont fixées comme suit :

$$ST = \{\alpha, ß, A, B, D, N\}.$$

Où α et ß sont des nombres à doubles précision. D (valeur d'indice de départ de la suite de Fibonacci généralisée, dans les exemples précédent nous avons pris D=1) et N= (c'est le nombre des échantillons de la suite super-croissante), A (=A1) et B (=A2) sont des constantes entières.

Si la précision de calcul de α, ß, est 10^{-16}et, $A \in [1, 128]$, $B \in [1, 128]$, $D \in [1, 64]$ et $N \in [1, 64]$.

Par conséquent, l'espace de clé est plus grand que 10^{-16} x 10^{-16} x 128 x 128 x 64 x 64, (avec $10^3 \approx 2^{10}$) dans ce cas on aura un champ de clé de l'ordre de $\mathbf{2^{132}}$**(la longueur de clé est de 132 bits)** est c'est énorme.

3.5. Contribution 6 : Suite de Fibonacci Généralisée appliquée à la confidentialité des Données

Dans ce qui suit on adapte la suite de Fibonacci Généralisée pour chiffrer les Données

3.5.1 Génération des nombres aléatoires

Pour tester l'aléa des nombres générés on adopte le test des trois groupes de la moyenne, de la variance, du facteur d'autocorrélations.

Dans notre cas et pour une valeur de n donnée, soit x_i, pour i=1, ..., n, est la suite de Fibonacci. Pour trouver la suite de nombres aléatoires distribuée uniformément dans l'intervalle [0,1[, il faut prendre des u_i=x_i/M, pour i=1, ..., n avec :

$$\boldsymbol{X_n = (X_{n-1} + X_{n-2}) mod\ M\ ;}$$

On doit saisir les deux germes qui sont X_0 (=l) et X_1(= k); et à partir de ces deux dernier la suite est générer.

Cette méthode est un cas particulier des générateurs à congruence additive.

- Des différents résultats de la suite avec des différents paramètres (l, k) et u=1000 éléments, associé à des graphes représentant leur moyenne, leur variance et leur fonction d'auto-corrélation pour quelques suites de sortie aléatoire avec les différents paramètres (l , k).

  ➔ Pour l=2, k=3 :

5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 597, 584, 181, 765, 946, 711, 657, 368, 25, 393, 418, 811, 229, 40, 269, 309, 578, 887, 465, 352, 817, 169, 986, 155, 141, 296, 437, 733, 170, 903, 73, 976, 49, 25, 74, 99, 173, 272, 445, 717, 162, 879, 41, 920, 961, 881, 842, 723, 565, 288, 853, 141, 994, 135, 129, 264, 393, 657, 50, 707, 757, 464, 221, 685, 906, 591, 497, 88, 585, 673, 258, 931, 189, 120, 309, 429, 738, 167, 905, 72, 977, 49, 26, 75, 101, 176, 277, 453, 730, 183, 913, 96, 9, 105, 114, 219, 333, 552, 885, 437, 322, 759, 81, 840, 921, 761, 682, 443, 125, 568, 693, 261, 954, 215, 169, 384, 553, 937, 490, 427, 917, 344, 261, 605, 866, 471, 337, 808, 145, 953, 98, 51, 149, 200, 349, 549, 898, 447, 345, 792, 137, 929, 66, 995, 61, 56, 117, 173, 290, 463, 753, 216, 969, 185, 154, 339, 493, 832, 325, 157, 482, 639, 121, 760, 881, 641, 522, 163, 685, 848, 533, 381, 914, 295, 209, 504, 713, 217, 930, 147, 77, 224, 301, 525, 826, 351, 177, 528, 705, 233, 938, 171, 109, 280, 389, 669, 58, 727, 785, 512, 297, 809, 106, 915, 21, 936, 957, 893, 850, 743, 593, 336, 929, 265, 194, 459, 653, 112, 765, 877, 642, 519, 161, 680, 841, 521, 362, 883, 245, 128, 373, 501, 874, 375, 249, 624, 873, 497, 370, 867, 237, 104, 341, 445, 786, 231, 17, 248, 265, 513, 778, 291, 69, 360, 429, 789, 218, 7, 225, 232, 457, 689, 146, 835, 981, 816, 797, 613, 410, 23, 433, 456, 889, 345, 234, 579, 813, 392, 205, 597, 802, 399, 201, 600, 801, 401, 202, 603, 805, 408, 213, 621, 834, 455, 289, 744, 33, 777, 810, 587, 397, 984, 381, 365, 746, 111, 857, 968, 825, 793, 618, 411, 29, 440, 469, 909, 378, 287, 665, 952, 617, 569, 186, 755, 941, 696, 637, 333, 970, 303, 273, 576, 849, 425, 274, 699, 973, 672, 645, 317, 962, 279, 241, 520, 761, 281, 42, 323, 365, 688, 53, 741, 794, 535, 329, 864, 193, 57, 250, 307, 557, 864, 421, 285, 706, 991, 697, 688, 385, 73, 458, 531, 989, 520, 509, 29, 538, 567, 105, 672, 777, 449, 226, 675, 901, 576, 477, 53, 530, 583, 113, 696, 809, 505, 314, 819, 133, 952, 85, 37, 122, 159, 281, 440, 721, 161, 882, 43, 925, 968, 893, 861, 754, 615, 369, 984, 353, 337, 690, 27, 717, 744, 461, 205, 666, 871, 537, 408, 945, 353, 298, 651, 949, 600, 549, 149, 698, 847, 545, 392, 937, 329, 266, 595, 861, 456, 317, 773, 90, 863, 953, 816, 769, 585, 354, 939, 293, 232, 525, 757, 282, 39, 321, 360, 681, 41, 722, 763, 485, 248, 733, 981, 714, 695, 409, 104, 513, 617, 130, 747, 877, 624, 501, 125, 626, 751, 377, 128

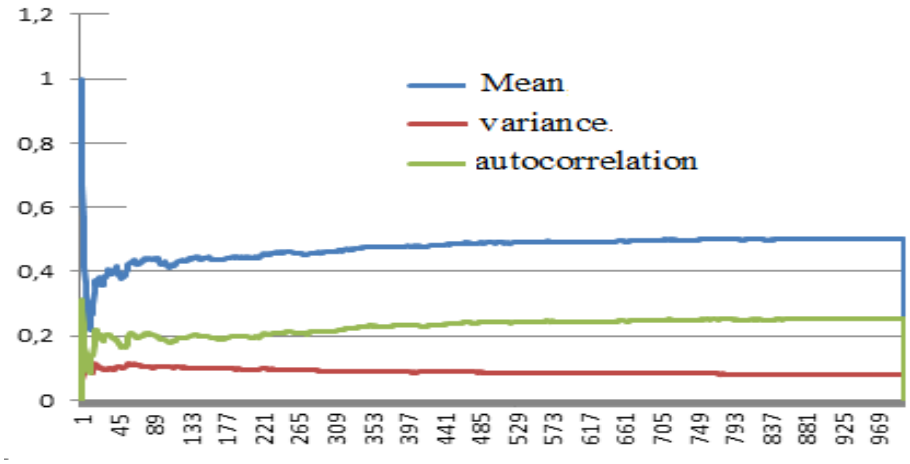


Figure 3.1. Test des Trois Graphes pour l=2, k=3

➔ Pour l=100, k=858 :

958, 816, 774, 590, 364, 954, 318, 272, 590, 862, 452, 314, 766, 80, 846, 926, 772, 698, 470, 168, 638, 806, 444
250, 694, 944, 638, 582, 220, 802, 22, 824, 846, 670, 516, 186, 702, 888, 590, 478, 68, 546, 614, 160, 774, 934
708, 642, 350, 992, 342, 334, 676, 10, 686, 696, 382, 78, 460, 538, 998, 536, 534, 70, 604, 674, 278, 952, 230
182, 412, 594, 6, 600, 606, 206, 812, 18, 830, 848, 678, 526, 204, 730, 934, 664, 598, 262, 860, 122, 982, 104
86, 190, 276, 466, 742, 208, 950, 158, 108, 266, 374, 640, 14, 654, 668, 322, 990, 312, 302, 614, 916, 530, 446
976, 422, 398, 820, 218, 38, 256, 294, 550, 844, 394, 238, 632, 870, 502, 372, 874, 246, 120, 366, 486, 852, 338
190, 528, 718, 246, 964, 210, 174, 384, 558, 942, 500, 442, 942, 384, 326, 710, 36, 746, 782, 528, 310, 838, 148
986, 134, 120, 254, 374, 628, 2, 630, 632, 262, 894, 156, 50, 206, 256, 462, 718, 180, 898, 78, 976, 54, 30, 84
114, 198, 312, 510, 822, 332, 154, 486, 640, 126, 766, 892, 658, 550, 208, 758, 966, 724, 690, 414, 104, 518
622, 140, 762, 902, 664, 566, 230, 796, 26, 822, 848, 670, 518, 188, 706, 894, 600, 494, 94, 588, 682, 270, 952
222, 174, 396, 570, 966, 536, 502, 38, 540, 578, 118, 696, 814, 510, 324, 834, 158, 992, 150, 142, 292, 434, 726
160, 886, 46, 932, 978, 910, 888, 798, 686, 484, 170, 654, 824, 478, 302, 780, 82, 862, 944, 806, 750, 556, 306
862, 168, 30, 198, 228, 426, 654, 80, 734, 814, 548, 362, 910, 272, 182, 454, 636, 90, 726, 816, 542, 358, 900
258, 158, 416, 574, 990, 564, 554, 118, 672, 790, 462, 252, 714, 966, 680, 646, 326, 972, 298, 270, 568, 838
406, 244, 650, 894, 544, 438, 982, 420, 402, 822, 224, 46, 270, 316, 586, 902, 488, 390, 878, 268, 146, 414, 560
974, 534, 508, 42, 550, 592, 142, 734, 876, 610, 486, 96, 582, 678, 260, 938, 198, 136, 334, 470, 804, 274, 78
352, 430, 782, 212, 994, 206, 200, 406, 606, 12, 618, 630, 248, 878, 126, 4, 130, 134, 264, 398, 662, 60, 722
782, 504, 286, 790, 76, 866, 942, 808, 750, 558, 308, 866, 174, 40, 214, 254, 468, 722, 190, 912, 102, 14, 116
130, 246, 376, 622, 998, 620, 618, 238, 856, 94, 950, 44, 994, 38, 32, 70, 102, 172, 274, 446, 720, 166, 886, 52
938, 990, 928, 918, 846, 764, 610, 374, 984, 358, 342, 700, 42, 742, 784, 526, 310, 836, 146, 982, 128, 110, 238
348, 586, 934, 520, 454, 974, 428, 402, 830, 232, 62, 294, 356, 650, 6, 656, 662, 318, 980, 298, 278, 576, 854
430, 284, 714, 998, 712, 710, 422, 132, 554, 686, 240, 926, 166, 92, 258, 350, 608

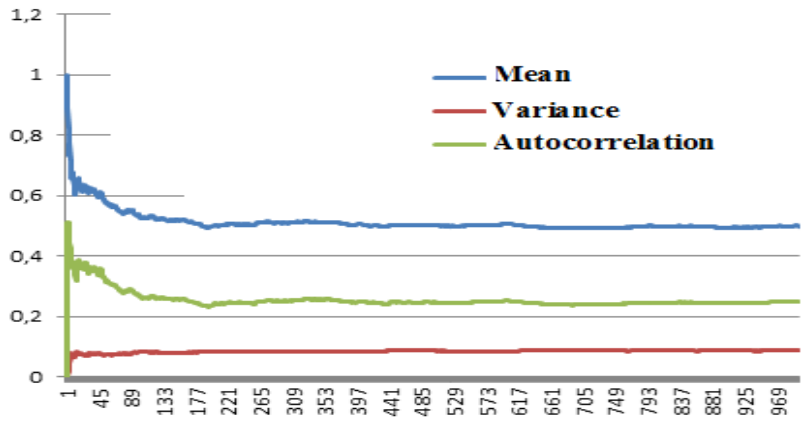


Figure 3.2. Test des Graphes pour l=100, k=858

Ces suites vérifient les conditions de l'aléa, donc on peut dire qu'elles sont aléatoires.

Passage au modulo256 de ces différentes suites :

On va modifier ces suites de nombres pseudo aléatoires, avec les mêmes séquences de Fibonacci, sauf quand va prendre les valeurs des suites (modulo 256) et les u_i=x_i/256, des différents résultats de la suite de fibonacci modifiée (modulo256) avec des différents paramètres (l, k) et u=500 éléments :

L'étude de l'aléatoire de ces suites modifiées (modulo256) :

On va étudier les suites modifiées avec les résultats de calculs par les graphes de moyenne, variance, et autocorrélation sous les mêmes conditions :

➔ Pour l=2 ; k=3 :

5, 8, 13, 21, 34, 55, 89, 144, 233, 121, 98, 219, 85, 72, 181, 253, 178, 199, 145, 112, 25, 137, 162, 43, 229, 40, 13, 53, 66, 119, 209, 96, 49, 169, 218, 155, 141, 40, 181, 221, 170, 135, 73, 208, 49, 25, 74, 99, 173, 16, 189, 205, 162, 111, 41, 152, 193, 113, 74, 211, 53, 32, 85, 141, 226, 135, 129, 8, 137, 145, 50, 195, 245, 208, 221, 173, 138, 79, 241, 88, 73, 161, 2, 163, 189, 120, 53, 173, 226, 167, 137, 72, 209, 49, 26, 75, 101, 176, 21, 197, 218, 183, 145, 96, 9, 105, 114, 219, 77, 40, 117, 181, 66, 247, 81, 72, 153, 249, 170, 187, 125, 56, 181, 5, 186, 215, 169, 128, 41, 169, 234, 171, 149, 88, 5, 93, 98, 215, 81, 40, 145, 185, 98, 51, 149, 200, 93, 37, 130, 191, 89, 24, 137, 161, 66, 227, 61, 56, 117, 173, 34, 207, 241, 216, 201, 185, 154, 83, 237, 64, 69, 157, 226, 127, 121, 248, 113, 129, 10, 163, 173, 80, 21, 125, 146, 39, 209, 248, 201, 217, 162, 147, 77, 224, 45, 13, 58, 95, 177, 16, 193, 233, 170, 171, 109, 24, 133, 157, 58, 215, 17, 0, 41, 41, 106, 147, 21, 168, 189, 125, 82, 231, 81, 80, 161, 9, 194, 203, 141, 112, 253, 109, 130, 7, 161, 168, 73, 9, 106, 115, 245, 128, 117, 245, 106, 119, 249, 112, 105, 241, 114, 99, 237, 104, 85, 189, 18, 231, 17, 248, 9, 1, 10, 35, 69, 104, 173, 21, 218, 7, 225, 232, 201, 177, 146, 67, 213, 48, 29, 101, 154, 23, 177, 200, 121, 89, 234, 67, 45, 136, 205, 85, 34, 143, 201, 88, 33, 145, 202, 91, 37, 152, 213, 109, 66, 199, 33, 232, 33, 9, 42, 75, 141, 216, 125, 109, 234, 111, 89, 200, 57, 25, 106, 155, 29, 184, 213, 141, 122, 31, 153, 184, 105, 57, 186, 243, 173, 184, 125, 77, 202, 47, 17, 64, 81, 169, 18, 187, 205, 160, 133, 61, 194, 23, 241, 8, 249, 25, 42, 67, 109, 176, 53, 229, 26, 23, 73, 96, 193, 57, 250, 51, 45, 96, 165, 29, 194, 223, 185, 176, 129, 73, 202, 19, 221, 8, 253, 29, 26, 55, 105, 160, 9, 193, 226, 163, 133, 64, 221, 53, 18, 71, 113, 184, 41, 249, 58, 51, 133, 184, 85, 37, 122, 159, 25, 184, 209, 161, 114, 43, 157, 200, 125, 93, 242, 103, 113, 216, 97, 81, 178, 27, 205, 232, 205, 205, 154, 103, 25, 152, 177, 97, 42, 139, 181, 88, 37, 149, 186, 79, 33, 136, 169, 73, 10, 83, 93, 200, 61, 5, 90, 95, 185, 48, 1, 73, 98, 171, 37, 232, 13, 245, 26, 39, 65, 104, 169, 41, 210, 251, 229, 248, 221, 213, 202, 183, 153, 104, 1, 105, 130, 235, 109, 112, 245, 125, 114, 239, 121, 128

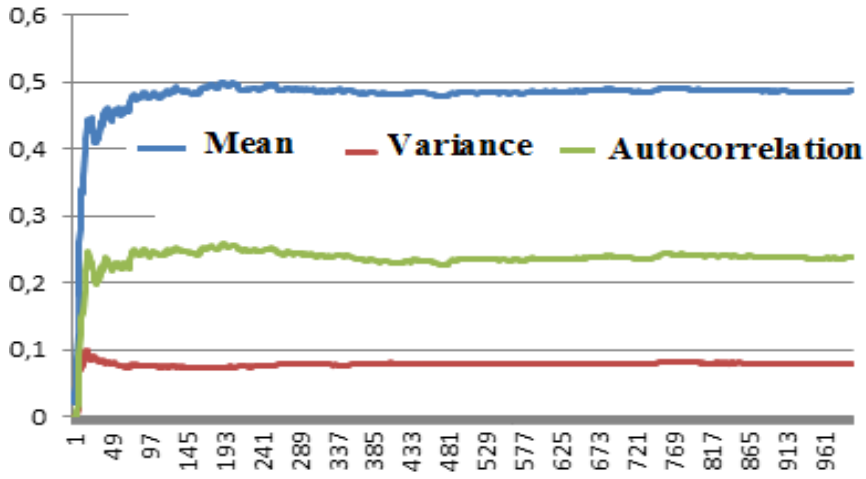


Figure 3.3. Test des Trois Graphes pour l=2, k=3

A partir des résultats de ces courbes, on constate que les conditions sont toujours réalisées avec les différents paramètres, alors les suites modifiées sont aléatoires.

➔ pour l=100, k=858 :

190, 48, 6, 78, 108, 186, 62, 16, 78, 94, 196, 58, 254, 80, 78, 158, 4, 186, 214, 168, 126, 38, 188, 250, 182, 176, 126, 70, 220, 34, 22, 56, 78, 158, 4, 186, 190, 120, 78, 222, 68, 34, 102, 160, 6, 166, 196, 130, 94, 224, 86, 78, 164, 10, 174, 184, 126, 78, 204, 26, 230, 24, 22, 70, 92, 162, 22, 184, 230, 182, 156, 82, 6, 88, 94, 206, 44, 18, 62, 80, 166, 14, 204, 218, 166, 152, 86, 6, 92, 122, 214, 104, 86, 190, 20, 210, 230, 208, 182, 158, 108, 10, 118, 128, 14, 142, 156, 66, 222, 56, 46, 102, 148, 18, 190, 208, 166, 142, 52, 218, 38, 0, 38, 38, 76, 138, 238, 120, 102, 246, 116, 106, 246, 120, 110, 230, 84, 82, 190, 16, 206, 246, 196, 210, 174, 128, 46, 174, 244, 186, 174, 128, 70, 198, 36, 234, 14, 16, 54, 70, 148, 218, 134, 120, 254, 118, 116, 2, 118, 120, 6, 126, 156, 50, 206, 0, 206, 206, 180, 130, 78, 208, 54, 30, 84, 114, 198, 56, 254, 54, 76, 154, 230, 128, 126, 254, 124, 146, 38, 208, 246, 198, 212, 178, 158, 104, 6, 110, 140, 250, 134, 152, 54, 230, 28, 26, 54, 80, 158, 6, 188, 194, 126, 88, 238, 94, 76, 170, 14, 184, 222, 174, 140, 58, 198, 24, 246, 38, 28, 66, 118, 184, 46, 254, 68, 66, 158, 224, 150, 142, 36, 178, 214, 160, 118, 46, 164, 210, 142, 120, 30, 174, 228, 170, 142, 56, 222, 46, 12, 82, 94, 176, 38, 238, 44, 50, 94, 168, 30, 198, 228, 170, 142, 80, 222, 46, 36, 106, 142, 16, 182, 198, 124, 90, 214, 48, 30, 102, 132, 2, 158, 160, 62, 222, 52, 42, 118, 160, 22, 206, 252, 202, 198, 168, 134, 70, 204, 42, 14, 56, 70, 150, 244, 138, 126, 32, 182, 214, 164, 146, 54, 224, 46, 14, 60, 74, 134, 232, 134, 110, 12, 146, 158, 48, 206, 22, 252, 42, 38, 80, 142, 222, 108, 98, 230, 96, 70, 166, 4, 170, 198, 136, 78, 214, 36, 18, 78, 96, 174, 14, 212, 226, 206, 200, 150, 94, 12, 106, 118, 248, 110, 126, 4, 130, 134, 8, 142, 150, 60, 210, 14, 248, 30, 22, 76, 98, 174, 40, 238, 46, 52, 98, 174, 40, 214, 254, 212, 210, 190, 144, 102, 14, 116, 130, 246, 120, 110, 230, 108, 106, 238, 88, 94, 182, 44, 226, 38, 32, 70, 102, 172, 18, 190, 208, 166, 118, 52, 170, 222, 160, 150, 78, 252, 98, 118, 216, 102, 86, 188, 42, 230, 16, 14, 54, 68, 146, 214, 128, 110, 238, 92, 74, 166, 8, 198, 206, 172, 146, 62, 232, 62, 38, 100, 138, 6, 144, 150, 62, 212, 42, 22, 64, 86, 174, 28, 202, 230, 200, 198, 166, 132, 42, 174, 240, 158, 166, 92, 2, 94, 96

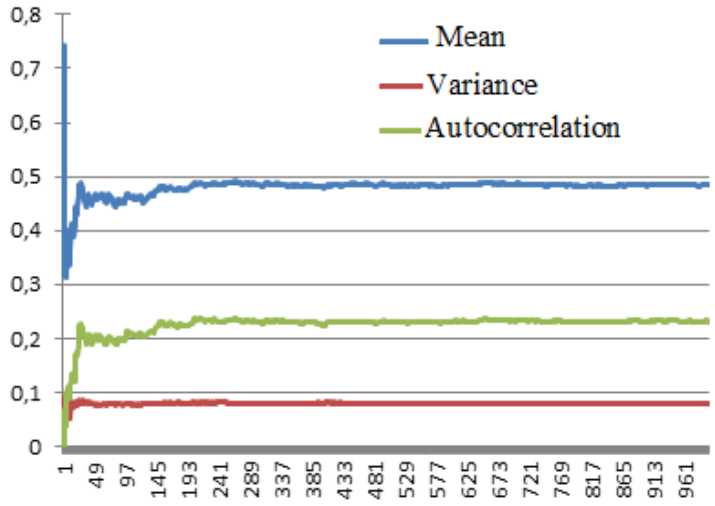


Figure 3.4. Test des Trois Graphes pour l=100, k=858

3.5.2 Chiffrement continue et Générateur pseudo Aléatoire :

Le système de chiffrement continue consiste à produire une suite chiffrant qui est le résultat de l'addition bit à bit du texte clair à la suite pseudo aléatoire (appelé codon). La réalisation la plus simple d'un algorithme de chiffrement en continu est illustrée par la figure 3.5.

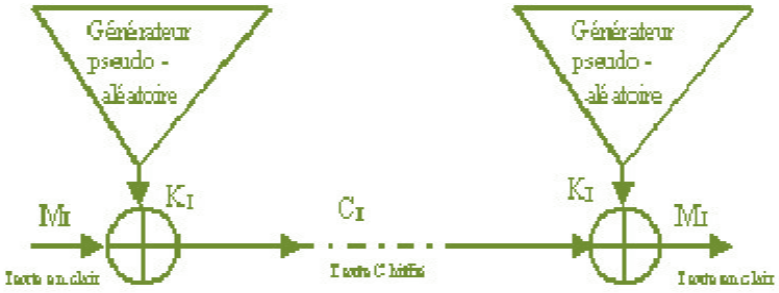


Figure.3.5 Chiffrement continu

Ce type de générateur engendre un flux de données aléatoires$K_1, K_2, K_3, \dots\dots\dots K_i$. Ce flux est combiné par une addition modulo(256) avec le flux de données du texte en clair $m_1, m_2, m_3, \dots\dots\dots m_i$: pour produire le flux de données chiffrés (équation 9).

$$C_i = m_i + K_i \; modulo \; (256) \; (9)$$

Du côté du déchiffrement, les données chiffrées sont combiné par une soustraction modulo 256 avec un flux identique de codons pour retrouver les données du texte en clair (équation 10).

$$m_i = C_i - K_i \; modulo \; (256) = ((m_i + K_i) - K_i \; mod \; 256) (10)$$

Le travail demandé est de réaliser cette étape :

- Les K_i est les valeurs Fibonacci modulés par 256.
- Le texte en clair, c'est-à-dire qu'on va chiffrer est m_i
- La clé de chiffrement c'est les valeurs de Fibonacci, K et L. on ajoute une autre valeur N qui est une valeur de départ de la suite de Fibonacci, par exemple N=13, c'est-à-dire qu'on commence à chiffrer à partir de la 13ème valeur de la suite de Fibonacci.

Donc on va prendre un texte qu'on transforme en ASCII, c'est-à-dire chaque caractère est converti en une valeur inférieure à 256. On va ajouter une valeur Fibonacci modulés le tout modulo 256.

Dans notre cas le générateur pseudo aléatoire est réaliser par la suite de Fibonnacci modifie avec comme clé ((L, K), N). On prend la clé chiffrement suivante :

- (L, K, M) paramètre de suite de Fibonnaci.
- (N) valeur de départ

3.5.3 Résultats et interprétations

3.5.3.1 Histogramme des Images

Pour une image monochrome, c'est-à-dire à une seule composante, l'histogramme est défini comme une fonction discrète qui associe à chaque valeur d'intensité le nombre de pixels prenant cette valeur. La détermination de l'histogramme est donc réalisée en comptant le nombre de pixel pour chaque intensité de l'image. L'histogramme peut alors être vu comme une densité de probabilité [51].

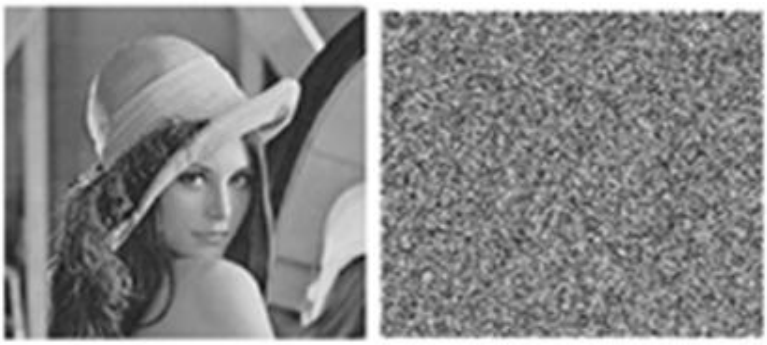


Figure.3.6.1 Image Lena.tif en claire et chiffrée

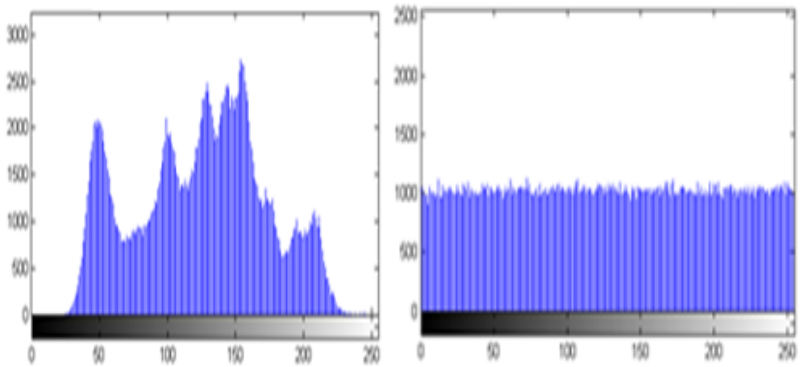


Figure.3.6.2 Histogrammes de l'Image Lena.tif en claire et chiffrée

Se référant aux résultats obtenus, nous pouvons clairement voir que l'image simple diffère sensiblement de celui correspondant cryptée. Par ailleurs, l'histogramme de l'image cryptée est assez uniforme ce qui rend difficile d'extraire les pixels nature statistique de l'image simple.

Les histogrammes des images claire et chiffrée de Lena montrant ainsi que le crypto système proposé fonctionne de façon correcte.

On constate que :

- Le chiffrage change la fréquence des pixels avec une distribution équiprobable pour toute l'image
- Les pixels sont très corrélés dans l'image en claire et que le chiffrage annule toute corrélation entre eux dans l'image chiffrée.
- L'image après chiffrage est devenue parasité et ne contenant aucune information visible qui se voit sur l'histogramme des deux images ne contient aucune information sur l'image en claire.

3.5.3.2 Corrélation entre deux pixels adjacents

Pour tester la corrélation entre deux pixels adjacents horizontalement, verticalement et diagonalement de l'image on calcule le coefficient de corrélation pour une séquence de pixels adjacents [51]. On va étudiant la distribution de 1000 pixels adjacents qui sont choisis aléatoirement dans l'image en claire et chiffré grâce à la fonction « randsrc ». Les figures qui suivants sont en nombre de trois de gauche à droite, comme suit : Corrélation des pixels horizontaux, Corrélation des pixels verticaux, Corrélation des pixels diagonaux.

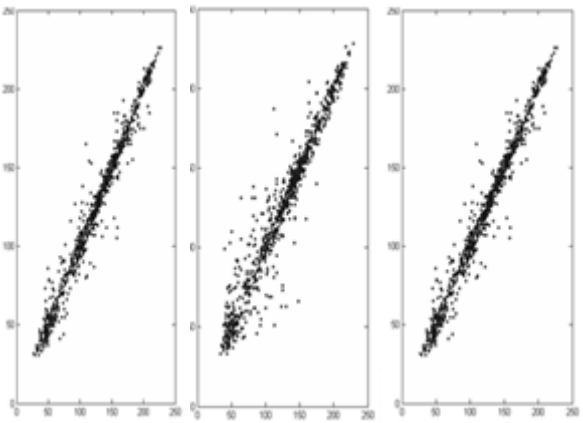


Figure.3.7.1 Distribution des pixels adjacents de l'image lena.png en claire

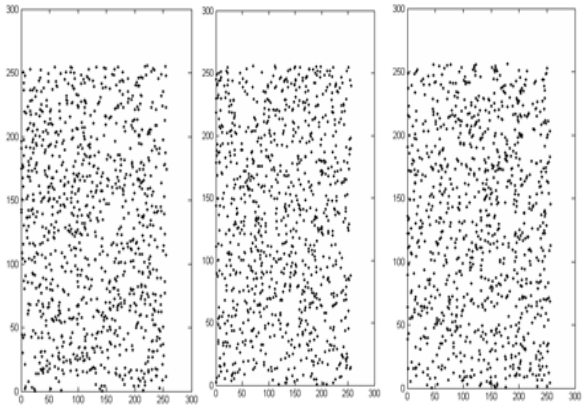


Figure.3.7.2 Distribution des pixels adjacents de l'image lena.png chiffrée

Picture	Coefficient de Corrélation de l'image en claire	Coefficient de corrélation de l'image chiffrée
lena.bmp	0.9719	0.0014
cameraman.bmp	0.9335	-0.0075
peppers.bmp	0.9913	-0.0042
coins.bmp	0.9749	-0.0036
football.bmp	0.9454	-0.0050
rice.bmp	0.9264	-0.0038
mandrill.bmp	0.8675	0.0052
house.bmp	0.9781	0.0017
clown.bmp	0.9711	0.0006
barbara.bmp	0.8954	-0.0017
boat.bmp	0.9381	-0.0040

Tableau.3.2 Comparaison des coefficients de corrélation entre les images en claire et chiffrée

On constate que :

- Les pixels adjacents sont très corrélés on que le cryptage créé un désordre très important.
- Les coefficients d'autocorrélation son proche de 1 pour les images en claire et le chiffrage l'annule qui prouve le bon fonctionnement de notre système.

3.5.3.3 Calcul de l'entropie :

La quantité d'information moyenne [51] associé à chaque symbole de la source sans mémoire est définie comme l'espérance mathématique (notée E{.}) de l'information propre fournie par l'observation de chacun des symboles possibles $\{S_1,...,S_n\}$.

Picture	Entropie de l'image en claire	Entropie de l'image chiffrée
lena.bmp	7.4455	7.9551
cameraman.bmp	7.0097	7.9415
peppers.bmp	6.9769	7.9465
coins.bmp	6.3071	7.9042
football.bmp	6.7058	7.9528
rice.bmp	7.0115	7.9512
mandrill.bmp	6.9010	7.9506
house.bmp	6.4971	7.9281
clown.bmp	7.3406	7.9542
barbara.bmp	7.6321	7.9547
boat.bmp	7.1912	7.9541

Tableau.3.3 Comparaison des Entropies entre les images en claire et chiffrée

On constate que l'entropie des images augmente jusqu'à presque atteindre 8 ce qui prouve que le cryptage crée un grand niveau de désordre.

3.5.4 Conclusion

Dans ce chapitre nous avons présenté des méthodes de construction des suites super-croissantes. Tous d'abord nous avons construire une suite super-croissante par le biais de la Carte Logistique. Puis nous avons adapté la suite de Fibonacci généralisée à coefficients réelles pour la génération d'une suite super-croissante. En fin nous avons utilisé cette suite pour la confidentialité des données.

CHAPITRE : 4
NEURONE CRYPTO-SYSTÈME

Chapitre 4

Neurone crypto-système

4.1 Introduction

Les réseaux de neurones artificiels [83] sont des modèles informatiques de réseaux d'automates dont la structure et le comportement sont "copiés" sur ceux des neurones réels. A la façon du cerveau, ils peuvent reconnaître des formes, réorganiser des données et, de façon plus intéressante, apprendre.

Plusieurs travaux de chiffrement sur la base des Réseaux de Neurones (RN) ont été proposés [80] [68][69], et on les trouve aussi associé pour résoudre un problème de cryptanalyse [81]. Dans ce qui suit on va essayer de les associer pour résoudre un problème de chiffrement.

4.2 Réseaux de neurone

4.2.1 Modèle Biologique d'une Cellule Neuronale

On décrit en figure.4.1, un modèle simple des neurones biologiques [83] qui a servi à la mise en place des premiers neurones. Dans le cerveau, les neurones sont reliés entre eux par l'intermédiaire d'axones et de dendrites. En première approche. On peut considérer que ces sortes de filaments sont conductrices d'électricité et peuvent ainsi véhiculer des messages depuis un neurone vers un autre. Les dendrites représentent les entrées du neurone et son axone sa sortie. Un neurone émet un signal en fonction des signaux qui lui proviennent des autres neurones. On observe en fait au niveau d'un neurone, une intégration des signaux reçus au cours du temps, c'est à dire une sorte de sommations des signaux. En général, quand la somme dépasse un certain seuil, le neurone émet à son tour un signal électrique.

La notion de synapse explique la transmission des signaux entre un axone et une dendrite. Au niveau de la jonction, il existe un espace vide à travers lequel le signal électrique ne peut pas se propager. La transmission se fait alors par l'intermédiaire de substances chimiques, les neuro-médiateurs. Quand un signal

arrive au niveau de la synapse, il provoque l'émission de neuro-médiateurs qui vont se fixer sur des récepteurs de l'autre côté de l'espace inter-synaptique. Quand suffisamment de molécules se sont fixées, un signal électrique est émis de l'autre côté et on a donc une transmission. En fait, suivant le type de la synapse, l'activité d'un neurone peut renforcer ou diminuer l'activité de ces voisins. On parle ainsi de synapse excitatrice ou inhibitrice.

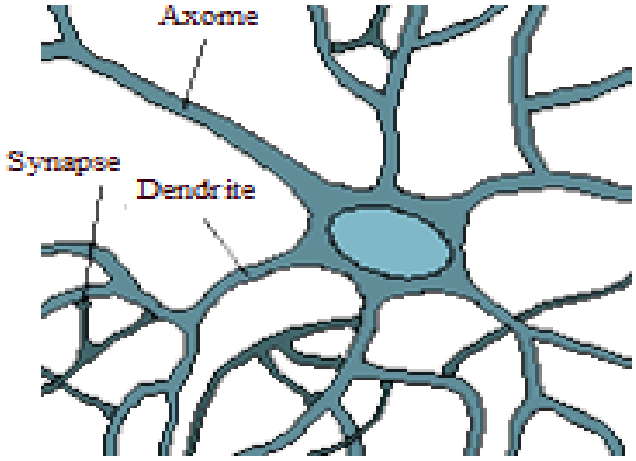


Figure.4.1 Neurone Biologique

4.2.2 Modèle d'un Neurone Formel

Les réseaux de neurones formels sont à l'origine d'une tentative de modélisation mathématique du cerveau humain [83]. Les premiers travaux datent de 1943 et sont l'œuvre de MM. Mac Culloch et Pitts. L'idée principale des réseaux de neurones "modernes" est la suivante: On se donne une unité simple, un neurone, qui est capable de réaliser quelques calculs élémentaires. On relie ensuite entre elles un nombre important de ces unités et on essaye de déterminer la puissance de calcul du réseau ainsi obtenu.

On constate tout d'abord, que le modèle biologique fait intervenir une notion temporelle qui est remplacée par une simple sommation des signaux arrivant au neurone (ces signaux sont communément appelés les entrées du neurone).

Dans notre cas (figure.4.2), on considère un neurone de 8 synapses excitatrices par les huit bits d'un pixel donné, en multipliant chaque synapse par un poids spécifique qui est calculé à travers un algorithme approprié. Cet algorithme utilise les données d'une carte logistique pour construire une suite super-croissante

de huit valeurs pour chaque neurone. En s'inspirant sur l'algorithme Merkel Helman, on fait naitre trois pixels chiffrés qui font constitues une image couleur de type RVB.

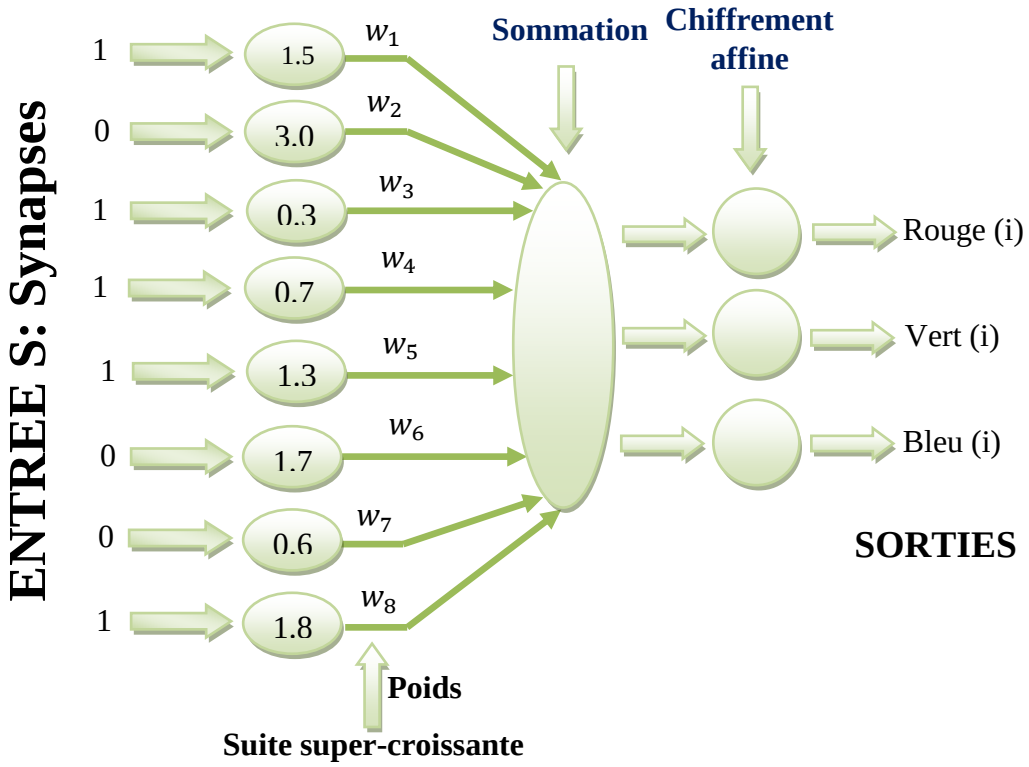


Figure.4.2 Neurone Formel proposé

4.2.3 Fonction d'Activation

Cette fonction permet de définir l'état interne du neurone en fonction de son entrée totale. Nous proposons dans ce travail d'utiliser la fonction affine. C'est l'une des fonctions d'activations les plus simples, sa fonction est définie par F(x) = a*x+d mod (256), avec a est premier avec 256.

4.3 Contribution 7 : Neurone Crypto-Système

On réalise un crypto-système inspiré d'une part du principe de crypto-système Merkel-Hellman avec une suite super-croissante et, d'autre part sur la base de la figure.4.2, avec comme hypothèses :

- Les entrées $\{M_i, i \geq 1...\}$ sont les pixels de l'image en clair.
- Synapses $\{W_i, i \geq 1\ldots\}$ sont des valeurs réelles positives issues de la carte logistique.
- Sommation $C = \sum W_i M_i$
- Sortie la valeur C associé au pixel de l'image chiffré.

- Pour une image monochromatique on la chiffre en une image couleur RGB.

4.3.1 Principe du Chiffrement

L'image en clair est chiffré pixel par pixel (pixel de 8 bits). Soit le pixel i, $i = (i_0, i_1, i_2, i_3, i_4, i_5, i_6, i_7)$, la valeur de $i \in \{0,1\}$.

Etape 1: On fait appel à un programme pour récupérer des données de la carte logistique et avec ses valeurs initiales prédéfinies. On prend les valeurs supérieures aux valeurs initiales et chaque fois pour trouver en final un paquet de huit valeurs, qu'on va les traitées, c'est-à-dire on va ordonner leur valeurs absolues de façon décroissante, on les multiplie par un scalaire ($K=10^8$) donné en s'assurant que les huit valeurs sont différentes comprises entre 0 et 255, sinon on déroule cet algorithme pour satisfaire cette dernière condition. On se rappelle de la dernière valeur trouvée pour qu'elle soit mise comme valeur initiale lorsqu'on fait appel à cet algorithme à nouveau.

Les valeurs initiales prédéfinies de la carte logistique, sont x(0), μ, N (indice de départ des valeurs de la carte logistique)

Soit x(j) la valeur de la carte logistique, sa valeur traitée sera P(i, j)= mod (floor(x(j)*K), 256), j=1, …8.

Exemple :

Soit pour un pixel i, le programme pour récupérer des données traitées et ordonnées de la carte logistique donne le résultat suivant : 10, 102, 130, 154, 160, 167, 220, 225

J	P(i,j)	J	P(i ,j)
1	10	5	160
2	102	6	167
3	130	7	220
4	154	8	225

Etape 2: Construire une suite super-croissante : Soit un pixel i, on initialise SCT(i)=0 et on calcul les synapses ou les huit valeurs (SC(i, j), j=1,...8) de la façon suivante :

J=1, la 1ère synapse prend la petite valeur traitée.

SC(i, 1)= P(i, 1) // $SCT(i,j) = SCT(i,j-1) + P(i,j)$

J=2, la 2ème synapse prend la somme de deux petites valeurs traitées.

SC(2)= P(i, 1) + P(i, 2) // $SCT(i,j) = SCT(i,j-1) + P(i,j)$

J=3, la 3ème synapse prend la somme des trois petites valeurs traitées,

SC(i, 3)= P(i, 1) + P(i, 2) +P(i, 3)

A partir de j=3, on calcul la valeur (SC(i, j)) comme suit :

$$SC(i,j) = SCT(i,j-1) + P(i,j)$$

Ensuite on calcul :

$$SCT(i,j) = SCT(i,j-1) + SC(i,j),$$

et ainsi de suite jusqu'à la 8ème synapse.

On aura vers la fin de l'étape 9 valeurs, les huit valeurs d'une suite supère-croissante, la 1ère valeur de synapse est inférieure strictement à la 2ème valeur de synapse qui est inférieure strictement à la 3ème valeur de synapse et ainsi de suite…plus leur somme SCT(i, 8).

Exemple : La suite construite par la carte logistique est : 10, 102, 130, 154, 160, 167, 220, 225

La première valeur SC(i, 1) de la suite est égale à : SC(i, 1)= 10 // SCT(i, 1)= 10

La deuxième valeur SC(i, 2) de la suite est égale à : SC(i, 2)=10+102=112 // SCT(i, 2)= 10+112=122

La troisième valeur SC(i, 3) de la suite est égale à : SC(i, 3)=10+102+ 130=242, // SCT(i, 3)= SCT(i, 2)+ SC(i, 3)=122+242= 364

La quatrième valeur SC(i, 4) de la suite est égale à : SC(i, 4)= SCT(i, 3)+154= 364+154= 518// SCT(i, 4)= SCT(i, 3)+ SC(i, 4)=364+518= 882

La cinquième valeur SC(i, 5) de la suite est égale à : SC(i, 5)= SCT(i, 4)+160= 882+160= 1042 // SCT(i, 5)= SCT(i, 4)+ SC(i, 5)=882+1042= 1924

La sixième valeur SC(i, 6) de la suite est égale à : SC(i, 6)= SCT(i, 5)+167= 1924+167= 2091// SCT(i, 6)= SCT(i, 5)+ SC(i, 6)=1924+2091= 4015

La septième valeur SC(i, 7) de la suite est égale à : SC(i, 7)= SCT(i, 6)+220= 4015+220= 4235, // SCT(i, 7)= SCT(i, 6)+ SC(i, 7)=4015+4235= 8250

La huitième valeur SC(i, 8) de la suite est égale à : SC(i, 8)= SCT(i, 7)+225= 8250+225= 8475, // SCT(i, 8)= SCT(i, 7)+ SC(i, 8)=8250+8475= 16725.

Alors la suite super-croissante est :

j	SC(i,j)	J	SC(i ,j)
1	10	5	1042
2	112	6	2091
3	242	7	4235
4	518	8	8475

On calcule aussi leur somme :

SCT(i, 8)= 10 + 112 + 242+ 518 + 1042 + 2091 + 4235 + 8475=16725

Etape 3: sur la base de l'équation 5 : On associé à la 8ème synapse le 1^{ier} bit à gauche du pixel en clair, le 2ème bit de ce pixel on lui associé la 7ème synapse, ainsi de suite jusqu'au 8ème bit on lui associé la 1ère synapse. La sommation est effectuée selon la valeur des bits du pixel en clair, multipliée par les valeurs des synapses qui leur sont associées.

$$SY(i) = \sum_{j=1}^{8} SC(i,j) * i(i, 9-j)$$

i.e.:SY(i)=SC(i,1)*i(i,8)+SC(i,2)*i(i,7)+SC(i,3)*i(i,6)+SC(i,4)*i(i,5)+SC(i,5)*i(i,4)+ SC(5,i)*i(i,3)+SC(6,i)* i(i,2)+ SC(7,i)*i(i, 1)

Exemple : Si on a le pixel i= 197= i(i, j=1, …8)=(1, 1, 0, 0, 0, 1, 0,1)

SY(i)=10 *i_8+ 112 *i_7+ 242*i_6+ 518*i_5+ 1042*i_4+ 2091*i_3+ 4235*i_2+ 8475*i_1

SY(i)= 10 *1+~~112 *0~~+ 242*1+~~518*0~~+~~1042*0~~+~~2091*0~~+ 4235*1+ 8475*1

SY(i)=10+242+4235+8475= 12962

1. La valeur de cette somme est divisée par la somme de tous les valeurs des synapses du pixel. On calcul

$$C(i) = \frac{SY(i)}{SCT(i,8)} \leq 1$$

Dans notre exemple C(i)=0.775007474

A partir de cette étape, on construit la matrice couleur RVB chiffrée résultante, donc on aura 3 matrices (Cr(i) : Matrice de la couleur rouge, Cv(i) Matrice de la couleur verte, Cb(i) Matrice de la couleur bleu)

Si, on a C(i) = 1 dans le cas où i(j)=1, ∀j=0,..7, on aura Cr(i)=1, Cv(i)=0, Cb(i)=0. et Si C(i) <1, dans ce cas on fait le calcul suivant :

2. C(i) est multiplié par 256.

   On garde la partie entière de la valeur trouvée dans la 1ère matrice (qui représente l'espace rouge),

   SXr(i)=S(i)*256

   Dans notre exemple S(i)*256=0.775007474*256=198.401913

   $$Cr(i) = \lceil SXr(i) \rceil = floor(S(i) * 256)$$

   Donc, Cr(i)= 198

   La partie décimale (0.401913) on la multiplie par 256 cette dernière valeur on garde sa partie entière dans la 2ème matrice (qui représente l'espace vert) :

   CXv(i)=SXr(i)-Cr(i)

   $$Cv(i) = \lceil SXv(i) * 256 \rceil = \left\lceil \left((S(i) * 256) - Cr(i) \right) * 256 \right\rceil$$

   Cv(i)=floor(0.401913*256)=floor(102.889728), Cv(i)=102

   Et sa partie décimale (0.889728) on la multiplie par 256 et on garde que la partie entière dans la 3ème matrice (qui représente l'espace bleu).

   CXb(i)= (CXv(i)*256)-Cv(i)$Cb(i) = \lceil CXb(i) * 256 \rceil$

   Cb(i)= floor(0.889728*256)=floor(227.770368), Cb(i)=227

   Pour le pixel i =197 on le code avec (198, 102, 227)

3. Pour le pixel i =197 on le code avec (198, 102, 227), on utilise trois nombres premiers (r, v, b) avec 256 pour transmettre ces données à l'étape qui suit :

   Cr(i)= r*Cr(i)mod(256),

   Cv(i)= v*Cv(i)mod(256),

   Cb(i)= b*Cb(i)mod(256).

   Dans notre cas r=v=b=1.

4. Les trois valeurs trouvées (Cr(i), Cv(i), Cb(i)) seront calculées suivant le système affine(ax+d) mod 256. Avec a premier avec 256. Dans cas a=1, d=0.

5. On passe au chiffrement d'un autre pixel en suivant les mêmes démarches, mais la dernière valeur chaotique trouvée dans le précédent pixel est mise comme valeur initiale lorsqu'on fait appel à l'algorithme chaotique.

   On peut représenter la première matrice par les bleu, la deuxième matrice par les vert et la troisième matrice par les rouge, dans ce cas on à choisir parmi ces 6 possibilités.

Bits	Ordre	Bits	Ordre
000	RVB	100	RVB
001	RBV	101	BVR
010	VRB	110	BRV
011	VBR	111	RVB

4.3.2 Principe de Déchiffrement

Dans un premier temps, on suit les mêmes étapes comme celle du chiffrement pour trouver la séquence super-croissante pour un pixel i donné à déchiffrer, et que nous avons reçu trois pixels chiffrées (ir, iv, ib) que peut lui associer.

1. La valeur trouvée est calculée suivant le système affine(ax+d) mod 256.
   Le cryptogramme reçu est déchiffrée par le système affine (a^{-1}(y-d) mod 256),

   Cr(i)= r-1*Cr(i)mod(256),

   Cv(i)= v-1*Cv(i)mod(256),

   Cb(i)= b-1*Cb(i)mod(256),

2. le pixel crypté est récupéré à partir des matrices couleur RVB selon cette formule :

   SY(i)=((Cr(i)+(Cv(i)+Cb(i)/256)/256)/256)*SCT(i, 8)

Si on a un nombre décimal, on prend l'entier qui lui est supérieur

Sr(i)= Sr((i)+(((Sb(i)/256)+Sv(i))/256)

Dans notre cas, si on reçoit :

(Cr(i), Cv(i), Cb(i))= (198, 102, 227) et SCT(i, 8)= 16725

SY(i) =((198+(102+227/256)/256)/256)*16725=12961.9992, SY(i) =12962

A partir de ce moment on utilise le déchiffrement de Merkel Helman pour retrouver le pixel cherché.

4.3.3 Clé de Chiffrement du Neurone Crypto-Système

Le système cryptographique tel qu'il est présenté à un comportement d'un système à clé secrète. Presque les mêmes étapes sont utilisées dans le chiffrement et de déchiffrement.

La clé de chiffrement peut être partagée en champs comme suit :

- Les valeurs initiales prédéfinies de la carte logistique, qui sont x(0), μ, N (index de départ pour récolter de la carte logistique). Où x_0, μ sont des nombres à double précision (10^{16}), N prend des valeurs de 1 à 256.
- Le multiplicateur K, en simple précision (10^8)
- Les quatre nombres premiers (a, r, v, b) avec 256 plus la valeur d de déplacement. Les cinq valeurs prennent des valeurs de 1 à 256.
- Les trois bits, pour représenter RVB.

Donc on a :

$$10^{16} \text{ x } 10^{16} \text{ x } 2^8 \text{ x } 10^8 \text{ x } 2^8 \text{ x } 2^8 \text{ x } 2^8 \text{ x } 2^8 \text{ x } 2^8 \text{ x } 2^3 = 10^{40} \text{ x } 2^{51}.(10^3 \approx 2^{10}),$$

$$40 = 13*3+1,\ 10^{40}=(10^3)^{13}*10 \approx 2^{133}.$$

Donc on aura une clé de ce Neuronal Crypto-Système de : 133+51=184 bits.

4.4 Validation du Système Cryptographique

On travaille avec les images claire et chiffrée de Lena.

Soient $x_0 = 0.01$ et r=3.9 : des valeurs initiales, $K=10^8$, N=25, a=r=v=b=1, d=0, RVB=000,

4.4.1 Histogramme des Images

Pour une image monochrome, c'est-à-dire à une seule composante, l'histogramme est défini comme une fonction discrète qui associe à chaque valeur d'intensité le nombre de pixels prenant cette valeur. Se référant aux résultats obtenus, nous pouvons clairement voir que l'image simple diffère sensiblement de celui correspondant cryptée. Par ailleurs, l'histogramme de l'image cryptée est assez

uniforme ce qui rend difficile d'extraire les pixels nature statistique de l'image simple.

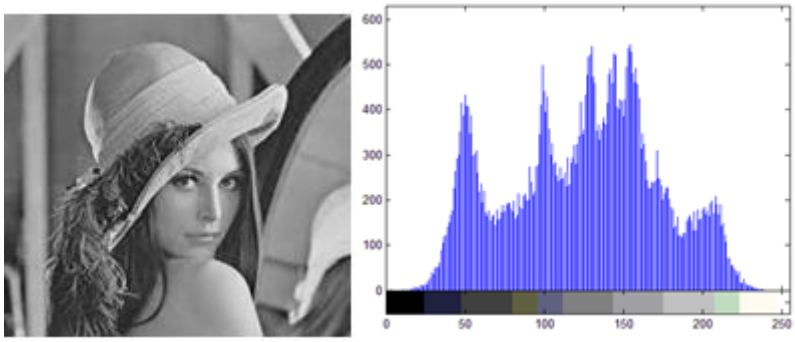


Figure4.3.1. Image Claire de Lena.bmp et son Histogramme

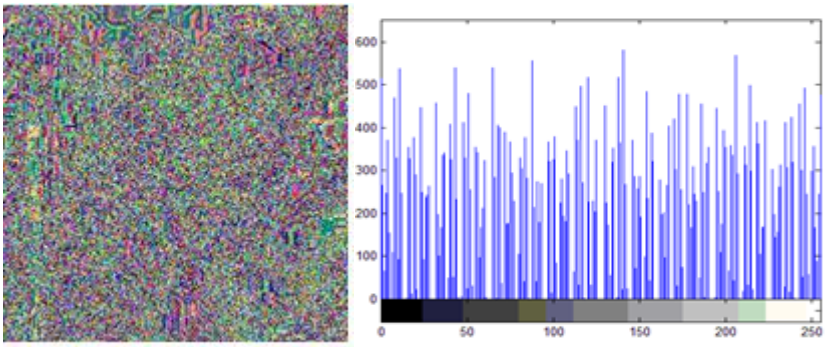


Figure.4.3.2 Image Chiffrée de Lena.bmp et son Histogramme dans l'espace rouge

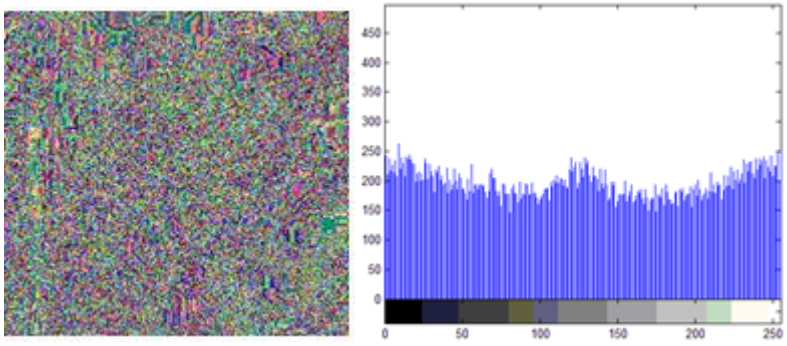


Figure.4.3.3 Image Chiffrée de Lena.bmp et son Histogramme dans l'espace vert

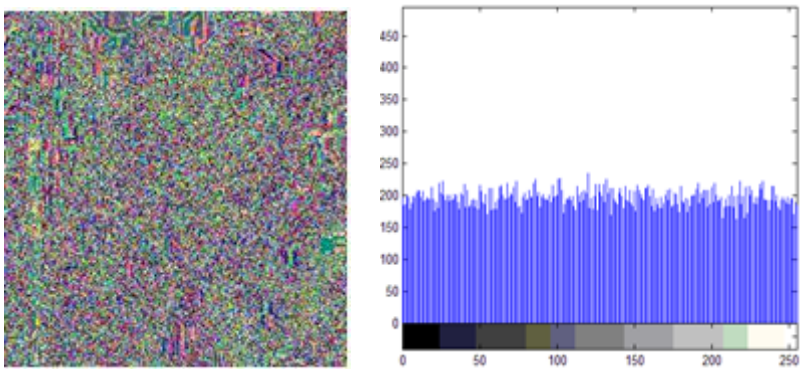


Figure.4.3.4 : Image Chiffrée de Lena.bmp et son Histogramme dans l'espace bleu

Les histogrammes des images claire (*Figure.4.3.1*) et chiffrée (*Figure.4.3.2, Figure.4.3.3, Figure.4.3.4*) de Lena montrant ainsi que le crypto-système proposé fonctionne de façon correcte.

4.4.2 Corrélation entre les Pixels Adjacents

En probabilités et en statistiques, étudier la corrélation entre deux variables aléatoires ou statistiques numériques, c'est étudier l'intensité de la liaison qui peut exister entre ces variables. La liaison recherchée est une relation affine, il s'agit de la régression linéaire.

Pour tester le coefficient de corrélation, nous avons choisi 1500 paires de deux pixels adjacents qui sont sélectionnés au hasard à la fois l'image claire et chiffrée.

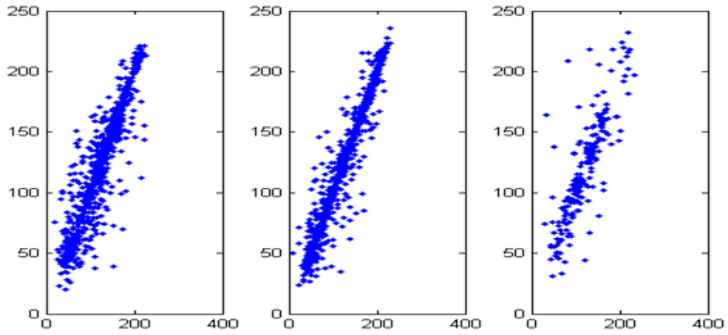


Figure.4.4.1 Corrélation entre les pixels horizontale verticale et diagonale adjacents: de l'image claire

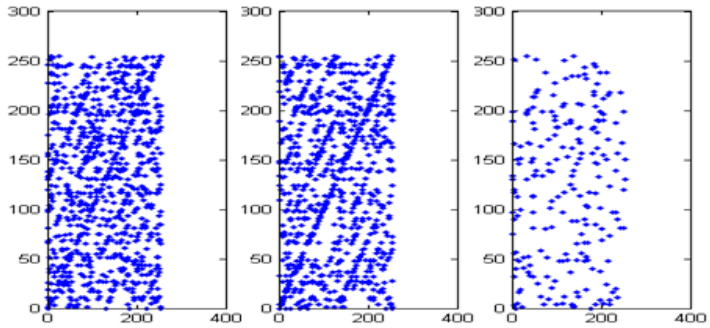


Figure.4.4.2 Corrélation entre les pixels horizontale, verticale et diagonale adjacents: de l'image cryptée à l'espace de couleur rouge

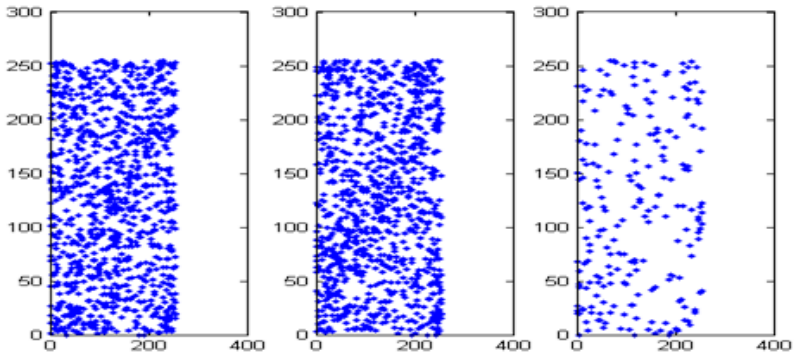


Figure.4.4.3 Corrélation entre les pixels horizontale, verticale et diagonale adjacents: de l'image cryptée à l'espace de couleur vert

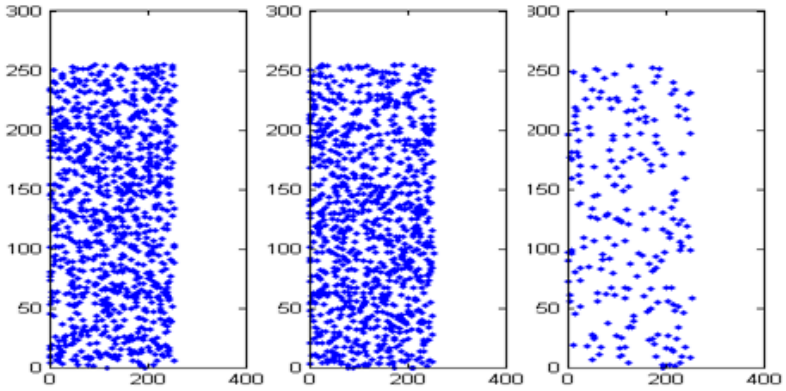


Figure.4.4.4 Corrélation entre les pixels horizontale, verticale et diagonale adjacents: de l'image cryptée à l'espace de couleur bleu

Nous voyons que les pixels voisins dans l'image claire ont une forte corrélation (coeff = 095247), tandis que dans le chiffrée il y'a un peu de corrélation (coeff =0.0037). Cette faible corrélation entre les pixels voisins dans l'image cryptée rend l'attaque de notre système cryptographique difficile.

Aussi, on voit clairement que dans l'image claire plusieurs droites peuvent s'ajuster à ce nuage de points mais parmi toutes ces droites on peut retenir celle qui jouit d'une propriété remarquable donnant lieu à une droite de la forme Y = aX + b représentant ainsi une corrélation linéaire.

4.4.3 Entropie

L'entropie de Shannon [51] est une fonction mathématique qui correspond intuitivement à la quantité d'information contenue ou délivrée par une source d'information. C'est l'information qui serait obtenue en moyenne en observant la sortie des symboles parallèles à partir d'un très grand nombre de sources identiques sans mémoire. Comme la source est stationnaire et sans mémoire, c'est aussi la moyenne des informations par symbole, qui serait obtenue en observant une série de très longs symboles émis par une source unique. Si, nous considérons une source qui a un alphabet de 256 caractères. Si tous ces caractères sont équiprobables, l'entropie associée à chaque caractère est $log_2(256) = log_2(2^8) = 8$ bits, ce qui signifie qu'il faut 8 bits pour transmettre un caractère, son entropie est égale à 8 bits.

L'entropie de différentes images à travers notre système cryptographique neurone nous donne :

Images	Texte en clair	Espace rouge	Espace vert	Espace bleu
Lena	7.4651	7.3338	7.9903	7.9968
Cameraman	7.0917	7.0124	7.9789	7.9941
Barbara	7.4189	7.3316	7.9906	7.9966

Tableau.4.1 Comparaison des Entropies entre les images en claire et chiffrée

Notre proposition de crypto-système et pour les deux espaces vert et bleu nous donne un rapport entre l'image cryptée et une source qui est délivre des caractères équiprobables est de 99,79%. Par conséquent, notre crypto-système proposé passe le test d'entropie.

4.5 Conclusion

A partir d'un modèle simple des neurones artificiels on a construit un système cryptographique. On sait que les connexions entre les neurones composant

le réseau décrivent la topologie du modèle. Elle peut être quelconque, mais le plus souvent il est possible de distinguer une certaine régularité (réseau à connexion complète).

Dans notre modèle nous avons utilisé la structure d'un réseau monocouche et, telle que des neurones organisés en entrée soient entièrement connectés à d'autres neurones organisés en sortie par une couche modifiable de poids par la sélection des valeurs aléatoires issus de la carte logistique.

Pour tester l'efficacité de ce crypto-système en plus de la visualisation de l'image chiffrée, on a calculé les trois indicateurs qui permettent d'estimer cet algorithme, à savoir :

1. Histogramme des images,
2. Corrélation entre deux pixels adjacents,
3. Entropie.

Tous les trois tests ont donné une bonne appréciation de ce crypto-système. De plus, ce crypto-système possède un très grand espace de clé pour résister à toutes sortes d'attaques par force brute.

La nouveauté de notre système était de chiffrer des images de niveaux de gris en des images en couleurs de type RVB.

CONCLUSION GÉNÉRALE

& PERSPECTIVES

Conclusion générale & Perspectives

1. Conclusion générale

La cryptographie reste toujours le moyen le plus performant d'obtenir une meilleur sécurité des données. Depuis longtemps l'humanité utilise cette technique pour assurer la confidentialité des messages, ils l'ont développé d'une façon simple mais efficace.

Cette technique réservée dans l'Antiquité aux univers de la guerre et de la diplomatique, peu à peu transformée en une partie d'une science de l'information. En effet de nos jours la cryptographie ne se limite plus à l'informatique mais elle est utilisée dans divers domaines scientifiques comme les mathématiques, l'automatique et l'électronique …

Il existe deux types de la cryptographie à base de clé : la cryptographie à clé secrète (chiffrement symétrique) et la cryptographie à clé publique (chiffrement asymétrique). A l'aide de cette clé on chiffre et on déchiffre l'information, c'est-à-dire la sécurité repose sur cette clé donc nous avons besoin des clés efficaces. La cryptographie chaotique nous répond à ce besoin.

La cryptographie basée sur la théorie du chaos est rapidement développée au cours de ces dernières années. Aujourd'hui la plupart des recherches se concentrent sur l'utilisation du chaos dans des crypto-systèmes en vue d'apporter une amélioration (temps de chiffrement, sécurité) par rapport aux méthodes standards de la cryptographie (DES, AES), ceci grâce aux caractéristiques des signaux chaotiques tels que : le déterminisme qui signifie que ces systèmes sont régis par des règles fondamentales non probabilistes, c'est-à-dire il est possible de reproduire le comportement chaotique. Une autre propriété intéressante de ces systèmes est la sensibilité aux conditions initiales, c'est-à-dire un petit changement ou une imprécision dans les conditions initiales engendre une des évolutions totalement différentes, ceci implique l'impossibilité de prédiction à long terme du comportement du système chaotique.

Cette thèse consiste à réaliser un crypto-système neuronal crypto-système basé sur un attracteur chaotique. Pour obtenir ce travail nous avons étudié un ensemble des cartes

chaotiques et les utilisées dans la cryptographie, et les exploitées pour construire des suites super-croissante et les implémentés dans la confidentialité des données.

Au premier lieu nous avons proposé un crypto-système basé sur la concaténation de la carte logistique et l'attracteur Hénon. Nous avons appliqué les tests de validation d'un crypto-système tel que l'histogramme des images, corrélation entre les pixels adjacents, et l'entropie. Tous les trois tests ont donné une bonne appréciation de ce crypto-système. De plus, ce crypto-système possède un très grand espace de clé (180 bits) pour résister à toutes sortes d'attaques par force brute. Nous avons observé que ce crypto-système est rapide et simple a implémenté et ne consomme que peu de ressource.

Ensuite, nous avons utilisé les mêmes systèmes chaotiques pour améliorer le chiffre Playfair, notre objectif est d'actualiser la sécurité de ce chiffre, nous avons modifié la dimension de tableau de 5x5 caractères à une dimension de 7x7 caractères, et nous avons substitué le mot clé ou la phrase à retenir, et nous l'avons remplacé par une clé de 180 bits. Cette augmentation dans l'espace de clé résiste à toutes sortes d'attaques par force brute.

Nous avons aussi exploité la carte PWLCM pour construire une suite aléatoire et l'intégrer au chiffre RC4, toujours dans l'objectif d'augmenter l'espace de clé qui implique la bonne sécurité.

Après nous avons construit une suite super-croissante par le biais de la carte logistique et l'utilisé pour transformer le crypto-système Merkel-Hellman en un algorithme à clé secrète et à complexité équivalente. Nous avons observé que l'espace de clé est plus grand que 10^{16} x 10^{16} x 4 x 64 x 1000 x 10000, (avec $10^3 \approx 2^{10}$) dans ce cas on aura un champ de clé de l'ordre de 2^{140} et c'est énorme, que implique que cette algorithme de chiffrement a un très grand espace de clé pour résister à toutes sortes d'attaques par force brute.

Nous avons adapté la suite de Fibonacci pour générer une suite super-croissante et l'utilisée aussi dans le crypto-système Merkel-Hellman, et nous avons constaté que l'espace de clé est plus grand que 10^{-16} x 10^{-16} x 128 x 128 x 64 x 64, (avec $10^3 \approx 2^{10}$) dans ce cas on aura un champ de clé de l'ordre de $\mathbf{2^{132}}$ **(** la longueur de clé est de 132 bits **)** et c'est énorme.

Et nous avons aussi appliqué cette suite de Fibonacci au chiffrement continu et pour tester l'efficacité de ce générateur, on a calculé les trois opérations : la moyenne, la variance,

et la fonction d'auto-corrélation, on a vérifié que les résultats obtenus se rapprochent le plus possible des résultats des cas idéaux.

En fin nous avons finalisé par une proposition d'un crypto-système qui exploite tous les travaux précédents, et qui est basé sur un modèle simple des neurones artificiels. Pour tester la validation de neuronal crypto-système nous avons appliqué les tests habituels tels que : l'histogramme des images, corrélation entre les pixels adjacents, et l'entropie, elles ont donne un bon résultat.

2. Perspectives

Le travail réalisé dans cette thèse ne crée pas une fin de recherche mais s'ouvre vers des contributions futures. Quelques idées sont citées ci-dessous :

- l'utilisation d'autre cartes chaotiques au lieu de la carte logistique, c'est-à-dire faire une recherche comparative entre ces cartes et choisir la meilleur pour ce travail.
- Générer ce crypto-système sur les autres types d'information tels que : des textes, de son, et encore des vidéos.
- L'introduction du calcul parallèle pour l'optimisation de la vitesse de l'opération de chiffrement et de déchiffrement. Ce qui nous permet de l'utiliser aux communications en temps réel.
- Notre crypto-système est un crypto-système sans perte qui nous offre la possibilité de l'utiliser pour les images intéressantes comme les images satellitaires ou les images médicales …

Bibliographie

[1] Prof. Jean-Marie Hannick, La cryptographie dans l'Antiquité gréco-romaine, janvier-juillet 2004

[2]Schneier, Bruce, « **Cryptographie appliquée** », International Thomson Publishing France, Paris, 1997.

[3] Stinson Douglas, « Cryptographie - Théorie et pratique », Vuibert Informatique, Paris, 2001.

[4] Adda ALI PACHA - Naima HADJ-SAID, La Cryptographie et ses principaux systèmes de références, RIST Vol, 12 n°01 Année 2002.

[5] Alexandru Spâtaru :" Fondements de la théorie de la transmission de l'information", Pesses Polytechniques romandes, 1987.

[6] J.DUBERTET, Initiation à la cryptographie, 2eme Edition VEBERT, Avril 2000.

[7] Marsault Xavier, Compression et cryptage des données multimédias, 2ème édition revue et augmentée, Editions Hermès, 1992.

[8] Bruce Schneier,"Applied Cryptography-Protocols, Algorithms and Source Code in C", John Wiley & Sounds, Inc, New York, Second Edition, 1996.

[9] Marion VIDEAU', "Critères de sécurité des algorithmes de chiffrement à clé secrète", L"université PARIS 2005.

[10] FIPS, Data Encryption. Standard (DES)(FIPS 46-3). csrc.nist.gov/publications/fips/ fips463/fips46-3.pdf ?,

[11] AllamMousa, Ahmad Hamad, Evaluation of the RC4 Algorithm for Data Encryption, International Journal of Computer Science & Applications Vol. 3, No.2, June 2006

[12] FIPS (Federal Information Processing Standard), Advanced Encryption Standard (AES) (FIPS PUB 197). 2. Category of Standard. Computer Security Standard, Cryptography. 3. Explanation.src.nist.gov/publications/fips-/fips197/fips-197.pdf, 26 nov. 2001.

[13] Christine Bachoc, "Cours de cryptographie symétrique", Master CSI Université Bordeaux I Année 2004-2005.

[14] W. Diffie and M. E. Hellman. "New directions in cryptography". IEEE Trans. Inform. Theory, IT-22 :644-654, Nov 1976.

[15] R. L. Rivest, A. Shamir, and L. M. Adleman. "A method for obtaining digital signatures and public-key cryptosystems". Communications of the ACM, 21(2) :120-126, 1978.

[16] A. Arazi, "Inegrating a key cryptosystem into the digital signature standard", Electron. Lett.,vol. 29, pp. 966-967, Nov. 1993.

[17] T. E. Gamal. "A public key cryptosystem and a signature scheme based on discrete logarithms". IEEE Trans. Inform. Theory, 31 :469-472, 1985.

[18] J. GLEICK "chaos theory" Albin Michel 1989.]

[19] Steven H. Strogatz, Nonlinear Systems and Chaos, Perseus publishing 1994.

[20] http://just.loic.free.fr/chaos/

[21] R. Chalabi, H. Hakim: “Study and implementation of a chaotic attractor with a view of their implementations to cryptography” PFE - USTO July 2006.

[22] L. Devaney, “A FIRST COURSE IN CHAOTIC DYNAMICAL SYSTEMS”, Now published by Westview Press, 1992.

[23] EtemadiBorujeni, S. and Ehsani, M.S. Modified Logistic Maps for Cryptographic Application. Applied Mathematics, 6, 773-782, 2015.

[24] Aman Jain, Namita Tiwari, Madhu Shandilya, “Image Based Encryption Techniques : A Review”, (IJCSIT) International Journal of Computer Science and Information Technologies, pp. 2886-2889, Vol. 5 (3) , 2014.

[25] Kavita Chaudhary Shiv Saxena, "A New Encryption Method Using Chaotic Logistic Map”, International Journal of Advanced Research in Computer Science and Software Engineering, pp.517-521, Volume 4, Issue 8, August 2014.

[26] HossamEldin H. Ahmed, Ayman H. Abd El-aziem, “Image Encryption Using Development of Chaotic Logistic Map Based on Feedback Stream Cipher”, pp. 274- 283, Recent Advances In Telecommunications, Informatics And Educational Technologies, December 2014.

[27] Ravindra K. Purwar& Priyanka, « An Improved Image Encryption Scheme Using Chaotic Logistic Maps”, International Journal of Latest Trends in Engineering and Technology (IJLTET), pp. 220-227, Vol. 2 Issue 3 May 2013.

[28] Dinka Pančić, “Verhulst logistic map in the study of nonlinear and chaotic systems”, pp. 198-203, Central European Conference on Information and Intelligent Systems, Varaždin, Croatia, September 18-20, 2013.

[29] S Rakesh, Ajitkumar A Kaller, B C Shadakshari and B Annappa, “Image Encryption using Block Based Uniform Scrambling and Chaotic Logistic Mapping”, DOI:10.5121/ijcis.2012.2105 49, International Journal on Cryptography and Information Security (IJCIS), pp. 49-57, Vol.2, No.1, March 2012.

[30] H. Ogras, M. Turk « Digital Image Encryption Scheme using Chaotic Sequences with a Nonlinear Function”, International Scholarly and Scientific Research & Innovation, pp. 459-462,Vol. 6 N°7. 2012.

[31] K. J. Persohn, R. J. Povinelli, “ Analyzing Logistic Map Pseudorandom Number Generators for Periodicity Induced by Finite Precision Floating-Point Representation”, submitted to Chaos, Solitons & Fractals, June 6, 2011.

[32] Mintu Philip, Asha Das « Survey: Image Encryption using Chaotic Cryptography Schemes”, IJCA Special Issue on “Computational Science - New Dimensions & Perspectives” NCCSE, 2011

[33] K. A. Ayanlowo, O. Folorunso, A. T. Akinwale and A. N. Njah, “ An improved chaotic encryption scheme”, African Journal of Mathematics and Computer Science Research Vol. 3(8), pp. 163- 172, August 2010.

[34] Donald E. Knuth, The Art of Computer Programming, Vol.2: Seminumerical Algorithms - chapitre 3: Random Numbers (Addison-Wesley, Boston, 1998)

[35] Louis Granger, “Simulation des systèmes à événements discrets », éditeur École polytechnique de Montréal, 2001.

[36] Hua Xue, Shubin Wang and XiandongMeng, Study on One Modified Chaotic System Based on Logistic Map, Research Journal of Applied Sciences, Engineering and Technology 5(3): 898-904, 2013 ISSN: 2040-7459; E-ISSN: 2040-7467 © Maxwell Scientific Organization, 2013

[37] Sun, Y. and G.Y. Wang, “An image encryption scheme based on modified logistic map”, The 4th International Workshop on Chaos-Fractals Theories and Applications (IWCFTA 2011), pp. 179-182, Hangzhou, Zhejiang, China, October 19-22, 2011

[38] Amsterdam Marotto F,” The dynamics of a discrete population model with threshold”. Math Biosci 58:123–128, 1982.

[39] Z.G. Xu, Q. Tian and L. Tian, Theorem to Generate Independently and Uniformly Distributed Chaotic Key Stream via Topologically Conjugated Maps of Tent Map, Mathematical Problems in Engineering 2012 (2012), Article ID: 619257.

[40] C. Li, S. Li, G. Alvarez, G. Chen and K. T. Lo. “Cryptanalysis of two chaotic encryption schemes based on circular bit shift and XOR operations”. Physics Letters A, 2007.

[41] R. Ranjith Kumar and M. Bala Kumar (2014), “A New Chaotic Image Encryption Using Parametric Switching Based Permutation and Diffusion”, ICTACT Journal on Image and Video Processing, May 2014, Volume: 04, Issue: 04, pages 795- 804.

[42] Long Bao, Yicong Zhou, C. L. Philip Chen, Hongli Liu (2012), “A New Chaotic System for Image Encryption”, 2012 International Conference on System Science and Engineering June 30-July 2, 2012, Dalian, China.

[43] G.A.Sathishkumar, .K.Bhoopathybagan and N.Sriraam (2011), “Image Encryption Based on Diffusion and Multiple Chaotic Maps”, International Journal of Network Security & Its Applications (IJNSA), Vol.3, No.2, DOI: 10.5121/ijnsa.2011.3214 181, March 2011

[44] Mintu Philip, Asha Das (2011), “Survey: Image Encryption using Chaotic Cryptography Schemes”, IJCA Special Issue on “Computational Science -New Dimensions & Perspectives” NCCSE, 2011.

[45] K. Sakthidasan@Sankaran and B. V. Santhosh Krishna (2011), “A New Chaotic Algorithm for Image Encryption and Decryption of Digital Color Images”, International Journal of Information and Education Technology, Vol. 1, No. 2, June 2011, pages 137-141.

[46] Noura, H. El Assad, S. Vladeanu (2010), “Design of a fast and robust chaos-based crypto-system for image encryption”, 8th International Conference on Communications (COMM), 2010, pages 423 – 426.

[47] Ai-hongZhu, Lia Li (2010), “Improving for Chaotic Image Encryption Algorithm Based on Logistic Map”, 2nd Conference on Environmental Science and Information Application Technology, 2010.

[48] Shubo Liu, Jing Sun, Zhengquan Xu (2009), “An Improved Image Encryption Algorithm based on Chaotic System”, journal of computers, vol. 4, no. 11, 2009, pp.1091-1100.

[49] Xiping He Qionghua Zhang (2008), “Image Encryption Based on Chaotic Modulation of Wavelet Coefficients”, Congress on IEEE Image and Signal Processing (CISP'08), Sanya, Hainan, Vol.1, 27-30 May 2008, pages 622-626.

[50] Xin Zhang, Weibin Chen (2008), “A New Chaotic Algorithm For Image Encryption”, IEEE ICALIP2008, pages 889 – 892.

[51] Guanrong Chen, Yaobin Mao, Charles K. Chui, “A symmetric image encryption scheme based on 3D chaotic cat maps”, doi:10.1016/j.chaos.2003.12.022, Chaos, Solitons and Fractals 21, pp: 749–761, 2004.

[52] Alam et al.: « Universal Playfair Cipher Using MXN Matrix », International Journal of Advanced Computer Science, Vol. 1, No. 3, Pp. 113-117, Sep. 2011.

[53] A. AftabAlam, B. Shah Khalid, and C. Muhammad Salam, « A Modified Version of Playfair Cipher Using 7×4 Matrix », International Journal ofComputer Theory and Engineering, Vol. 5, No. 4, August 2013.

[54] Gaurav Shrivastava, ManojChouhan, ManojDhawan, A Modified Version Of Extended Plafair Cipher (8x8), International Journal Of Engineering And Computer Science, ISSN:2319-7242, Volume 2 Issue 4, Page No. 956 -961, April, 2013.

[55] Nisarga Chand, Subhajit Bhattacharyya, « A Novel Approach for Encryption of Text Messages Using PLAY-FAIR Cipher 6 by 6 Matrix with Four Iteration Steps », International Journal of Engineering Science and Innovative Technology (IJESIT), Volume 3, Issue 1, January 2014.

[56] JitendraChoudhary, Ravindra Kumar Gupta, Shailendra Singh, « A GENERALIZED VERSION OF PLAY FAIR CIPHER », COMPUSOFT, An international journal of advanced computer technology, ISSN:2320-0790, (Volume-II, Issue-VI), June-2013.

[57] Ashish Negi, Jayveer Singh Farswan, V.M Thakkar, SiddharthGhansala, « Cryptography Playfair Cipher using Linear Feedback Shift Register », IOSR Journal of Engineering, Vol 2 N° 5 pp, 1 212-1216, May. 2012.

[58] Vinod Kumar,SantoshkrUpadhyay, Satyam Kishore Mishra, Devesh Singh, « Modified Version of Playfair Cipher Using Linear Feedback Shift Register and Transpose Matrix Concept », International Journal of Innovative Technology and Exploring Engineering (IJITEE), ISSN: 2278-3075, Volume-3, Issue-1, June 2013.

[59]PackirisamyMurali and GandhidossSenthilkumar, « Modified Version of Playfair Cipher using Linear Feedback Shift Register », IJCSNS International Journal of Computer Science and Network Security, VOL.8 No.12, December 2008.

[60] OudayNidhal Ameen Hanosh1, BaraaWasfi Salim, « 11 × 11 Playfair Cipher based on a Cascade of LFSRs », IOSR Journal of Computer Engineering (IOSR-JCE), e-ISSN: 2278-0661, www.iosrjournals.org, PP 29-35, Volume 12, Issue 1 (May. - Jun. 2013).

[61] DalalAbdulmohsinHammood, « Breaking A Playfair Cipher Using Memetic Algorithm », Journal of Engineering and Development, , ISSN 1813- 7822, Vol. 17, No.5, November 2013.

[62] V.U.K. Sastry, N.RaviShankar and S. DurgaBhavani, « A Generalized Playfair Cipher involving Intertwining, Interweaving and Iteration », International Journal of Network and Mobile Technologies, VOL 1 / ISSUE 2 / NOVEMBER 2010.

[63] JitendraChoudhary, Ravindra Kumar Gupta, Shailendra Singh, « A Survey of Existing Playfair Ciphers » , International Journal of Engineering and Advanced Technology (IJEAT), ISSN: 2249 – 8958, Volume-2, Issue-4, April 2013.

[64] Safwat Hamad, « A Novel Implementation of an Extended 8x8 Playfair Cipher Using Interweaving on DNA-encoded Data », International Journal of Electrical and Computer Engineering (IJECE), http://iaesjournal.com/online/index.php/IJECE Vol. 4, No. 1, pp. 93~100, Feburary 2014.

[65] Goutam Paul, Subhamoy Maitra,RC4 State Information at Any Stage Reveals the Secret Key,

[66] R. Merkle and M. Hellman, "Hiding information and signatures in trapdoor knapsacks," IEEE Trans. Inform. Theory, vol. IT-24, pp. 525- 530, Sept. 1978.

[67] Guillaume Poupard, « L'étrange sac à dos de Merkle et Hellman », http://www.enseignement.polytechnique.fr/profs/informatique/Guillaume.Poupard/PI

[68] Ashish Agarwal, "Encrypting Messages using the Merkle-Hellman Knapsack Cryptosystem", IJCSNS International Journal of Computer Science and Network Security, pp.12-14, VOL.11 No.5, May 2011

[69] G. Lokeshwari, G. Aparna, Dr. S. Udaya Kumar, "A Novel Scheme for Image Encryption using Merkle-Hellman Knapsack Cryptosystem-Approach, Evaluation and Experimentation", pp.336-339, International Journal of Computer Science & Technology: IJCST, Vol. 2, Issue 4, Oct- Dec. 2011.

[70] Richard M. Karp, "Reducibility Among Combinatorial Problems. In Complexity of Computer Computations", Proc. Sympos. IBM Thomas J. Watson Res. Center, Yorktown Heights, N.Y. New York: Plenum, p.85-103. 1972.

[71] Andrew Clark, ED Dawson and Helen Bergen, Combinatorial Optimization And The Knapsack Cipher, Cryptologia, vol. 20,no. 1,pp.85-93,Jan 1996.

[72] G. Allaire, S.M. Kaber, "Algèbre linéaire numérique". Ellipses, 2002

[73] M. Schatzmann, "Numerical Analysis, A Mathematical Introduction", Oxford University Press, 2002.

[74] L. BOURNON, « Étude comparative de plusieurs familles de sacs à dos », mémoire de D .E .A . Université d'Aix Marseille II, septembre 1991.

[75] Martine Petit, « Étude mathématique de certains systèmes de chiffrement, les sacs à dos », Thèse présentée à l'Université de Rennes, septembre 1982.

[76] E. Karnin, and M. Hellman, "The Largest Super-Increasing Subset of a Random Set", IEEE Trans. Inform. Theory, vol. IT-29, N°1, pp. 146- 148, Jan 1983

[77] B.Chor and R.LRivest, "A knapsack type public-key cryptosystem based on arithmetic in finite fields", IEEE Trans. Inform Theory N°34/5, pp. 901-909, 1988.

[78] Beckett Brian : " Introduction aux méthodes de la cryptologie", Editions Masson, 1990.

[79] Marsault Xavier : " Compression et cryptage des données multimédias", 2e édition revue et augmentée, Editions Hermès, 1992.

[80] B. Chandra, and P. Paul Varghese, “Applications of Cascade Correlation Neural Networks for Cipher System Identification”, World Academy of Science, Engineering and Technology 26.2007.

[82] C. Li, S. Li, D. Zhang, and G. Chen, "Cryptanalysis of a Chaotic Neural Network Based Multimedia Encryption Scheme", in Proc. PCM (3), 2004, pp.418-425.

[83] J. Hérault et Ch. Jutten, "Réseaux neuronaux et traitement du signal", édition Hermès, 1990.

[84] J. Hertz, A. Krogh & R. G. Palmer, "An introduction to the theory of Neural Computation", aux editions Addison-Wesley, 1991.