REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

Université de Mohamed El-Bachir El-Ibrahimi - Bordj Bou Arreridj

Faculté *des Sciences et de la technologie*

Département d'électronique

Mémoire

Présenté pour obtenir

LE DIPLOME DE MASTER

FILIERE : **ELECTRONIQUE**

Spécialité : Electronique des systèmes embarqués

Par

- **SANAA ASMA**
- **ABDELLI FERIEL**

Intitulé

Etude d'un système embarqué de contrôle d'accès biométrique sécurisé par cryptage chaotique.

Soutenu le : ……

Devant le Jury composé de :

Nom & Prénom	***Grade***	***Qualité***	***Etablissement***
Pr. Mohamed El-Hocine DAACHI	Professeur	Président	Univ-BBA
Dr. Seif Eddine AZOUG	MCB	Encadrant	Univ-BBA
Pr. Mustapha SARRA	Professeur	Examinateur	Univ-BBA

Année Universitaire 2024/2025

Abstract :

This project presents the design of an embedded biometric access control system based on fingerprint recognition and secured through chaotic encryption. The main objectives are to develop and implement a reliable authentication system using fingerprints, to ensure the security of biometric data through encryption based on the Lotka-Volterra chaotic system, and to integrate the entire solution into an embedded platform. By combining biometrics, chaotic encryption, and USB key authentication, the project aims to enhance the confidentiality, integrity, and robustness of access control systems, while ensuring fast execution and ease of use on a Raspberry Pi platform.

Résumé :

Ce mémoire présente la conception d'un système embarqué de contrôle d'accès biométrique basé sur les empreintes digitales et sécurisé par un cryptage chaotique.Les principaux objectifs sont de développer et de mettre en œuvre un système fiable d'authentification basé sur l'empreinte digitale, de garantir la sécurité des données biométriques par un chiffrement à base du système chaotique Lotka-Volterra, et d'intégrer le tout sur une plateforme embarquée. En combinant la biométrie, le chiffrement chaotique et l'authentification par clé USB, le projet vise à renforcer la confidentialité, l'intégrité et la robustesse des systèmes d'accès, tout en assurant une exécution rapide et une facilité d'utilisation sur Raspberry Pi.

ملخص :

يقدم هذا البحث تصميم نظام مدمج للتحكم في الدخول البيومتري يعتمد على بصمات الأصابع، ويؤمن البيانات من خلال تشفير فوضوي. تتمثل الأهداف الرئيسية في تطوير وتنفيذ نظام موثوق للمصادقة يعتمد على بصمة الإصبع، وضمان أمان البيانات البيومترية باستخدام خوارزمية تشفير قائمة على النظام الفوضوي Lotka-Volterra، ودمج هذا النظام على منصة مدمجة. من خلال الجمع بين القياسات البيومترية والتشفير الفوضوي والمصادقة عبر مفتاح USB، يهدف المشروع إلى تعزيز خصوصية وسلامة ومتانة أنظمة الدخول، مع ضمان سرعة التنفيذ وسهولة الاستخدام على منصة Raspberry Pi.

Remerciements

En premier, nous remercions Dieu le Tout-Puissant de nous avoir donné le courage, la patience, la santé et la volonté d'entamer et de terminer ce mémoire.

Nous tenons à remercier notre encadrant, Dr.Seif Eddine AZOUG, pour le soutien et l'encadrement qu'il nous a donnés, ainsi que pour le temps consacré à la réalisation de ce mémoire, la confiance qu'il nous a accordée, sa disponibilité, ses remarques, ses conseils et sa patience tout au long de ce chemin.

Nous remercions l'ensemble des enseignants qui nous ont suivis durant notre cycle d'études.

Nos respects aux membres du jury, Pr.Mohamed El-Hocine DAACHI, qui nous a assuré la présidence du jury, et Pr.Mustapha SARRA, qui nous feront l'honneur d'apprécier, d'examiner et de juger ce travail.

Nous ne trouvons pas les mots pour exprimer notre profonde gratitude envers nos parents pour leur soutien inestimable et leur aide tout au long de notre parcours. Leur appui précieux durant toutes ces longues années d'études a été fondamental.

Nous tenons à exprimer toute notre reconnaissance à tous ceux qui ont contribué, de près ou de loin, à la réalisation de ce travail.

Nous remercions ALLAH en premier et en dernier.

Dédicace

Au nom d'Allah le Miséricordieux
Louange à Dieu, par la grâce duquel les bénédictions et les bonnes actions sont accomplies.

Je dédie ce modeste travail à :
Le sourire de ma vie, Leur satisfaction est le secret de ma réussite,
Mes parents :
La source de tendresse, mon adorable mère,
La lumière de mon chemin, mon cher père.

À la joie de la maison : mes sœurs Mouna, Chaima, Hafsa, et mon frère Mohammed.
À mon binôme de mémoire, pour sa collaboration précieuse, son engagement et son soutien tout au long de cette aventure.

Je tiens à exprimer ma profonde gratitude et mes sincères remerciements à tous les enseignants qui ont contribué à notre éducation.

SANAA Asma

Dédicace

Avec tout mon respect et l'expression sincère de ma reconnaissance, je dédie cette réussite, ce diplôme et la joie qui l'accompagne :

À mon paradis sur terre, à la prunelle de mes yeux, à la source de mon bonheur et de ma paix intérieure,

à ma lune qui éclaire mes nuits et à mon étoile d'espoir qui illumine mon chemin :
Ma mère, ma moitié, mon tout.

À celui qui m'a donné la vie, ma source d'amour, de force et de tendresse, mon soutien indéfectible dans chaque étape de mon parcours :
Mon père, mon héros, mon roi.

À mon frère, sa femme, et à mes chères sœurs : merci pour votre soutien constant, votre humour contagieux et votre présence rassurante. Vous êtes ma joie et ma fierté, et je suis honorée de vous avoir à mes côtés.
À mes neveux et nièces, qui remplissent ma vie d'amour, d'innocence et de bonheur.

À mes très chères amies, qui ont été mes piliers dans les épreuves et mes compagnes de bonheur dans les moments de fête.
Merci pour votre amitié sincère, votre soutien inconditionnel et votre amour fidèle.

Sans oublier mon binôme, pour sa patience, sa gentillesse et son soutien moral tout au long de ce projet.

Enfin, à tous ceux qui ont participé à ma réussite, et à tous ceux qui m'aiment et que j'aime :
merci du fond du cœur.

ABDELLI Feriel

Table des matières

Liste des tableaux

Liste des figures

Abréviations

ADN	Acide désoxyribonucléique (DNA : Deoxyribonucleic Acid)
CCD	Charge-Coupled Device (Un appareil photo à transfert de charge)
RTI	Réflexion Interne Totale (Total Internal Reflection)
LED	Light Emitting Diode (Diode électroluminescente)
FTIR	Réflexion Interne Totale Frustrée (Frustrated Total Internal Reflection)
AES	Advanced Encryption Standard (Norme de chiffrement avancée)
RSA	Rivest–Shamir–Adleman (Algorithme de chiffrement à clé publique)
LE	Exposant de Lyapunov (Lyapunov Exponent)
SHA-1	Secure Hash Algorithm 1 (Algorithme de hachage sécurisé 1)
USB	Universal Serial Bus (Bus Universel en Série)
HK	Hash-Key (Clé de hachage)
PK	Personal Key (Clé personnelle)
PHK	Personal Hashed Key (Clé personnelle hachée)
XOR	Exclusive OR (OU exclusif)
ET	Encrypted Template (Template chiffré)
PT	Plain Template (Template en clair)
CV	Confusion Vector (Vecteur de confusion)
DV	Diffusion Vector (Vecteur de diffusio)
DT	Decrypted Template (Template déchiffré)
UART	Universal Asynchronous Receiver Transmitter
RAM	Random Access Memory (Mémoire vive)
FAR	False Acceptance Rate (Taux de fausse acceptation)
FRR	False Rejection Rate (Taux de faux rejet)
PID	Packet Identifier (Identifiant de paquet)
CPU	Central Processing Unit (Processeur central)
IOT	Internet of Things (Internet des objets)
GPIO	General Purpose Input/Output (Entrée/Sortie à usage général)
IDE	Environnement de développement intégré
LCD	Liquid Crystal Display (Afficheur à cristaux liquides)
NPCR	Number of Pixels Change Rate (Taux de changement de pixels)
UACI	Unified Average Changing Intensity
FPGA	Field Programmable Gate Array

Introduction générale

Avec l'émergence des objets connectés, la sécurité devient essentielle. Face aux limites des méthodes classiques, l'authentification biométrique s'impose en s'appuyant sur des caractéristiques uniques comme l'empreinte digitale, l'iris, la voix ou le visage.[1]

Parmi ces techniques, l'identification par empreinte digitale est l'une des plus largement adoptées, notamment dans les systèmes embarqués et les applications à faible coût, grâce à sa facilité d'intégration, sa précision et sa faible dépendance à des équipements complexes.[2]

Elle assure une reconnaissance rapide et fiable, tout en limitant les risques de falsification. Toutefois, la sensibilité des données biométriques pose un enjeu majeur de confidentialité, car une empreinte, à la différence d'un mot de passe, modifiée en cas de compromission.[2]

Pour sécuriser ces données, des méthodes de chifferement sont utilisées : le chiffrement symétrique (AES) [3], ou asymétrique (RSA)[4] permet de masquer les templates biométriques. Plus récemment, les systèmes chaotiques, issus de la théorie du chaos, ont été explorés pour leur capacité à générer des séquences pseudo-aléatoires sensibles aux conditions initiales, renforçant ainsi la résistance au déchiffrement.

Ce travail propose un système sécurisé de contrôle d'accès biométrique par empreinte digitale. Les templates sont chiffrés via une méthode chaotique basée sur le modèle Lotka-Volterra, combinée au hachage SHA-1 pour garantir confidentialité et intégrité.

La mise en œuvre repose sur des composants abordables comme le capteur DY50 et le Raspberry Pi 3 B+, démontrant la faisabilité du système en conditions réelles.

Dans le premier chapitre, nous allons voir le principe de l'authentification biométrique et comment s'effectue l'authentification par reconnaissance d'empreinte.

Dans le deuxième chapitre, nous allons voir les différentes méthodes existantes pour la sécurisation des données biométriques en se concentrant sur l'utilisation du chiffrement chaotique et son intérêt en cryptographie.

Le troisième chapitre, présente une étude détaillée d'un système de contrôle d'accès par reconnaissance d'empreinte sécurisée par un chiffrement chaotique dans le but d'assurer la confidentialité des empreintes et leurs intégrités par l'utilisation d'une fonction de hachage de 160bits.

Le dernier chapitre présente la réalisation pratique du système étudié ainsi que les algorithmes implémentés. Les résultats d'une étude comparative avec un autre système chaotique est également présentée.

Chapitre 1

Généralités sur l'authentification biométrique

1.1 Introduction

La biométrie constitue l'une des technologies les plus utilisées dans les applications courantes d'authentification et d'identification.

Ce chapitre traite la biométrie dans les systèmes embarqués, en expliquant l'authentification biométrique et ses différentes catégories en se focalisant sur les spécificités des empreintes digitales pour l'authentification biométrique.

1.2 L'authentification biométrique

La biométrie est dérivée des termes grecs, « bios » signifiant la vie et « métrique », qui signifie mesure. [1]

L'authentification biométrique est l'identification automatique d'un individu à travers ses caractéristiques physiques biométriques qui peuvent être divisés en trois catégories :

— L'analyse biologique.

— L'analyse comportementale.

— L'analyse morphologique.

La figure 1.1 montre trois différentes caractéristiques physiques biométriques illustrative pour ces catégories.

L'iris pour l'analyse morphologique, la signature personnelle pour l'analyse comportementale et l'ADN pour l'analyse biologique.

(1) Iris (2) Signature (3) ADN

FIGURE 1.1 – Différentes caractéristiques biométriques

L'identification par l'analyse biologique de l'ADN utilisée principalement dans le cadre d'enquêtes criminelles est une technique très fiable. Elle s'appuie sur l'unicité du profil génétique de chaque personne.[5]

L'identification comportementale par le style d'écriture unique à chaque personne est souvent employée dans les domaines juridiques/administratifs en se basant sur sa signature et

en utilisant une tablette graphique. Elle donne la possibilité de concevoir un modèle distinctif pouvant être utilisé pour son identification.[5]

L'identification morphologique par l'analyse du motif distinctif de l'iris se caractérise par sa fiabilité exceptionnelle, avec une probabilité de confusion estimée à 1 sur10^{72}.[5]

Un capteur photo et une lumière infrarouge sont utilisés pour capturer l'image de l'iris, dans le but de réduire les impacts de la lumière visible, de la fatigue ou des gestes spontanés. [5]

Toutes ces caractéristiques offrent des performances élevées en matière d'authentification biométrique mais restent couteuses et complexes telles que la nécessité d'une tablette graphique à haute précision pour la capture de la signature, d'une caméra à haute définition pour la capture de l'image de l'IRIS, ou d'un automate complexe pour l'analyse ADN.

De ce fait, le recours à des techniques d'authentification biométrique moins couteuses est à privilégier pour la gestion de la sécurité des systèmes embarqués comme par exemple l'analyse morphologique de l'empreinte qui nécessite des capteurs moins chers.

1.3 Authentification par empreinte digitale

La reconnaissance d'empreintes est aujourd'hui très utilisée dans les systèmes embarqués pour la gestion et le contrôle d'accès [6]. Elle offre plusieurs avantages :

— Capteurs d'empreintes compacts et abordables.

— Facilité d'utilisation nécessitant une intervention minimale de l'utilisateur.

— Un degré élevé de précision lors de l'identification.

La reconnaissance d'empreintes digitales passe par plusieurs étapes comme le montre la figure 1.2 :

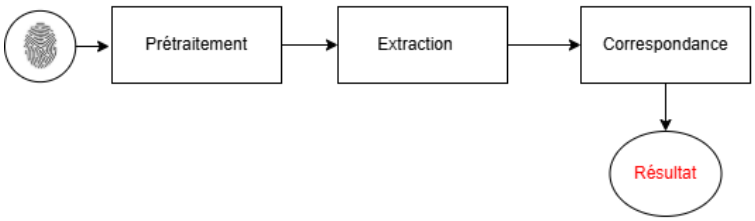


FIGURE 1.2 – Différentes phases d'authentification par empreinte digitale. [6]

— **Capture de l'empreinte** : Un capteur enregistre l'empreinte d'un individu pour générer une représentation numérique brute.[6]

— **Prétraitement** : L'image de l'empreinte est filtrée et ajustée pour faciliter l'extraction de ses principales caractéristiques.[6]

— **Extraction des caractéristiques** : Une analyse approfondie de l'empreinte afin d'isoler ses caractéristiques distinctives.[6]

— **Correspondance** : où le vecteur de caractéristiques de l'empreinte digitale entrée est confronté à un ou plusieurs modèles préexistants.[6]

— **Comparaison** : Un appariement est effectué entre ces caractéristiques et plusieurs modèles précédemment stockés dans la base de données afin d'authentifier la personne.[6]

1.3.1 Caractéristiques distinctives des empreintes

Une empreinte digitale est constituée d'une série de lignes organisées de façon essentiellement parallèle, créant un motif distinctif qui est spécifique à chaque personne comme le montre la figure 1.3.[2]

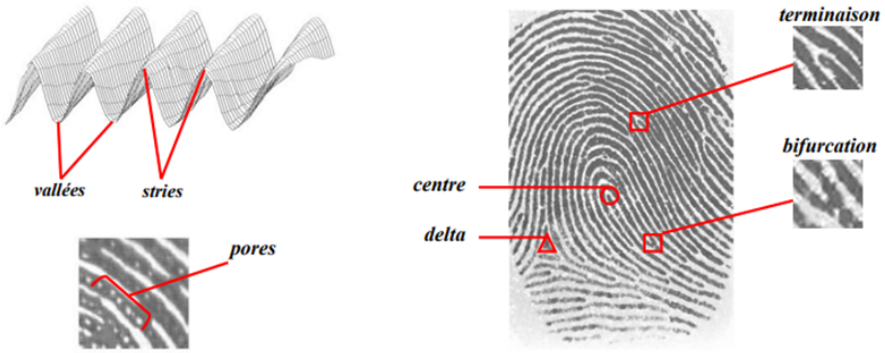


FIGURE 1.3 – Caractéristiques d'une empreinte digitale. [2]

Ce motif comporte deux éléments majeurs : les crêtes (également appelées stries), qui font référence aux zones en relief qui touchent une surface lorsqu'on les effleure, et les sillons, qui sont les dépressions entre deux crêtes.[2]

De plus, les crêtes présentent un agencement régulier de pores, ajoutant une autre dimension d'unicité à l'empreinte.[2]

Ces caractéristiques à l'échelle microscopique sont cruciales dans les procédures de reconnaissance biométrique.[2]

Chaque empreinte comporte un ensemble de points singuliers, à la fois globaux (les centres et les deltas) et locaux (les minuties).[2]

Les centres sont associés aux endroits où convergent les lignes, tandis que les deltas se réfèrent aux zones où celles-ci divergent.[2]

L'emploi d'un nombre précis de centres et de deltas permet la catégorisation des empreintes. En fonction de leur motif général, nous identifions principalement trois grandes

catégories comme le montre la figure 1.4 :

— Les boucles constituent 65% des empreintes identifiées.

— Les traces en forme de spirale (whorl) constituent 30% des empreintes identifiées.

— Les empreintes en forme d'arches (arch) constituent 5% du total.

FIGURE 1.4 – Les trois principales classes d'empreintes, boucle (1), spire (2), arche (3).[2]

Chaque individu possède une empreinte digitale distincte. Malgré l'influence de la génétique sur la structure globale, les détails spécifiques se manifestent de façon sporadique.

Elle demeure constante tout au long de l'existence, ce qui en fait un outil d'identification sûr, même en cas de vrais jumeaux.[2]

1.3.2 Capteurs d'empreintes

Parmi les différents types de capteurs d'empreintes digitales, on peut citer :

— **Capteurs capacitifs** : Ils évaluent la capacité électrique formée entre la peau et le détecteur, ce qui permet de différencier les crêtes et les vallées en fonction du relief de l'empreinte.[2]

— **Capteurs pyroélectriques** : exploitent la chaleur émise par la peau pour créer une représentation de l'empreinte digitale. Un matériau pyroélectrique sensible à la chaleur permet de détecter ses changements.[2]

— **Capteurs optiques** : exploitent l'optique pour l'identification des empreintes digitales, cette technique est la plus répandue .[2]

Le tableau 1.1 illustre de manière comparative les différences entre ces types de capteurs :

Type de capteur	Exemples	Avantages	Inconvénients
Optique	DY50, R305, ZFM-20	— Peu coûteux — Facile à utiliser et intégrer	— Sensible à la lumière
Capacitif	Touch ID (Apple), FS7600	— Précis — Difficile à falsifier — Compact	— Moins efficace si doigt humide ou sale
Pyroélectrique	NEC pyroélectrique, capteurs IR	— Détecte la chaleur (doigt vivant) — Sécurisé	— Lent — Affecté par la température ambiante

TABLE 1.1 – Comparaison entre les types de capteurs d'empreinte.

1.3.3 Principe de fonctionnement d'un capteur optique

Un capteur optique fonctionne selon le principe de la réflexion interne totale (RTI). Dans ce type de capteur, un prisme en verre est employé pour rendre cette réflexion plus aisée. Une lumière LED, généralement de teinte bleue, est projetée à travers l'une des faces du prisme à un angle spécifique pour déclencher le phénomène de la RTI. Le prisme renvoie la lumière à une autre de ses faces où se trouvent une lentille et un capteur d'image.[7]

Quand aucun doigt n'est apposé sur le prisme, la lumière est totalement réfléchie, produisant une image homogène.[7]

Lorsqu'un doigt est présenté, seules les crêtes entrent en contact avec le verre, tandis que les vallées demeurent isolées par des bulles d'air.[7]

Cette variation de contact, causée par des indices de réfraction distincts entre l'épiderme et l'air, trouble l'onde évanescente, un processus désigné sous le nom de réflexion interne totale frustrée (FTIR).[7]

Cela change l'intensité de la lumière réfléchie(Voir la figure 1.5), donnant au capteur la capacité de produire une image numérique contrastée de l'empreinte digitale.[7]

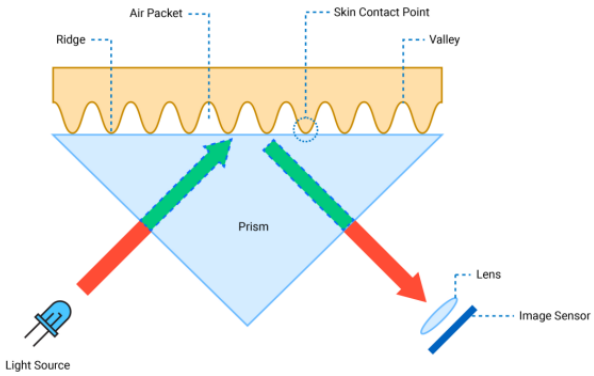


FIGURE 1.5 – Principe de fonctionnement d'un capteur optique. [7]

1.4 Conclusion

Dans ce chapitre nous avons revus les principes de base de l'authentification biométrique, avec une attention particulière portée à la reconnaissance d'empreintes digitales.

Cette technique d'authentification est l'une des plus sûres pour la gestion d'accès, en raison de la singularité et de la constance des empreintes digitales.

Nous avons aussi revus divers capteurs qui facilitent la collecte et l'étude de ces empreintes, ainsi que les phases cruciales du processus de vérification biométrique.

Malgré cela, l'emploi des données biométriques pose d'importants défis liés à la sécurité et à la confidentialité.

C'est pourquoi le prochain chapitre se focalisera sur les méthodes et processus assurant une protection efficace de la confidentialité des données biométriques.

Chapitre 2

Chiffrement chaotique des données biométriques

2.1 Introduction

Avec la démocratisation des technologies biométriques, la confidentialité et la sécurité des données personnelles biométriques devient une priorité absolue.

Ce chapitre s'intéresse aux différentes techniques utilisées pour protéger la confidentialité des données par l'utilisation d'outils cryptographiques tels que les algorithmes de chiffrement pour protéger la confidentialité des données et les fonctions de hachage pour assurer l'intégrité des données ainsi que de nouvelles approches basées sur le chaos et ses propriétés cryptographiques.

2.2 Définitions et objectifs

Le terme « cryptographie » trouve ses origines dans la langue grecque : il provient de krýpto, qui veut dire « caché », et du verbe gráfo, qui se traduit par « écrire ».[8]

La cryptographie n'est pas le seul outil pour garantir la sécurité des données, elle s'inscrit dans un ensemble de méthodes.[9]

L'objectif de la cryptographie c'est masquer le sens d'un message. Elle s'appuie sur des méthodes mathématiques mises en œuvre pour garantir les points suivants :

— **La confidentialité** : visant à restreindre l'accès au contenu des données aux seules entités autorisées. Elle est synonyme de secret et de vie privée. Pour assurer cela, des algorithmes mathématiques de chiffrement des données sont utilisés dans le but de rendre les données inintelligibles à toute personne non autorisée.[9]

— **Intégrité des données** : Le principe d'intégrité des données vise à prévenir toute altération non autorisée de l'information. Pour garantir cette propriété, il est impératif de disposer de mécanismes permettant de détecter toute manipulation illicite, notamment les opérations d'insertion, de suppression ou de substitution de données au long de leur cycle de vie.[9]

— **Authentification** : L'authentification telle que l'authentification biométrique permet à deux parties de vérifier leurs identités.[9]

— **Non-répudiation** : La non-répudiation est un service destiné à empêcher qu'une entité impliquée dans une communication ou une transaction ne puisse ultérieurement nier sa participation ou ses engagements.[9]

2.3 Généralités sur le chiffrement

Le principe du chiffrement de données biométriques consiste à rendre des données en clair en données chiffrées afin d'assurer la confidentialité des données avec une ou plusieurs

clés de chiffrement. En effet, il existe deux principaux types de chiffrement de données : chiffrement symétrique et chiffrement asymétrique.[9]

2.3.1 Chiffrement symétrique

Le chiffrement symétrique, aussi appelé à clé secrète permet un chiffrement basé sur l'usage d'une clé confidentielle, généralement une suite de caractères.

Sans cette clé, le déchiffrement des données chiffrées devient extrêmement difficile, voire irréalisable (en fonction de la sécurité de l'algorithme et la complexité de la clé).

Le chiffrement symétrique est très utilisé pour la protection des données biométriques.[10]

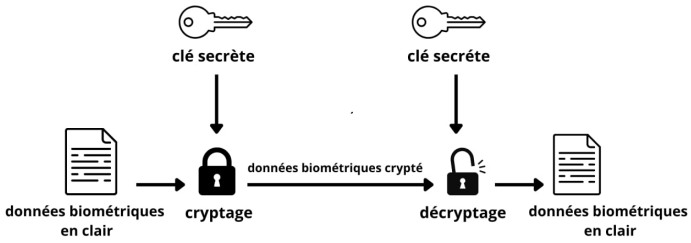


FIGURE 2.1 – Organigramme du chiffrement symétrique

2.3.2 Chiffrement asymétrique

Le chiffrement asymétrique, aussi connu sous le nom de chiffrement à clé publique, adopte un fonctionnement très différent de celui du chiffrement symétrique.

Dans ce système, deux clés sont utilisées : Une publique et l'autre privée. La clé publique est accessible à toutes les parties et sert seulement pour le chiffrement, et la clé privée est utilisée seulement pour le déchiffrement.

Les nombres premiers jouent un rôle fondamental dans la robustesse des algorithmes de cryptographie asymétrique.[10]

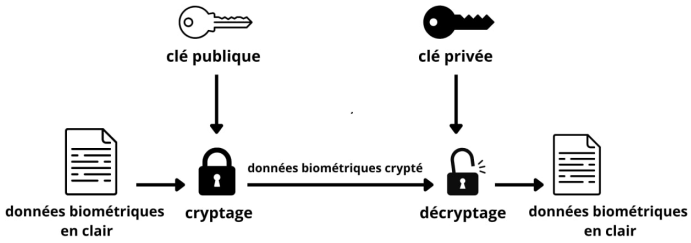


FIGURE 2.2 – Organigramme du chiffrement asymétrique

2.4 Algorithmes conventionnels de chiffrement

2.4.1 Advanced Encryption Standard (AES)

Dans [11, 3], les auteurs utilisent le Advanced Encryption Standard (AES) une méthode de chiffrement symétrique par blocs pour sécuriser les données biométriques.

L'algorithme AES utilise une clé confidentielle de 128/192/256 bits pour le chiffrement et le déchiffrement d'un segment de 16 octets (128 bits). L'AES se base sur des opérations de substitution ou de transposition pour réaliser le chiffrement et le déchiffrement en 10, 12 ou 14 tours (rounds).

Le nombre précis de tours est déterminé par la longueur de la clé secrète. Quatre types d'opérations sont généralement inclus dans chaque tour de l'algorithme AES :

— Substitution des octets (SubBytes).

— Décalage des lignes (ShiftRows).

— Mélange des colonnes (MixColumns).

— Inclusion de l'opération AddRoundKey.

Le dernier tour ne comprend pas l'opération MixColumns, ce qui le différencie légèrement des tours standards.En pratique.

AES présente une résistance face aux attaques par force brute qui consistent à essayer toutes les combinaisons possibles de la clé jusqu'à trouver la bonne et réussir à déchiffrer les données.

L'algorithme AES-256 utilise une clé de 256 bits et 14 cycles de chiffrement ce qui lui offre la capacité de résister aux différentes attaques avec une complexité globale évaluée à 2^{131} en temps et 2^{65} en mémoire.

2.4.2 RSA

RSA est un algorithme de chiffrement asymétrique connu qui peut être utilisé pour la protection des données biométrique [4]. Il repose sur la complexité de la factorisation de grands nombres entiers. Cet algorithme, intègre la génération de nombres complexes, l'exponentiation et les mathématiques modulaires.

Dans le modèle RSA, un utilisateur génère et diffuse une clé publique qui résulte de la multiplication de deux grands nombres premiers, tout en gardant secrét les facteurs premiers.

Avec la clé publique de l'utilisateur, un autre utilisateur a la possibilité de recourir à cette clé pour le chiffrement d'un message. Uniquement le détenteur de la clé publique qui est au courant des facteurs premiers a la capacité de déchiffrer le message. Le processus RSA se divise en trois phases, à savoir la génération de clés, le chiffrement et le déchiffrement.

Dans [4], les auteurs utilisent l'algorithme RSA pour générer des paires de clés RSA en combinant les empreintes digitales et un mot de passe utilisateur.

L'approche utilise les distances entre les points de minutie des empreintes, codées en Gray, pour créer une chaîne binaire stable.

Pour compenser les variations biométriques, un code de correction d'erreurs Reed–Solomon est appliqué. Le mot de passe est également transformé en une chaîne stable via une fonction de dérivation de clé.

Ces deux éléments sont ensuite combinés pour générer une paire de clés RSA sans nécessiter de stockage permanent des clés privées, renforçant ainsi la sécurité contre les attaques telles que l'interception ou la substitution.

2.5 Chiffrement par le chaos

Le chiffrement par le chaos ou le chiffrement chaotique est une nouvelle approche pour la protection de la confidentialité des données biométriques.[12]

La théorie du chaos vise l'analyse du comportement des systèmes dynamiques non linéaires, décrits par des équations déterministes qui peuvent afficher un comportement chaotique si elles sont appliquées de manière itérative, formant ainsi ce qu'on appelle des suites dites chaotiques ou en anglais « chaotic maps ».[12]

Ces suites chaotiques sont pseudo-aléatoires ce qui leurs offre des propriétés utiles en cryptographies qu'on peut résumer sur les trois points suivants :

— **Pseudo-aléatoire**

Les suites chaotiques, qui reposent sur des équations déterministes et non probabilistes, ont la capacité de générer un comportement aléatoire à partir de ces équations. Cela offre la possibilité d'utiliser les suites chaotiques en cryptographie comme générateurs de nombres pseudo-aléatoires.[12]

— **Sensibilité aux conditions initiales**

Les suites chaotiques sont très réactives à leurs conditions initiales, où même une légère modification de l'état initial peut produire une variation radicale de l'état final. Cette sensibilité élevée est utilisée en chiffrement symétrique où leurs paramètres sont perçus comme une clé secrète symétrique.[12]

— **Ergodicité**

Un processus chaotique est ergodique, ce qui signifie que peu importe la distribution de la variable en entrée, il affichera la même distribution en sortie. Une caractéristique importante en chiffrement de données.[12]

2.6 Méthodes d'identification des systèmes chaotiques

Il existe plusieurs méthodes pour déterminer la stabilité d'un systéme dynamique sous forme d'une suite mathématique. L'étude de la stabilité d'une suite permet de déterminer si elle est chaotique ou non (périodique).[13]

Ces méthodes sont en général sous formes graphiques ce qui permet d'observer le comportement du système en cours de développement en fonction des valeurs initiales de ses paramètres de contrôle. Cela permet d'identifier les valeurs pour lesquelles le système est chaotique.[13]

Parmi les méthodes les plus connues on trouve :

— L'exposant de Lyapunov.

— Le diagramme de bifurcation.

— L'attracteur.

Comme il existe plusieurs suites chaotiques dans la littérature [14], nous allons nous focaliser dans cette section sur la suite logistique servant à démontrer le principe d'identification des systèmes chaotiques.

2.6.1 La suite logistique

La suite logistique est une suite chaotique unidimensionnelle défini par l'equation 2.1 :

$$x_{n+1} = r x_n (1 - x_n) \tag{2.1}$$

Elle est caractérisée par la simplicité de son équation de récurrence où $x \in [0, 1]$, n le nombre d'itérations, et $r \in [0, 4]$ est un paramètre de contrôle.[15]

La suite logistique peut être employée en cryptographie pour la génération des clés de chiffrement à deux variables ($x(0)$, r), dont le choix influence le résultat du chiffrement.[15]

2.6.2 Exposant de Lyaponov

L'indicateur principal employé pour évaluer si un système dynamique présente un comportement chaotique est l'exposant de Lyapunov (LE).[15]

Il évalue la sensibilité aux conditions de départ, c'est-à-dire le rythme auquel deux trajets similaires se séparent au fur et à mesure que le temps passe dans un espace de phases limité (l'attracteur étrange).[15]

Plus cette divergence se produit rapidement, plus le système présente un caractère chaotique.

Le nombre d'exposants de Lyapunov est équivalent à la dimension de l'espace des phases.[15]

Dans un système unidimensionnel discret, si le système est chaotique, deux conditions initiales x_0 et $x_0 + \varepsilon$ similaires auront tendance à diverger de manière exponentielle.[15]

d'où l'exposant de Lyaponov défini comme suit :

$$\lambda = \lim_{n\to\infty} \frac{1}{n} \left(\sum_{i=0}^{n-1} \ln \left| \frac{df(x_i)}{dx_i} \right| \right) \tag{2.2}$$

— Si $\lambda > 0$, alors il existe une sensibilité aux conditions initiales.

— Si $\lambda < 0$, les trajectoires convergent et l'information sur les conditions initiales est perdue.

La figure 2.3 suivante présente le diagramme de l'exposant de Lyapunov de la suite logistique pour $x0 = 0.1$ et $r \in [3, 4]$. Nous pouvons observer le comportement chaotique de la suite logistique dans l'intervalle [3.55,4] où λ est positif.

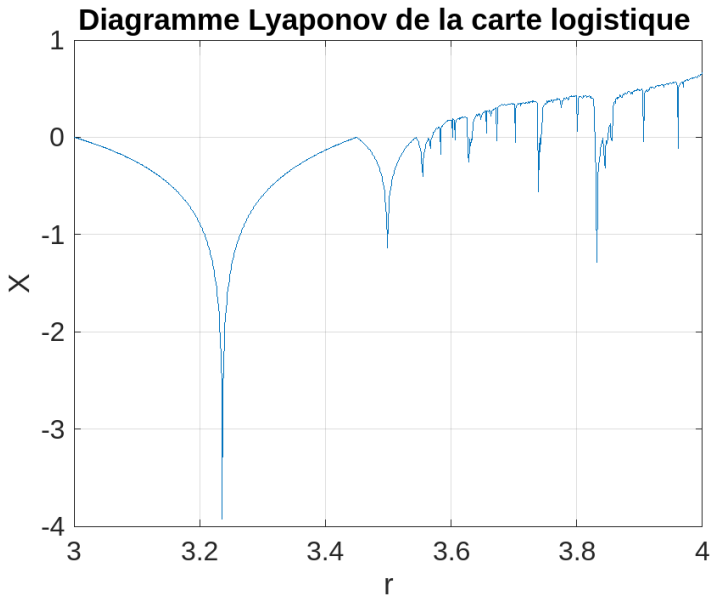


FIGURE 2.3 – Diagramme de Lyaponov de la suite logistique

2.6.3 Bifurcation du système

La théorie des bifurcations examine les modifications qualitatives du comportement d'un système dynamique en réponse à une variation de paramètre ce qui offre la possibilité de tracer les divers régimes dynamiques possibles.[15]

La figure 2.4 montre le diagramme de bifurcation de la suite logistique lorsque $x0 = 0.1$ et $r \in [0, 4]$. On peut voir le phénomène de chaos et de bifurcation où la suite logistique devient chaotique seulement lorsque le paramètre de contrôle $r \in [3.55, 4]$.

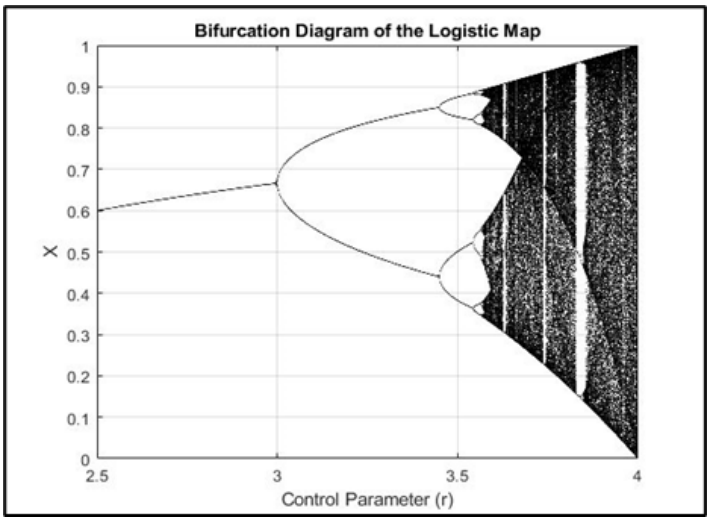


FIGURE 2.4 – Diagramme de bifurcation de la suite logistique

2.6.4 Attracteur

Dans les systèmes dynamiques, l'ensemble des états vers lesquels les trajectoires finissent par converger est appelé « attracteur ».

Ces attracteurs reflètent la dynamique asymptotique du système et peuvent prendre diverses formes géométriques.[15]

On distingue généralement quatre catégories :

— Le point fixe : représentant un état stationnaire .

— Le cycle limite : associé à une solution périodique .

— Le tore : traduisant des oscillations quasi périodiques dues à plusieurs fréquences indépendantes .

— Les attracteurs étranges : qui possèdent une structure fractale complexe révélant un comportement chaotique.

La figure 2.5 présente le diagramme d'attracteur de la suite logistique, qui est calculé par l'équation suivante :

$$x_{n+1} = r \cdot x_n \cdot (1 - x_n) \tag{2.3}$$

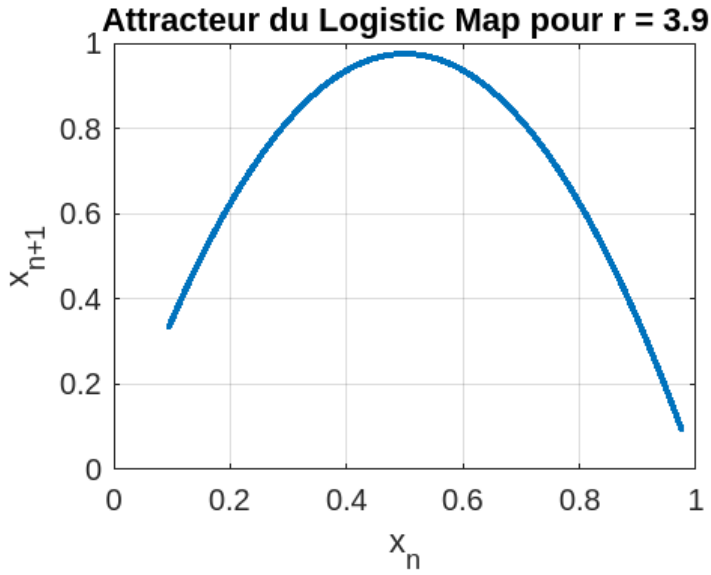


FIGURE 2.5 – Diagramme d'attracteur de la suite logistique [16]

2.7 Fonctions de hachage

Une fonction de hachage est un mécanisme qui convertit des données biométriques de taille variable en une valeur de taille fixe.[17]

On retrouve fréquemment ces fonctions, particulièrement dans les bases de données, pour optimiser la recherche grâce aux tables de hachage.

Une table de hachage lie une valeur à une clé en employant une fonction de hachage pour produire une adresse dans un tableau, ce qui rend l'accès aux informations presque immédiat.[17]

Toutefois, les fonctions de hachage cryptographiques se distinguent des fonctions utilisées pour l'indexation : elles doivent accorder certaines caractéristiques spécifiques en lien avec la sécurité de l'information, telles que la résistance aux collisions et l'impossibilité d'extraire la donnée originale à partir de son empreinte.[17]

2.7.1 SHA-1

SHA-1 est une fonction de hachage 160 bits utilisant des blocs de message de 512 bits et possédant un état de 160 bits.[18]. La méthode de compression SHA-1 s'appuie sur la structure Davies-Meyer où l'opération XOR est remplacée par cinq additions modulo 2^{32}. Ces additions sont effectuées entre les cinq registres de 32 bits issus de la variable de chaînage et les cinq registres résultant de la permutation.[18]

Le bloc de message, initialement divisé en 16 mots de 32 bits $(M_0, \ldots, M_{15})$, subit ensuite une expansion linéaire sur le corps permettant d'obtenir 80 mots de 32 bits.[18]

On peut exprimer la fonction d'étape de SHA-1 à l'aide d'une seule variable :

$$A_{i+1} = (A_i \ll 5) + f_i(A_{i-1}, A_{i-2} \gg 2, A_{i-3} \gg 2) + (A_{i-4} \gg 2) + K_i + M_i \qquad (2.4)$$

où K_i sont des constantes prédéterminées , f_i sont des fonctions booléennes.

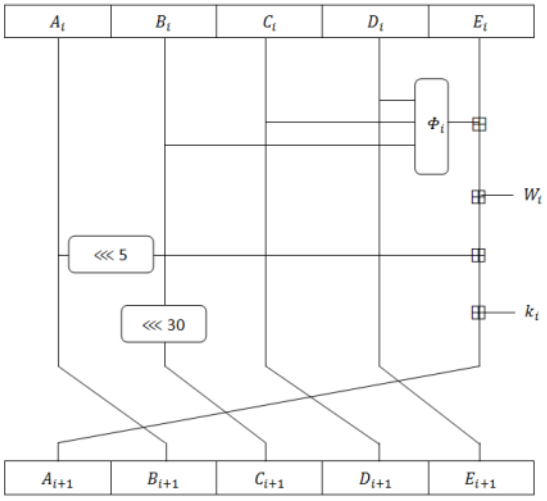


FIGURE 2.6 – Une étape de la fonction de compression de SHA-1[19]

2.8 Conclusion

En conclusion , nous avons revu les différentes techniques existantes de sécurisation des données biométriques et particuliérement l'utilisation de systèmes chaotiques et les fonctions de hachage. Ces concepts représentent les bases théoriques essentielles pour créer un système d'authentification fiable et sécurisé.

Chapitre 3

Etude d'un système d'accès biométrique basé sur le chaos

3.1 Introduction

Ce chapitre présente un système de contrôle d'accès par reconnaissance des empreintes digitales où les empreintes digitales sont protégées par un chiffrement chaotique.

3.2 Objectif

Les systèmes de contrôle d'accès basés sur les empreintes digitales présentent une vulnérabilité importante : les données extraites sont généralement non chiffrées mais codées sous forme d'un modèle appelé template. Cela rend ces templates exploitables en cas de vol ou d'accès non autorisé (physique ou à distance) au systéme d'authentification exposant ainsi en clair les templates des empreintes des utilisateurs ce qui met en danger la vie privée de l'utilisateur car une empreinte digitale ne peut être changée ou mise à jour comme c'est le cas des mots de passe.

Pour répondre à ce problème, nous étudions le système de contrôle d'accés présenté sur [14] dans lequel les templates d'empreintes sont chiffrés à l'aide d'un chiffrement chaotique, rendant leur exploitation pratiquement impossible sans la clé de chiffrement.

3.3 Système d'accès par empreintes basé sur le chaos

Dans cette section, nous proposons une version modifiée du système de contrôle d'accès par empreinte basé sur le chiffrement chaotique proposé sur [14].

3.3.1 Principe de base

En général, un capteur d'empreinte transforme l'image de l'empreinte scannée en un modèle numérique appelé template ou PT (Plain Template) qui est sous forme claire non chiffrée.

Une clé hash HK (Hash Key) de 160 bits doit être générée avec une fonction de hachage SHA-1 sur le template PT avant de le stocker sous forme chiffrée.

En effet, pour assurer la confidentialité du template PT d'un utilisateur autorisé, il doit être stocké sous un format chiffré spécifique que l'on nomme ET/HK (Encrypted Template/Hash Key) comme le montre la figure 3.1.

Le template ET est obtenu à l'aide d'un chiffrement chaotique en utilisant une clé secréte PHK (Personal Hash Key). Ainsi, l'annexion de HK avec ET forme le nouveau template ET/HK et la clé PHK est obtenue en effectuant une opération logique XOR (OU Exclusif) bit par bit entre la clé HK de l'utilisateur et une autre clé personnelle PK (Personal Key) de 160 bits secréte propre au système.

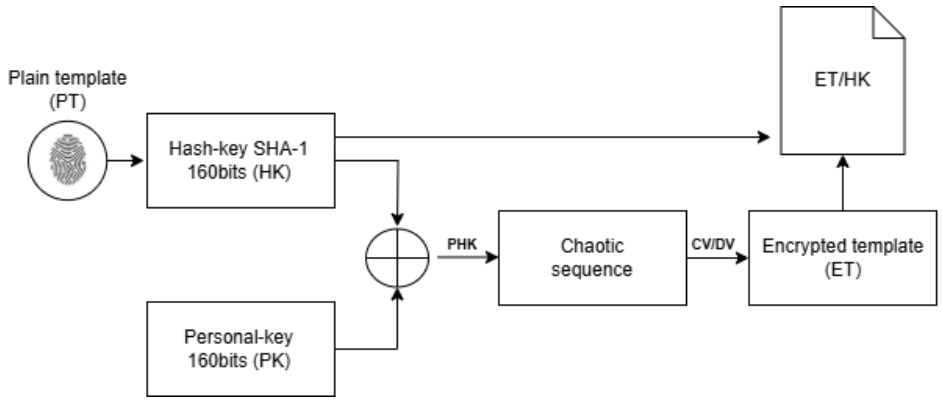


FIGURE 3.1 – Principe de génération du template ET/HK

Ce chiffrement chaotique est basé sur le principe de la confusion et de diffusion CV/DV (Confusion Vector / Diffusion Vector) :

Le vecteur de confusion CV implique l'emploi de suites chaotiques comme générateur de vecteurs de permutation afin de rendre complexe la relation entre le template chiffré ET et le template original PT.[20]

Le vecteur de diffusion DV a pour objectif de diffuser l'erreur de changement d'un seul élément du template PT en clair. Cela aura un impact plus grand sur la valeur des éléments du template chiffré ET.[20]

Ainsi, il est presque impossible de retrouver le template original sans la clé secréte PHK.

Contrairement au systéme proposé dans [14], l'utilisation du template ET/HK a pour avantage de renforcer la sécurité des empreintes où le template chiffré ET de l'utilisateur n'est déchiffré que si l'utilisateur présente un HK valide lors de l'authentification. De plus, cela offre une recherche plus rapide du template de l'utilisateur en se basant seulement sur sa clé HK.

3.3.2 Organigramme générale du systéme

Comme le montre la figure 3.2, chaque utilisateur autorisé a un ET/HK qui lui est propre dans la mémoire systéme qui servira à le reconnaître lors de l'authentification. De ce fait, une copie de la clé HK doit être remise à l'utilisateur sous forme d'un TOKEN ou jeton sur un support de stockage (clé USB ou un tag RFID).

Avant que l'utilisateur ne soit inviter à placer son doigt sur le capteur d'empreinte, il doit d'abord insérer le TOKEN précédent. Une fois le TOKEN détécté par le systéme, la clé HK sera extraite du TOKEN et vérifier si une copie identique est déjà répertorié dans la mémoire du systéme. Si c'est le cas, la clé PHK est générée comme décris dans la section précédente dans le but de déchiffrer le template ET correspondant au TOKEN.

Comme pour le chiffrement, le déchiffrement repose sur une clé PHK et un déchiffrement

chaotique produisant des vecteurs de permutation (confusion vector CV) et de diffusion (diffusion vector DV).

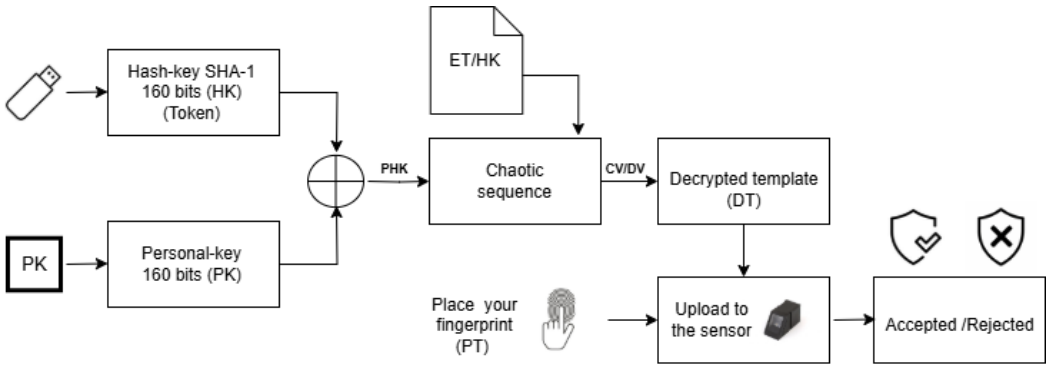


FIGURE 3.2 – Organigramme générale du système de contrôle d'accès par empreinte

Si pour une raison ou une autre la clé HK (TOKEN) est incorrecte, l'utilisateur se verra refuser l'accés et l'opération de déchiffrement ne sera pas entamée.

Dans le cas contraire, le template déchiffré nommé DT (Decrypted Template) doit être transféré vers la mémoire du capteur d'empreintes digitales. Cette opération est nécessaire pour l'authentification de l'utilisateur car la mémoire du capteur a été préalablement vidée pour garantir qu'aucune empreinte n'est présente dans le capteur.

L'utilisateur est alors inviter à placer son doigt sur le capteur pour effectuer une comparaison entre le template PT de l'utilisateur et le template déchiffrée DT présent dans la mémoire du capteur. En fonction du résultat de correspondance, le système décidera d'accorder ou non l'accés à l'utilisateur.

Pour renforcer la sécurité, toutes les données enregistrées dans la mémoire du capteur sont automatiquement effacées à la fin du processus d'authentification et ne garder aucune trace des empreintes en clair.

Au final, ce système permet d'assurer la protection de la confidentialité des données biométriques par un chiffrement chaotique. Il permet également de réduire le risque de l'imposture d'identité par l'utilisation d'une fonction de hachage SHA-1 pour générer un token propre à chaque utilisateur ce qui assure l'intégrité des données.

3.3.3 Choix du capteur d'empreintes

Notre choix s'est porté sur le capteur optique DY50, en raison de sa compatibilité avec plusieurs plateformes et sa capacité à stocker et comparer des empreintes en interne. Il offre un bon équilibre entre performance, coût et flexibilité.

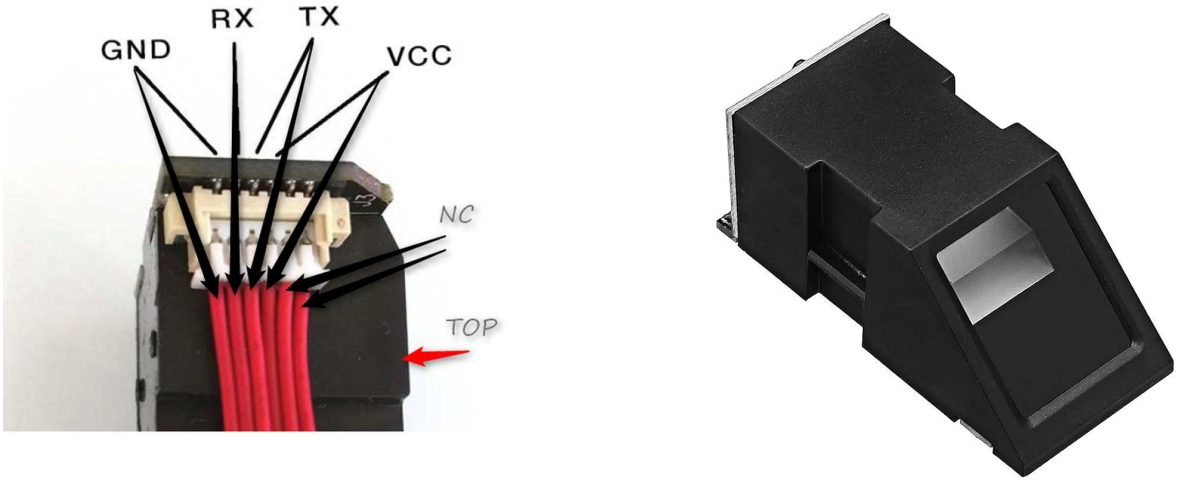
Le DY50 est capable de numériser les empreintes et d'envoyer les informations traitées à un microcontrôleur par liaison série UART.[7]

Ce module conserve toutes les empreintes enregistrées et a la capacité d'enregistrer jusqu'à 162 empreintes digitales distinctes.[7]

Ce capteur est équipé d'une mémoire tampon pour l'image située dans la RAM du module ainsi que de deux autres mémoires tampon de 512 octets chacune pour les caractéristiques d'empreintes .[8]

À l'aide d'instructions, les utilisateurs ont la possibilité de lire et d'écrire dans n'importe quel tampon parmi ceux-ci.[8]

La figure 3.3 présente la disposition des broches du module, ainsi que le capteur biométrique utilisé pour la collecte des empreintes digitales.



(a) Brochage du module (b) Capteur d'empreintes digitales

FIGURE 3.3 – Illustration du brochage du module et du capteur d'empreintes digitales

L'échange de données et des commandes entre le lecteur et le coeur du systéme s'effectue en half-duplex sous forme de paquets structurés [21] comme le montre le tableau 3.1.

Header	**Adresse**	**Identifiant du paquet**	**Longueur**	**Contenu du paquet**	**Checksum**
2 octets	4 octets	1 octet	2 octets	Variable	2 octets

TABLE 3.1 – Format d'un paquet de communication avec le lecteur d'empreintes.[21]

— **Header** : Valeur constante 0xEF01.

— **Adresse** : Valeur par défaut 0xFFFFFFFF.

— **Identifiant Paquet (PID)** :

  — 0x01 : Paquet de commande.

  — 0x02 : Paquet de données.

  - 0x07 : Accusé de réception.
  - 0x08 : Signale la fin des paquets de données.
- **Longueur du paquet** : Se rapporte à la longueur du contenu du paquet, à laquelle il faut ajouter la taille du checksum. L'unité de mesure est l'octet. La taille maximale autorisée est de 256 octets. Le premier octet de poids fort est transmis en premier.
- **Contenu de paquet** : Cela peut inclure des commandes, des informations, des réglages de commandes, des résultats de confirmation de réception, etc.
- **Checksum** : Il s'agit d'un calcul arithmétique qui combine l'identifiant du paquet, la longueur du paquet et l'intégralité de son contenu.

Le tableau 3.2 présente en détail les caractéristiques techniques du module AS608, intégré à notre système biométrique :

Nom du module	AS608 (lecteur optique d'empreintes digitales)
Tension de fonctionnement	3,3 à 5 V DC
Interface de communication	Série TTL
Débit en bauds	9600 à 57600 (par défaut : 57600)
Courant nominal	120 mA
Temps de capture d'image	< 1 seconde
Capacité de stockage	162 empreintes
Taille d'un template	512 octets
Taux du faux positif (FAR)	< 0,001 % (niveau de sécurité 3)
Taux du faux négatif (FRR)	< 0,1 % (niveau de sécurité 3)
Niveau de sécurité	1 à 5 (de faible à élevé)
Température de fonctionnement	-20 à +50 °C
Fenêtre de détection	16 x 18 mm
Dimensions du module	56 x 20 x 21 mm

TABLE 3.2 – Caractéristiques techniques du lecteur optique d'empreintes AS608. [7]

3.3.4 Choix de la suite chaotique

L'article[14] propose une suite hyperchaotique appelée Lotka–Volterra adaptée au chiffrement des empreintes. Lotka-Volterra est un système dynamique qui permet de générer des suites chaotiques à partir des équations suivantes :

$$x_{n+1} = \alpha x_n(1 - x_n) - \beta x_n y_n \tag{3.1}$$

$$y_{n+1} = \delta x_n y_n \tag{3.2}$$

d'où $x \in (0, 0.87)$ et $y \in (0, 1)$ et ses paramètres de contrôle : $\alpha = 3.5,\ \beta > 1.4$ et $3.7 < \delta < 4$.

Le système génère deux séquences pseudo-aléatoires x et y, utilisées respectivement pour créer l'effet de la confusion et l'effet de la diffusion sur le template d'empreinte à crypter. L'article démontre également que les valeurs de x et y sont améliorées pour une meilleure répartition aléatoire.

3.3.5 Algorithme de chiffrement

L'empreinte d'un nouveau utilisateur doit être enregistrée sous format chiffré afin de préserver sa confidentialité. De ce fait, dans notre système d'accès nous avons repris le méme algorithme de chiffrement/déchiffrement sur [14] avec un changement concernant la création du vecteur de permutation CV.

Au début, comme expliqué précédemment la clé personnelle (PK) et la clé de hachage SHA-1 (HK) sont fusionnées par une opération XOR binaire pour dériver la clé PHK . Cette dernière est ensuite divisée en six segments en chiffres hexadécimaux afin de déterminer les valeurs suivantes :

$$A = \frac{PHK_{1-8}}{2^{32}} \tag{3a}$$

$$B = \frac{PHK_{9-16}}{2^{32}} \tag{3b}$$

$$C = \frac{PHK_{17-24}}{2^{32}} \tag{3c}$$

$$D = \frac{PHK_{25-32}}{2^{32}} \tag{3d}$$

$$E = \frac{PHK_{33-40}}{2^{32}} \tag{3e}$$

Deux séquences chaotiques Lotka-Volterra $x \in (0, 1)$,$y \in (0, 1)$ de taille identique N=512 sont générés avec une précision de 10^{-15}.en utilisant l'équation (3.1) et (3.2) avec les paramètres de controle suivants : $\alpha = 3.5$, $\beta = 3.2$, $\delta = 3.9 + (A \times B \times C)/10 \mod 1$, $x = (D \times E)/10 \mod 1$, et $y = 0.2$.

Afin d'améliorer la distribution des élements de x et y de manière pseudo-aléatoire nous effectuons les changements suivants sur chaque élement des séquences :

$$X_n = 100 \times x_n \mod 1 \tag{3.3}$$

$$Y_n = 100 \times y_n \mod 1 \tag{3.4}$$

avec $n = 1, 2, 3, \ldots, N$ et X,$Y \in (0, 1)$ sont les données chaotiques avec une distribution renforcée.

Ces nouvelles séquences X et Y sont exploités pour créer de la confusion via un vecteur de confusion CV et pour provoquer la diffusion de l'erreur par un autre vecteur de difusion DV. Les vecteurs CV/DV sont obtenus comme suit :

$$\text{permvect} = \text{argsort}(X_n) \tag{3.5}$$

$$CV_n = \text{permvect}[1 : N] \tag{3.6}$$

$$DV_n = \text{round}(255 \times Y_n) \tag{3.7}$$

argsort(X(1 :N)) est une fonction qui permet d'obtenir la nouvelle position de chaque indice des éléments du vecteur X après que ses éléments soient réorganisés dans un ordre ascendant ou descendant.

Les nouvelles positions des indices sont stockés dans un vecteur de permutation permvect[1 : N] afin de les exploiter dans le processus de permutation.

Ainsi, le vecteur CV permet de changer la position d'un élément du template de l'empreinte alors que le vecteur DV permet de changer sa valeur. Cela s'effectue au cours du processus de chiffrement où un tour de confusion et de diffusion est appliqué au template de l'empreinte PT à l'entrée selon l'équation suivante :

$$ET_n = PT(CV_n) + DV_n \pmod{256} \tag{3.8}$$

avec $n = 1, 2, 3, \ldots, N$, $ET \in [0, 255]$ est le template chiffré, et PT est le template en clair.

Le processus de chiffrement est résumé sur l'Algorithme 1.

Enfin, la clé de cryptage comprend le token (HK) et la clé personnelle (PK) qui doivent être gardés secrétes.

Algorithm 1: Algorithme de chiffrement

Input: PT, PK
Output: ET

1 : $HK \leftarrow SHA1(PT)$
2 : $PHK \leftarrow PK \oplus HK$
3 : $A \leftarrow PHK(1:8)/2^{32}$
4 : $B \leftarrow PHK(8:16)/2^{32}$
5 : $C \leftarrow PHK(16:24)/2^{32}$
6 : $D \leftarrow PHK(25:32)/2^{32}$
7 : $E \leftarrow PHK(33:40)/2^{32}$
8 : $\alpha \leftarrow 3.5$
9 : $\beta \leftarrow 3.2$
10 : $\delta \leftarrow 3.9 + (A \times B \times C)/10 \mod 1$
11 : $x(1) \leftarrow (D \times E)/10 \mod 1$
12 : $y(1) \leftarrow 0.2$
13 : $X(1) \leftarrow 100x(1) \mod 1$
14 : $Y(1) \leftarrow 100y(1) \mod 1$
15 : **for** $n = 1$ **to** 511 **do**
16 $\quad x(n+1) \leftarrow \alpha x(n)(1 - x(n)) - \beta x(n)y(n)$
17 $\quad y(n+1) \leftarrow \delta x(n)y(n)$
18 $\quad X(n+1) \leftarrow 100x(n+1) \mod 1$
19 $\quad Y(n+1) \leftarrow 100y(n+1) \mod 1$
20 : $permvect \leftarrow \text{argsort}(X(1:512))$
21 : $CV(1:512) \leftarrow permvect[1:512]$
22 : $DV(1:512) \leftarrow \text{round}(255Y(1:512))$
23 : $ET(1:512) \leftarrow PT(CV(1:512)) + DV(1:512) \mod 256$

3.3.6 Algorithme de déchiffrement

Pour pouvoir réaliser l'authentification, le template chiffré ET doit être déchiffré avant de l'envoyer à la mémoire du lecteur pour la procédure de comparaison et validation.

L'algorithme de déchiffrement est identique à l'algorithme de chiffrement précédent sauf que cette fois en entrée on a le template chiffré ET au lieu de PT.

La différence en déchiffrement réside seulement à l'étape 23 de l'algorithme 1 qui doit être remplacé par l'équation suivante pour obtenir le template déchiffré DT.

$$DT(CV(1:512)) \leftarrow ET(1:512) - DV(1:512) \mod 256 \tag{3.9}$$

Sachant que le template déchiffré DT est identique au template original PT seulement si la clé de déchiffrement est identique à la clé de chiffrement.

3.4 Résultats des simulations du chiffrement

En vue de préparer le systéme étudié à une implémentation et une réalisation pratique, nous avons d'abord simuler le comportement des suites chaotiques Lotka-Voltera et les algorithmes précédents de chiffrement et de déchiffrement. Pour cela, nous avons utilisé MATLAB, un environement de calcul scientifique dédié aux traitements numériques, à la modélisation et à la simulation.

Nous avons généré divers graphiques tels que les diagrammes de bifurcation de Lotka-Volterra, son attracteur, ses exposants de Lyapunov, ainsi que les histogrammes relatifs aux phases de chiffrement et de déchiffrement.

3.4.1 Attracteur Lotka-Volterra

La figure 3.4 illustre l'attracteur généré par le système chaotique de Lotka-Volterra obtenu dans le plan (X,Y) en utilisant les paramètres suivants :$\alpha = 3.5$, $\beta = 3.2$, $\delta = 3.9$, des conditions initiales : $x_0 = 0.5$, $y_0 = 0.2$ et nombre d'itérations : 5000.

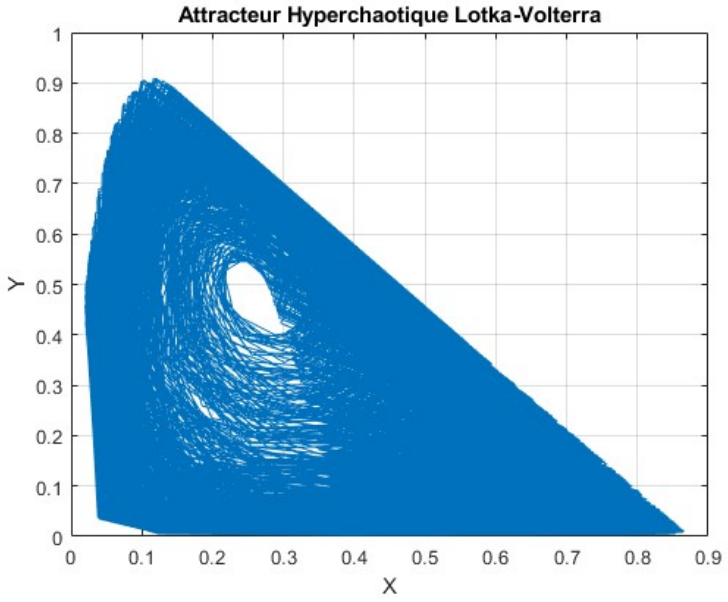


FIGURE 3.4 – Attracteur Lotka-Volterra

3.4.2 Diagramme de bifurcation Lotka-Volterra

Figure 3.5 montre les graphes de bifurcation du modèle hyperchaotique de Lotka-Volterra, démontrant la réactivité du système à diverses valeurs de ses paramètres dynamiques (α, β, δ) à travers six sous-diagrammes : X et Y en relation avec chacun de ces paramètres. Ces graphiques illustrent l'évolution des variables X et Y après un nombre considérable d'itérations, lorsque nous modifions l'un des paramètres tout en maintenant les autres constants.

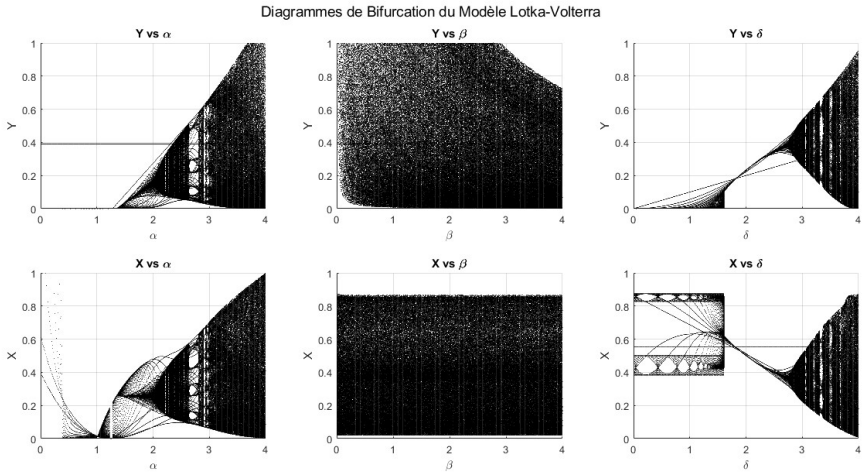


FIGURE 3.5 – Diagramme de bifurcation Lotka-Volterra

3.4.3 Exposant de Lyaponov

Pour analyser le comportement dynamique du système de Lotka-Volterra modifié, nous avons calculé les exposants de Lyapunov en utilisant les paramètres suivants : $\alpha = 3{,}5$, $\beta = 3{,}42$, $\delta = 3{,}8$, avec les conditions initiales $x(0) = 0{,}2$ et $y(0) = 0{,}3$. Le nombre d'itérations utilisé est $N = 5000$. Les résultats obtenus montrent deux exposants de Lyapunov strictement positifs, à savoir $\lambda_1 = 0{,}1583$ et $\lambda_2 = 0{,}0780$, indiquant un comportement **hyperchaotique** du système. La figure 3.6 présente l'évolution des deux exposants de Lyapunov en fonction du nombre d'itérations, montrant une convergence stable vers leurs valeurs respectives.

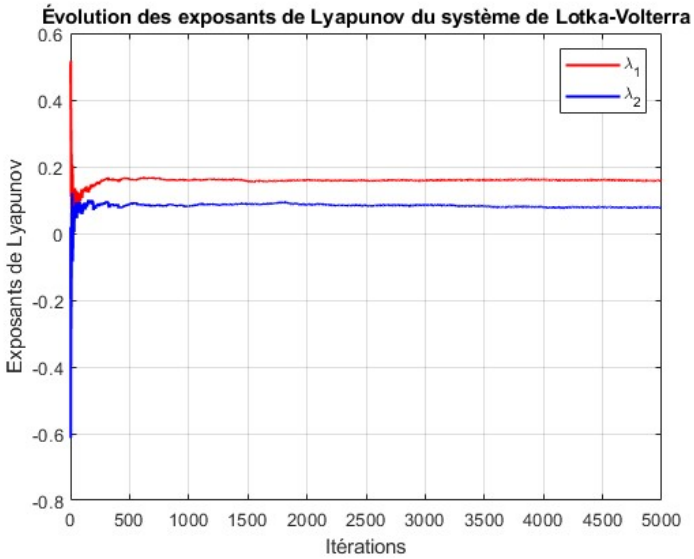


FIGURE 3.6 – Évolution des exposants de Lyapunov λ_1 et λ_2 du système de Lotka-Volterra en fonction du N

3.4.4 Histogramme

Un histogramme est un graphique composé de barres verticales, utilisé pour représenter la distribution d'un ensemble de données numériques. Chaque barre correspond à un intervalle de valeurs, et sa hauteur indique la fréquence des données dans cet intervalle.

Les histogrammes de la séquence originale x et de la séquence améliorée X et Les histogrammes de la séquence initiale y et de la séquence améliorée Y sont respectivement illustrés dans la figure 3.7.

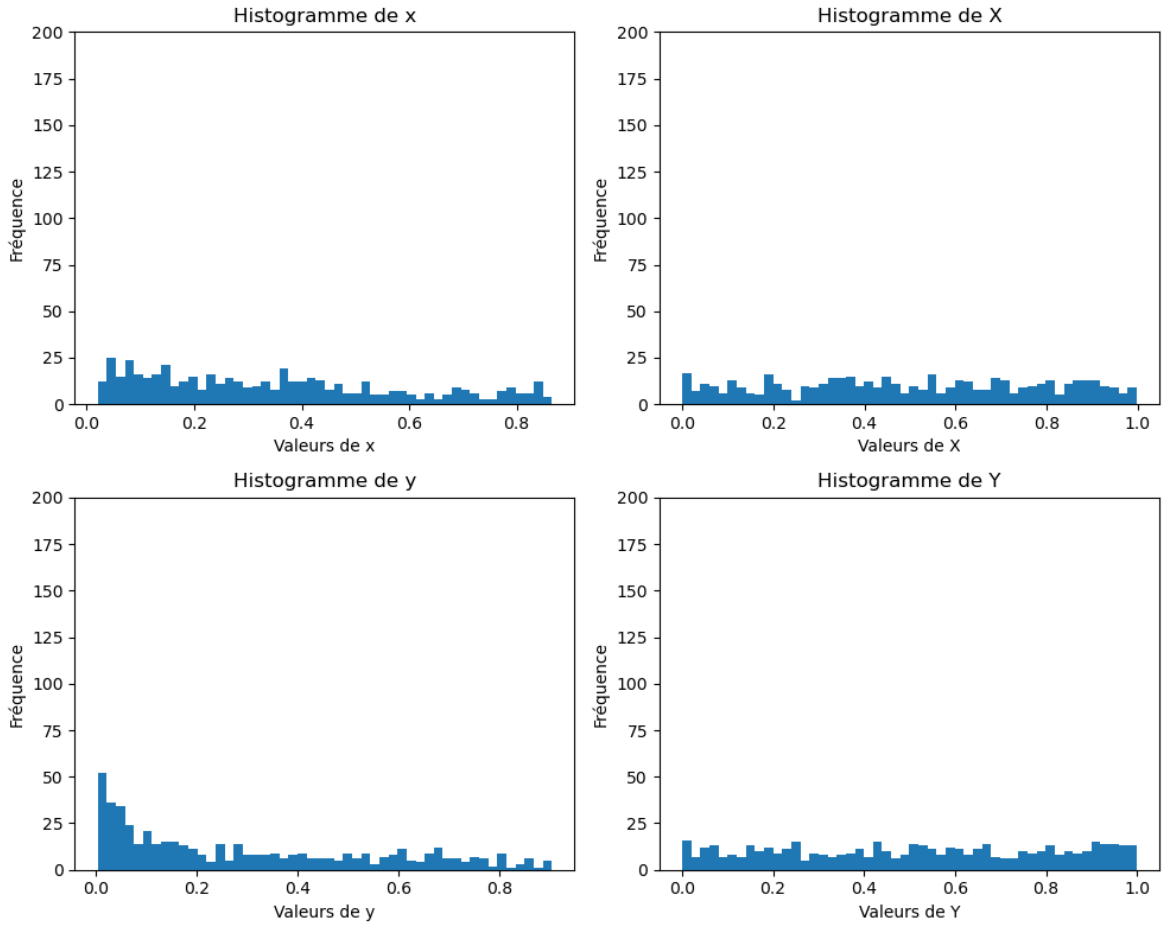


FIGURE 3.7 – Histogramme des séquences originales(x,y) et améliorées(X,Y)

3.5 Conclusion

Dans ce chapitre, nous avons présenté un système biométrique sécurisé basé sur le chiffrement des modèles ou templates des empreintes digitales présents dans la plupart des capteurs d'empreintes. Confrontés à la sensibilité des informations biométriques non chiffrées, nous avons suivi une approche qui s'appuie sur l'usage d'un capteur spécifique le DY50 et d'une méthode chaotique pour le chiffrement des empreintes avec la suppression de l'em-

preinte de la mémoire du lecteur pour éviter toute fuite à l'extérieur. Les résultats de simulation montrent les performances théoriques du chiffrement chaotique. Pour confirmer ces résultats en pratique, une implémentation et une réalisation pratique est nécessaire.

Chapitre 4

Implémentation et réalisation pratique

4.1 Introduction

Ce chapitre présente les résultats obtenus lors de la réalisation pratique du système étudié. Il décrit brièvement l'environnement matériel et logiciel utilisé, ainsi que les principaux modules.

Une présentation détaillée du système de controle d'accés sécurisé réalisé est fournie avec les algorithmes implémentés pour la gestion du fonctionnement du systéme et ses périphériques.

4.2 Environnement de développement

4.2.1 Hardware

Pour la partie pratique nous avons utilisé un Raspberry PI 3B+ comme le coeur du systéme. Le Raspberry PI 3B+ est un nano-ordinateur monocarte qui offre un CPU Quad-core Cortex-A53 (64-bit) cadencé à 1.4 GHz, il s'agit d'une version améliorée du Raspberry pi 3, offrant de meilleures performances. Il est utilisé dans de nombreux domaines : la robotique, la domotique,IOT et les systèmes embarqués. [22]

Le choix du Raspberry Pi 3B+ dans notre système de sécurité s'explique par ses capacités de traitement avancé, son support réseau, son environnement de développement riche, et sa compatibilité avec les périphériques USB et GPIO.

Cette figure 4.1 présente la carte Raspberry Pi 3 Model B+ ainsi que ses divers éléments matériels et leurs rôles respectifs.

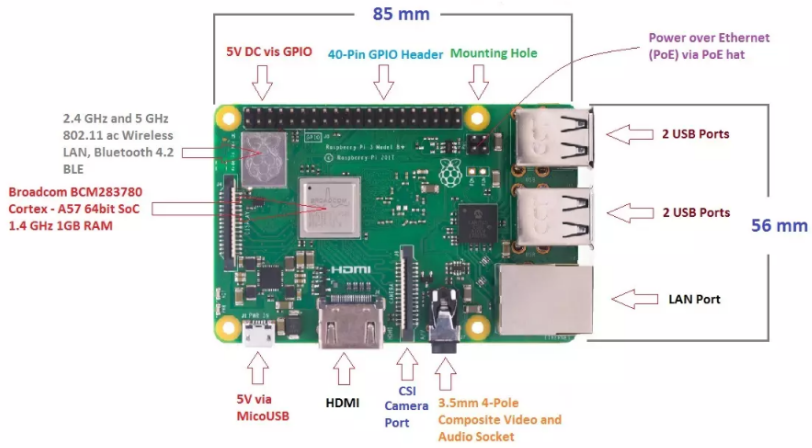


FIGURE 4.1 – Différents composants matériels de Raspberry pi 3 B+ [22]

Le tableau 4.1 présente les caractéristiques principales du Raspberry pi 3B+ :

La figure 4.2 montre le brochage GPIO de Raspberry pi 3B+.

Élément	Spécification
Processeur	Broadcom BCM2837B0, Quad-core Cortex-A53 (64-bit) à 1.4 GHz
RAM	1 Go LPDDR2 SDRAM
Wi-Fi	802.11 b/g/n/ac (double bande 2.4/5 GHz)
Bluetooth	Bluetooth 4.2, BLE
Ethernet	Gigabit Ethernet (via USB 2.0, 300 Mb/s max)
Ports USB	4 × USB 2.0
Sorties vidéo/audio	HDMI pleine taille, prise jack 3.5 mm
GPIO	40 broches compatibles HAT
Alimentation	5V / 2.5A via micro-USB
Carte SD	Slot microSD pour le système et les données

TABLE 4.1 – Spécifications techniques d'un modèle de Raspberry Pi

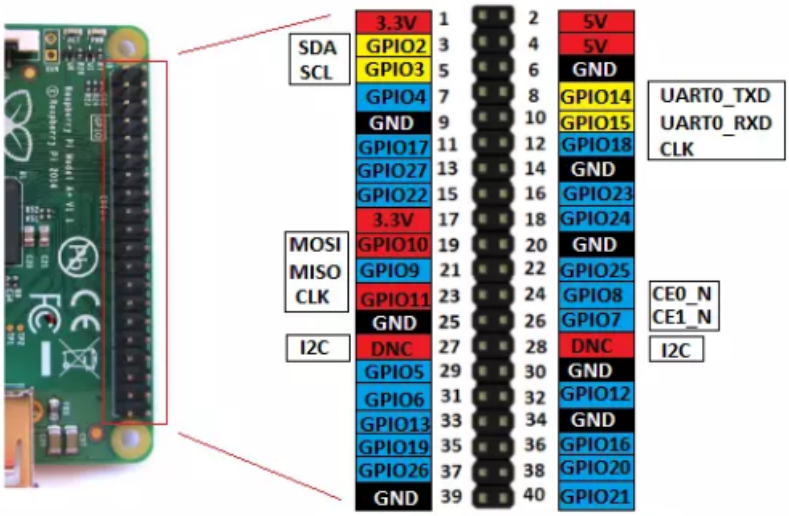


FIGURE 4.2 – Brochage de Raspberry pi 3 B+
[22]

4.2.2 Software

L'implémentation a été réalisé en langage Python, en utilisant l'environnement de développement Spyder, intégré à l'écosystème scientifique Anaconda, afin d'assurer la portabilité et la vérification des résultats sur un poste de travail.

Sur Raspberry Pi, nous avons utilisé Thonny Python, un environnement de développement intégré (IDE) léger et simple, particulièrement conçu pour le système d'exploitation Raspberry Pi OS. Ce choix se justifie par sa facilité d'utilisation, son intégration directe avec Python et la possibilité d'exécuter les scripts développés dans d'autres environnements tels que Spyder.

Des tests statistiques, notamment l'entropie, l'autocorrélation et d'autres indicateurs de performance, ont également été réalisés en PYTHON et sur MATLAB.

4.3 Architecture du systéme réalisé

Figure 4.3 montre la structure de base du système de contrôle d'accès biométrique réalisée qui fonctionne grâce au processeur central (CPU) du Raspberry PI coordonnant divers périphériques.

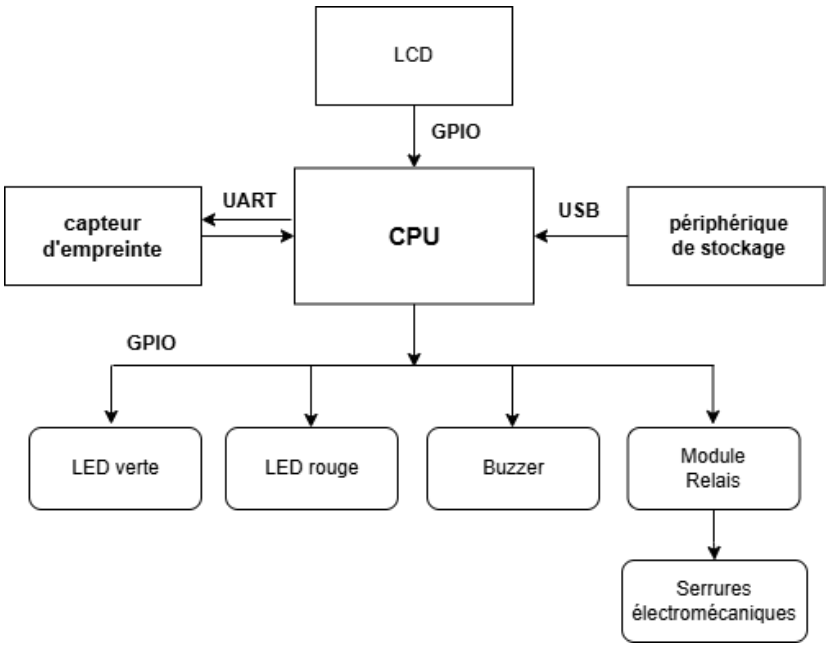


FIGURE 4.3 – Architecture générale du système de contrôle d'accès biométrique

Le capteur d'empreintes digitales est relié au processeur via une interface UART afin de capturer et de transmettre les informations biométriques.

Un écran LCD est connecté via les broches GPIO et est utilisé pour montrer à l'utilisateur des messages d'instruction ou de résultat.

Le dispositif comprend aussi un support de stockage connecté par USB, servant à conserver le hash key HK (TOKEN) faisant partie de la clé de sécurité.

Le processeur gère deux voyants LED (l'une verte et l'autre rouge), un buzzer (alarmes sonores) ainsi qu'un module relais à travers ses broches GPIO pour délivrer un feedback visuel et sonore et gérer l'ouverture et la fermeture de la serrue.

La lumière verte s'active lorsque l'accès est permis, la rouge en cas de refus, tandis que l'avertisseur sonore indique les opérations ou les erreurs.

Pour sa part, le module relais a la capacité de déclencher un appareil externe tel que l'ouverture d'une porte.

Ce dispositif offre donc une gestion globale de l'authentification par empreinte digitale, associant sécurité, notification et interaction avec l'utilisateur.

4.3.1 Module relais

Le module relais à quatre canaux comprend quatre relais de 5V et les éléments nécessaires pour la commutation et l'isolation, simplifiant ainsi l'interfaçage avec un microcontrôleur ou un capteur, en exigeant peu de composants et de connexions. [23]

Chaque relais est conçu pour gérer 250 V en courant alternatif (AC) et 30 V en courant continu (DC), supportant une intensité de 10 A dans les deux situations. [23]

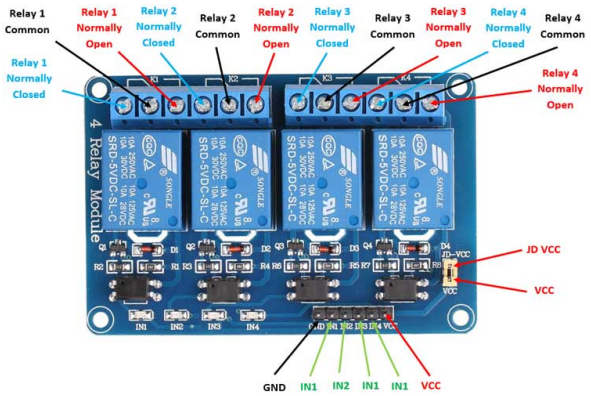


FIGURE 4.4 – Brochage du module relais à quatre canaux [23]

4.3.2 LCD 1602A

Nous avons utilisé un écran LCD 1602A, un module alphanumérique à 16 colonnes et 2 lignes, pouvant présenter 32 caractères . Ce contrôleur HD44780 gère l'affichage, rendant son utilisation plus facile avec des microcontrôleurs. [23]

Il comporte 16 broches, permettant son alimentation, sa commande et la transmission des données.Le brochage de cet écran est illustré dans la figure 4.5.

Les caractéristiques de l'écran LCD HD44780 :

— La tension de fonctionnement est comprise entre 4,7 V et 5,3 V

— Consommation de courant : 1 mA (sans rétroéclairage).

— Chaque caractère est formé dans une boîte de 5×8 pixels.

— Fonctionne en mode 8 bits ou 4 bits.

— Peut également afficher des caractères personnalisés générés par l'utilisateur.

— Disponible avec un rétroéclairage vert ou bleu.

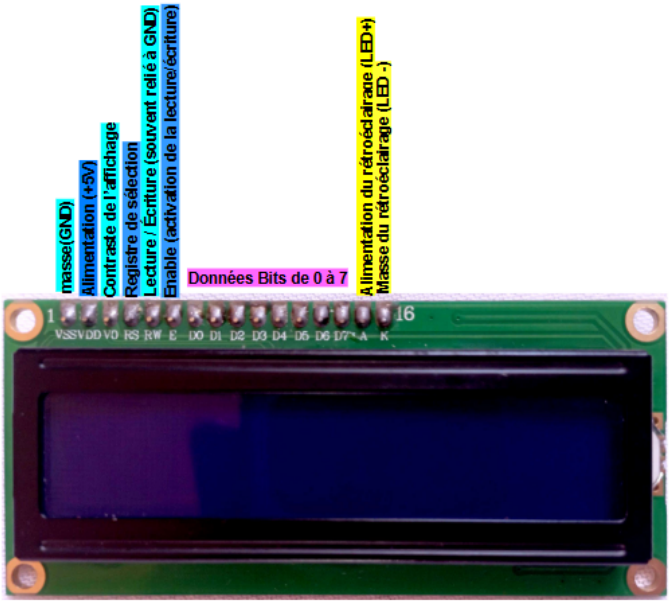


FIGURE 4.5 – Brochage de module LCD 1602A

4.4 Algorithmes de gestion implémentés

4.4.1 Algorithme principal

Figure 4.6 présente le logigramme de l'algorithme principal utilisé dans notre projet.

Il est essentiel de commencer par l'importation des bibliothèques requises pour établir les connexions, gérer les dispositifs et exécuter les fonctions biométriques.

Cela comprend les modules pour la communication série, le traitement d'empreintes digitales, les opérations de déchiffrement et de lecture de fichiers, l'interfaçage avec le capteur ainsi que le contrôle des dispositifs (LED, buzzer, relais, LCD).

Une clé de hachage est placée dans une liste noire en cas d'échec sur trois tentatives d'authentification ce qui permet de bloquer l'accès aux utilisateurs non autorisés en cas de perte du TOKEN.

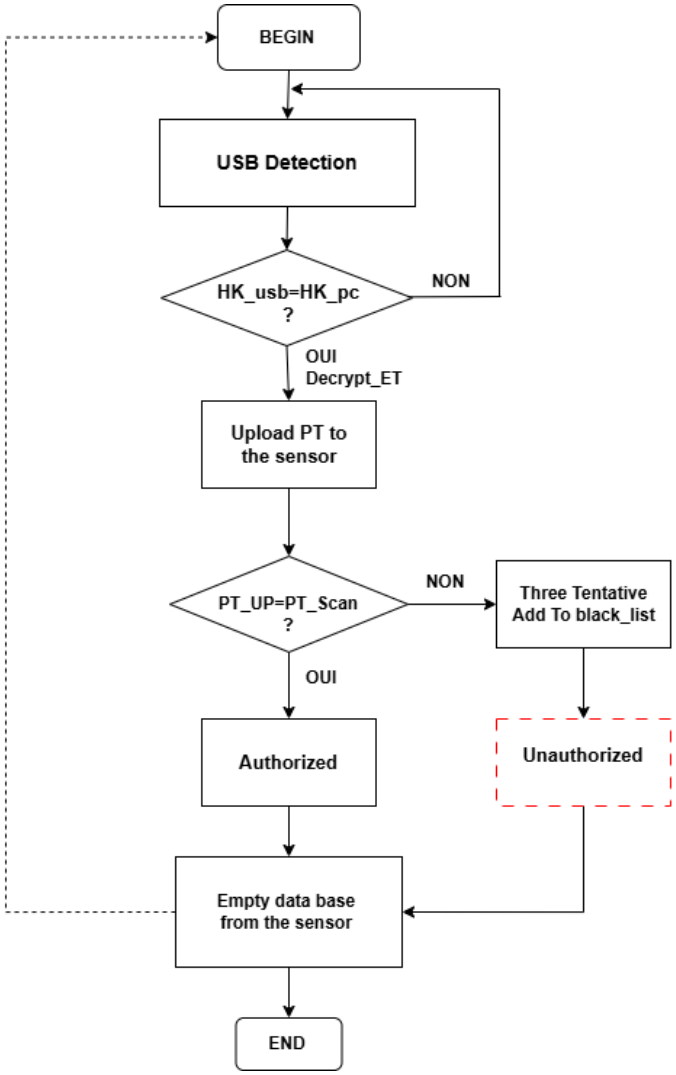


FIGURE 4.6 – Logigramme de l'algorithme principal

Le systéme nécessite divers éléments d'interface pour la mise en route et la visualisation, incluant deux voyants lumineux (une verte et une rouge), un relais, un avertisseur sonore, ainsi qu'un écran LCD comme interface homme machine.

Figure 4.7 montre comment initialiser les périphériques de sortie.

— La LED verte sert à montrer que l'accès a été accordé, alors que la LED rouge indique un refus ou une erreur d'authentification.

— Le relais autorise la gestion d'un dispositif externe (tel qu'une serrure ou un système d'ouverture) lorsque l'accès est confirmé.

— Le buzzer produit un son en cas de non-authentification ou pour indiquer certaines conditions du système.

— Le moniteur LCD présente des messages explicites pour orienter l'utilisateur durant le processus d'identification et indiquer la condition actuelle du système.

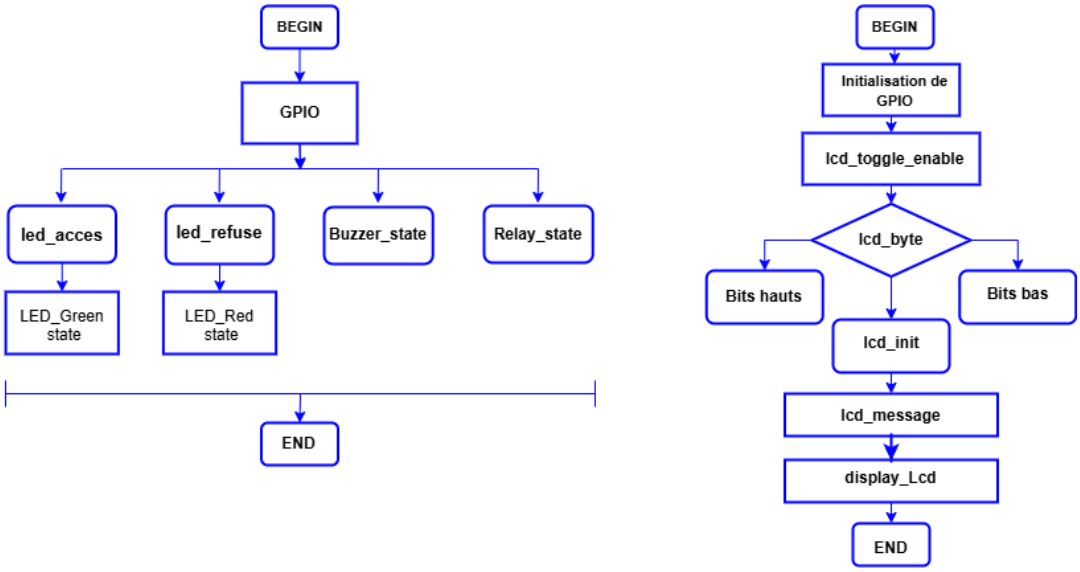


FIGURE 4.7 – Logigramme d'initialisation et de gestion des périphériques de sortie

4.4.2 Détection / Vérification du TOKEN

Dans cette partie, nous expliquons comment effectuer la lecture depuis la clé USB ainsi que depuis le fichier sauvegardé localement sur le système (CPU). Ce processus est illustré dans la Figure 4.8.

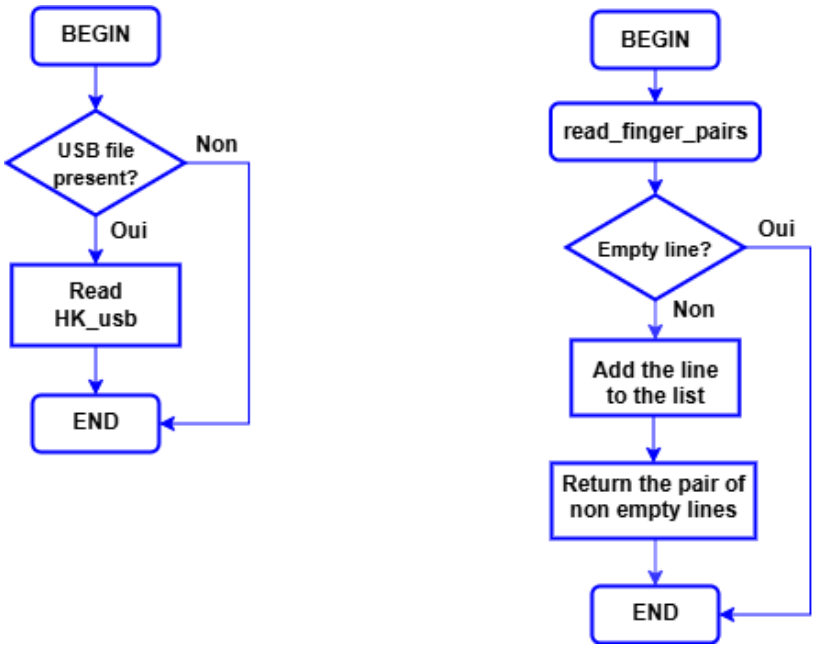


FIGURE 4.8 – Logigramme de lecture de la clé HK et vérification des données locales

Il peut être résumé comme suit :

— Le dispositif est en attente de l'insertion d'une clé USB.

— Si aucune clé valide n'est détectée, un message d'erreur est affiché régulièrement sur l'écran.

— Dès qu'une clé est reconnue, le système en extrait la clé de hachage (HK).

— Cette HK est ensuite comparée à celles enregistrées dans le fichier finger.txt, stocké localement sur le système.

4.4.3 Déchiffrement du template

Le processus de déchiffrement du template chiffré (ET) grâce à un algorithme chaotique basé du système Lotka-Volterra est illustré dans la figure 4.9 ci-dessous.

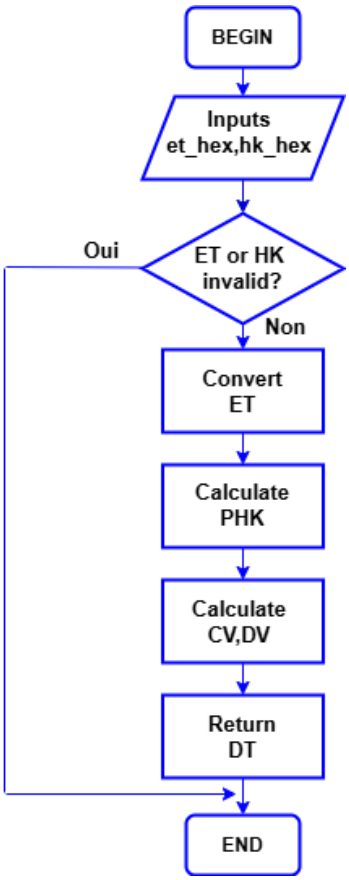


FIGURE 4.9 – Logigramme de la fonction de déchiffrement

Cette démarche est essentielle dans notre dispositif biométrique pour reconstituer le template original (PT) à partir de sa version cryptée, en utilisant deux clés : une clé personnelle (PK) et une clé de hachage (HK) extraite de la clé USB.

Le logigramme de flux souligne les phases cruciales du processus :

— Contrôle de la validité des longueurs ET et HK.

— Création d'une clé intermédiaire PHK par l'application d'un XOR entre PK et HK.

— Initialisation des conditions du système Lotka-Volterra.

— Création de deux vecteurs chaotiques : CV (permutation) et DV (diffusion).

— Reconstitution du modèle d'origine par inversion du chiffrement.

Notre système est basé sur un algorithme de chiffrement/déchiffrement de lotka volterra. Nous détaillerons ce processus dans une section prochaine.

4.4.4 Transfert du template déchiffrée

la figure 4.10 ci-dessous illustre l'intégralité du processus d'envoi du template déchiffré DT vers le capteur DY50, employé dans notre système biométrique.

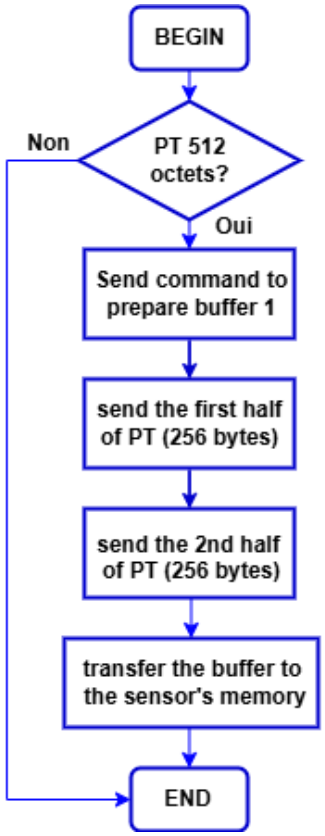


FIGURE 4.10 – Logigramme de la fonction de chargement de template déchiffrer

Ce logigramme montre les diverses phases d'interaction entre CPU et le capteur à travers le port série, en suivant le protocole particulier du capteur DY50.Cette procédure inclut :

— La validation du format du modèle (512 octets).

— Le partage du modèle en deux parties.

— Le transfert progressif des données vers le buffer 1 du capteur, suivi du déplacement de ce buffer vers la mémoire interne du capteur.

4.4.5 Comparaison des empreintes

La figure 4.11 illustre les quatre phases essentielles de la procédure :

— L'attente de la détection d'un doigt par le capteur.

— La transformation de l'image en caractéristiques biométriques.

— La comparaison avec l'empreinte DT déchiffré précédemment et stockée temporairement dans la mémoire du capteur.

— La suppression de toutes les empreintes une fois la comparaison effectuée pour éviter la fuite des données biométriques.

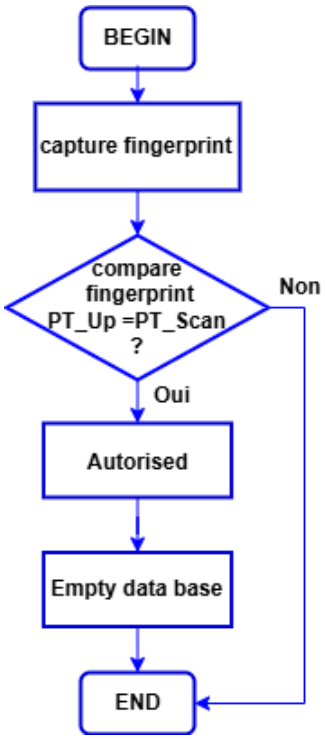


FIGURE 4.11 – Logigramme de vérification et de réinitialisation du capteur

4.5 Tests pratiques

Pour extraire les informations du modèle non chiffrées à partir de l'empreinte, nous avons utilisé une carte Arduino UNO a cause de sa simplicité d'utilisation, de sa compatibilité avec de nombreux capteurs, comme présenté sur la figure 4.12.

Le capteur DY50 a été configuré avec une vitesse de communication série (baud rate) de 9600 bauds, un format de données de 8 bits, sans parité, et un bit d'arrêt. L'alimentation a été assurée par la broche 3.3V de l'Arduino, et les broches RX/TX du capteur ont été connectées respectivement aux broches 2,3 de la carte Arduino .

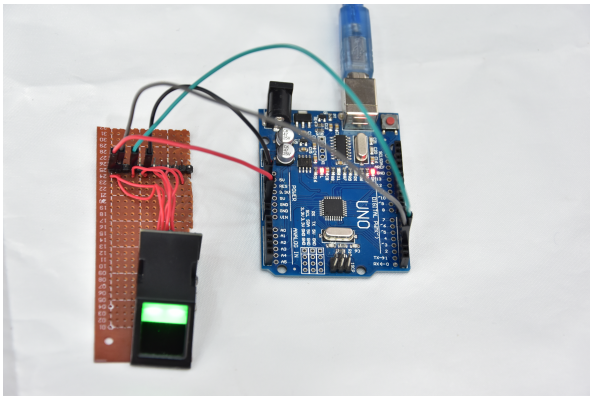


FIGURE 4.12 – Connexion entre PC, une carte Arduino et le capteur d'empreinte

Nous avons connecté le capteur d'empreintes digitales DY50 au PC via une connexion USB afin d'effectuer des tests de chargement (upload) et d'extraction du template (PT), en utilisant un code Python développé sous l'environnement Spyder (voir la figure 4.13).

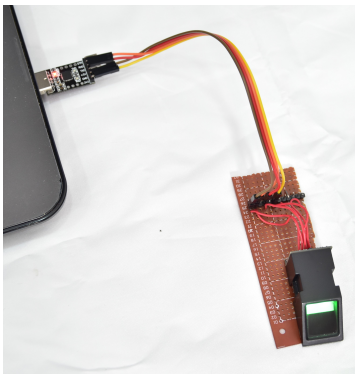


FIGURE 4.13 – Connexion entre PC et le capteur d'empreinte

Nous avons également extrait l'image de l'empreinte digitale afin de permettre une visualisation directe des données capturées par le capteur, comme illustré à la figure 4.14

FIGURE 4.14 – Image d'empreinte digitale extraite à partir du capteur

Figure 4.15 illustre Le système prototype utilisé pour les résultats expérimentaux . Le système propose un accès sécurisé basé sur l'empreinte digitale.

Le prototype est basé sur un Raspberry Pi 3B+, mais peut être adapté à d'autres plateformes embarquées comme les microcontrôleurs 8/16 bits ou les FPGA, avec une intégration des modules nécessaires (capteur, mémoire, interface).

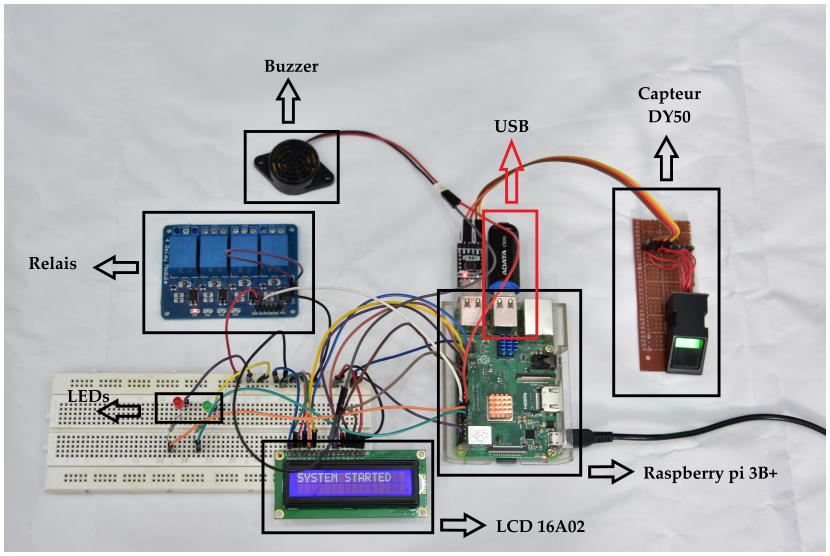
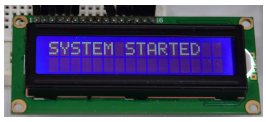
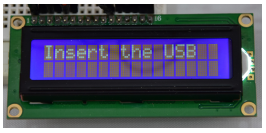


FIGURE 4.15 – Prototype monté pour les résultats expérimentaux

Les figures suivantes illustrent les différents messages affichés par l'écran LCD au cours du fonctionnement du système :



(a) Démarrage

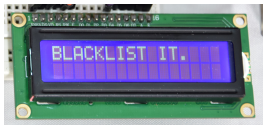


(b) Entrée clé USB

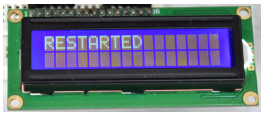
(c) Poser votre doigt

(d) Autorisation

(e) Non autorisé



(f) Ajout à la blacklist

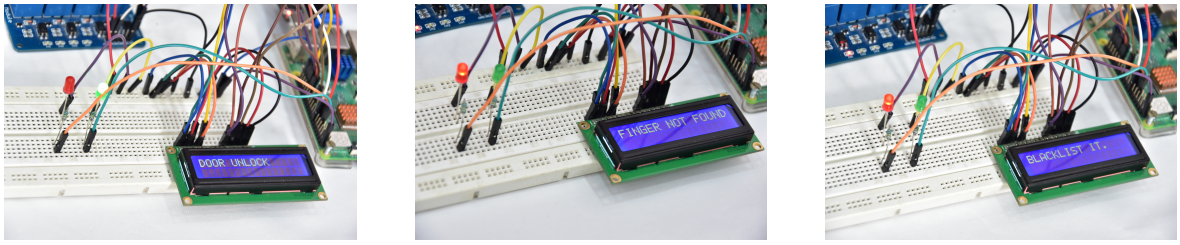


(g) Redémarrage

FIGURE 4.16 – Affichages successifs sur l'écran LCD au cours du fonctionnement du système biométrique.

Figure 4.17 illustre le comportement visuel du système lors du processus d'authentification. Lorsque l'utilisateur est reconnu et que l'accès lui est autorisé, une LED verte s'allume pour indiquer que la porte peut être ouverte.

Si l'utilisateur est non reconnu ou inscrit sur la liste noire, une LED rouge s'active pour signaler un refus d'accès. Ce retour visuel assure une interaction rapide et intuitive avec l'utilisateur.



(a) LED verte activée (b) LED rouge activée (c) Blacklist (accés refusé)

FIGURE 4.17 – Illustration du retour visuel du système selon le statut de l'utilisateur.

4.6 Analyse de sécurité

Afin d'analyser la sécurité du systéme, nous employons deux modèles distincts d'empreintes digitales, affichés en hexadécimal (512 nombres décimaux sur 8 bits) dans le tableau 4.2.

Empreint digitale	**Plain template (les premiers 250 hexadecimal de 512 digitales)**
Utilisateur 1	0303571E000120017B00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000090002007B00000000000000000000000000000000000000000000000011EDEF01FFFFFFFF0200824B084E1E209B9E7E5D158CDF381BA1BF2DA5DF1F6E2DA47F21AF5DBF60AFD01F3A341B9F55B8133F343CDA
Utilisateur 2	0303611640014A019400000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000120008007B00010101010101010101010101010101010101132BEF01FFFFFFFF020082511E6CBE6B9E951E59A22C3E3D27029E6F31955E29B3027E3D34589E29B79ADE61BFC05E4AC818DE3C4C82

TABLE 4.2 – Plain template (les premiers 250 hexadecimal de 512 digitales)

La figures 4.18 illustrent la répartition des données des 512 valeurs décimales sur 8 bits pour l'utilisateur 1 et l'utilisateur 2, respectivement.

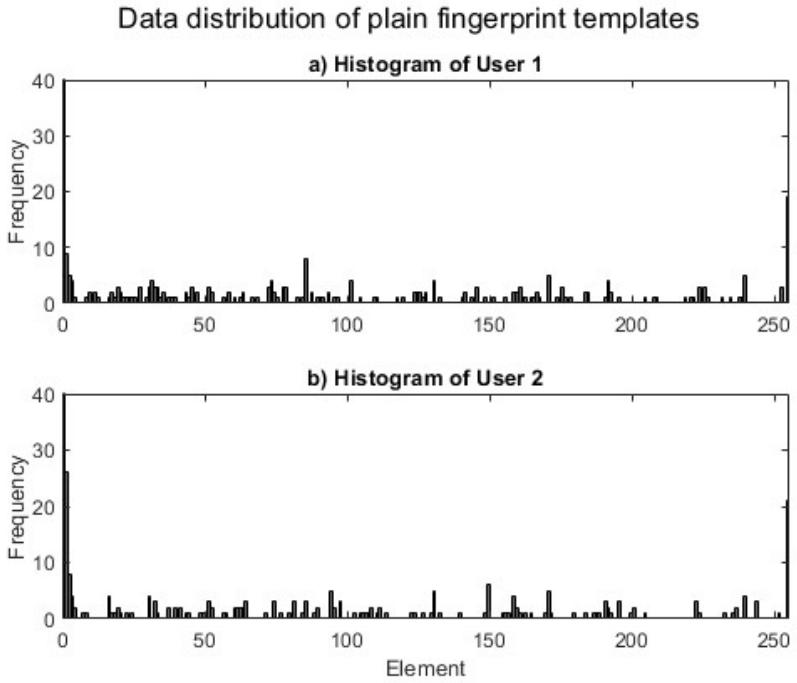


FIGURE 4.18 – Histogramme de plain template de l'utilisateurs 1 et 2

Les clé de hashage et les clés secrètes individuelles (PK) de 160 bits employées dans le processus de chiffrement, ainsi que les clés issues de l'opération XOR (PHK), sont exposées dans le Tableau 4.3 pour l'Utilisateur 1 et l'Utilisateur 2. Le système hyperchaotique de Lotka-Volterra est initialisé à l'aide de la clé PHK.

Empreinte digitale	**Clé de hachage (HK) de 160 bits en hexadécimal**
Utilisateur 1	6aa3dfce939bece58b5359cefc25af922c63716a
Utilisateur 2	fbf0f26acbdc676e15d36b9991b67946b7f0068a
	Clé personnelle (PK) de 160 bits en hexadécimal
Utilisateur 1	d4e5f67890abcdef1234567890a1b2c3d4e5f678
Utilisateur 2	b3529a498e5c823bd6710400a7b331fc07b32ac1
	Clé XOR (PHK) de 160 bits en hexadécimal
Utilisateur 1	be4629b60330210a99670fb66c841d51f8868712
Utilisateur 2	48a268234580e555c3a26f99360548bab0432c4b

TABLE 4.3 – Clés de hachage, clés personnelles et clés XOR des utilisateurs

Le Tableau 4.4 présente les 250 premiers chiffres hexadécimaux des modèles chiffrés pour l'Utilisateur 1 et l'Utilisateur 2 et la figure 4.19 présentent ces répartition des données.

Empreinte digitale	Template chiffré (les premiers 250 hexadecimal de 1024 digitales)
Utilisateur 1	2D3D76D8667995630DD0E32213D6D991966197A 6E78277F4ED64803F7C69443824AFE9E9F22627 DCF027439419324C20531E15F2656B2460E5AF6 6AE1E504B65E45566C155DDD3F474D83B83E81D E50476A0AFA1C6DBFF9CB5D952F8679EB9287E3 9E25D9F85EEE29870705B1D58D6696994130198 31B0AE7A65759634
Utilisateur 2	F269CB1A958AF862FC79765458A4AC01DE7BEB2 C5DF80165245A9805E98727B0D595A3DB567EDB 0C9C6982320552070BFED4040A652680E363294 1B0D0EC8FBDEA0EB18CD9E4EABBDB8DCC606856 7654E27C4F79CE0BAF638C5B5694CC70E130FF9 B911E92C158145EE655128226A5F9BCF102D488 7F346C29116B9F14

TABLE 4.4 – Template chiffré (les premiers 250 hexadecimal de 1024 digitales)

4.6.1 Sensibilité de la clé secrète

L'une des propriétés essentielles du chaos est qu'une légère modification des conditions initiales provoque une variation significative dans la trajectoire des systèmes chaotiques.[14]

Dans cette analyse, le modèle d'empreinte digitale en clair de l'Utilisateur 1 est utilisé à deux reprises (la clé de hachage étant identique) pour générer deux cryptogrammes, cependant avec des clés privées (PK) similaires définies dans le Tableau 4.5, ont été utilisées : par exemple, un simple changement du caractère '**c**' en '**a**' dans la clé hexadécimale à la position 13.

Ce test permet de vérifier si une légère modification de la clé personnelle (PK) entraîne une différence significative dans le résultat chiffré.

Clé	Clé personnelle utilisée pour l'analyse de sensibilité
clé 1	d4e5f67890ab**c**def1234567890a1b2c3d4e5f678
clé 2	d4e5f67890ab**a**def1234567890a1b2c3d4e5f678

TABLE 4.5 – Clés secrètes personnelles utilisées pour l'analyse de sensibilité

La figure 4.20 montre que le chiffrement est très sensible à la clé privée (PK) de 160 bits. En utilisant deux clés presque identiques avec la même empreinte, les résultats chiffrés sont très différents, cela démontre que notre système est fortement sensible à la clé, ce qui renforce sa sécurité.

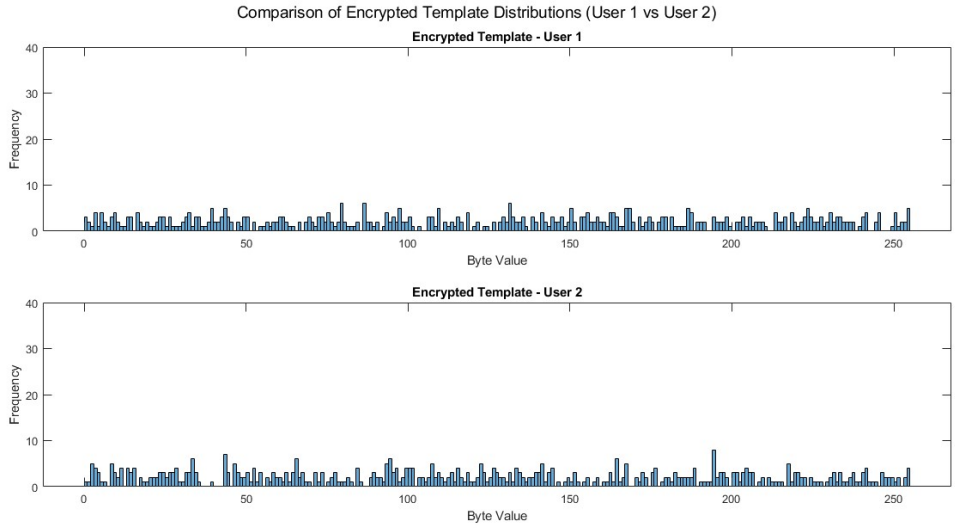


FIGURE 4.19 – Histogramme de template chiffrée de l'utilisateurs 1 et 2

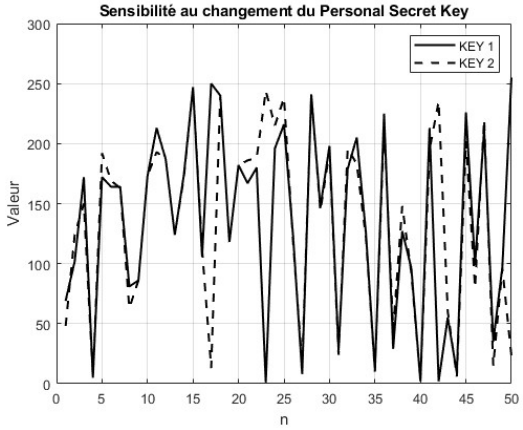


FIGURE 4.20 – Sensibilité à la clé dans le processus de chiffrement

4.6.2 Sensibilité du template PT

Nous utilisons NPCR qui mesure le porcentage d'éléments différents entre deux séquences de même taille [14]. il est défini par la formule suivant :

$$\text{NPCR} = \frac{100}{512} \sum_{n=1}^{N=512} W_n \quad \text{où} \quad W_n = \begin{cases} 0 & \text{si } C1_n = C2_n \\ 1 & \text{si } C1_n \neq C2_n \end{cases} \tag{4.1}$$

Et UACI pour évaluer l'intensité moyenne des différences entre deux séquences [14] , et est calculé selon :

$$\text{UACI} = \frac{100}{512} \sum_{n=1}^{N=512} |C1_n - C2_n| \tag{4.2}$$

Où $C1$ et $C2$ représentent les deux cryptogrammes. Dans ce test, on utilise ET_1 de l'Utili-

sateur 1 présenté dans le Tableau 4.4 comme $C1$. On effectue alors une modification sur le premier chiffre hexadécimal du modèle non chiffré de l'Utilisateur 1 (Tableau 4.2), le faisant passer de `03` à `13`, soit de $030357_{16}\ldots$ à $130357_{16}\ldots$. Cela génère une nouvelle clé de hachage, et donc un nouveau $C2$.[14]

Les résultats obtenu est : NPCR= 99.80% et UACI= 32.17% ; Ce résultat indique la haute sensibilité du schéma de chiffrement proposé aux moindres changements du modèle non chiffré.

4.6.3 Analyse statistique

L'histogramme (Voir figure 4.18) du PT révèle une forte concentration de certaines valeurs, ce qui le rend sensible aux attaques par analyse statistique. Par contre, le modèle basé sur un template chiffré (Voir figure 4.19) affiche une répartition uniforme, ce qui renforce la protection de l'algorithme contre ce genre d'attaque.[14]

4.6.4 Calcul de la fréquence flottante

Le test de fréquence flottante sert à déterminer si le code présente des points faibles, en calculant le nombre de symboles distincts (entre [0, 255]) qui apparaissent dans des segments de 256 éléments.[14]

Ce test est utilisé pour vérifier si le code de chiffrement produit un résultat bien réparti et difficile à deviner. En analysant la fréquence des symboles utilisés dans les données chiffrées, on peut détecter s'il existe des motifs répétitifs ou des faiblesses. Si les symboles sont bien répartis, cela signifie que le chiffrement est efficace et rend les données plus difficiles à décrypter par un attaquant.

dans la figure 4.21 a) l'analyse de fréquence révèle que le modèle non chiffré(PT) ne comporte que 26.51% des symboles possibles, alors que dans la figure 4.21 b) le modèle chiffré (ET) en inclut 63.35%. Donc l'algorithme de chiffrement répartit les valeurs de manière plus efficace et n'a pas de points faibles, ce qui améliore sa sécurité.

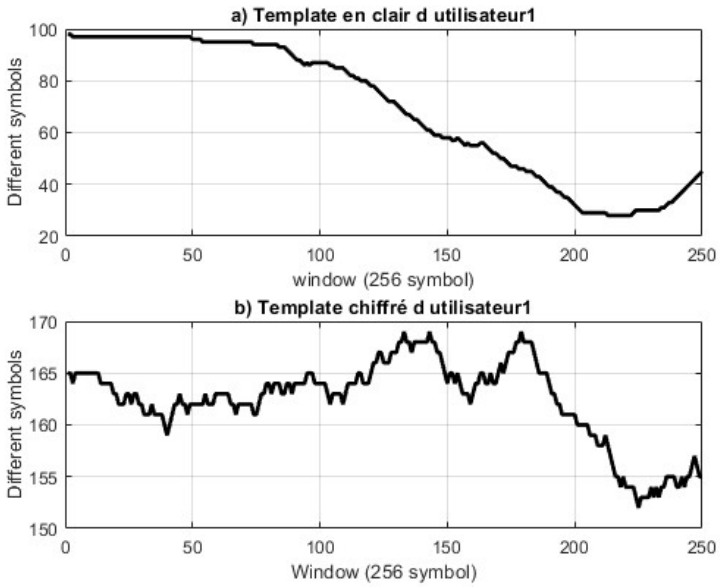


FIGURE 4.21 – Sensibilité à la clé dans le processus de chiffrement

4.6.5 Autocorrélation

L'autocorrélation sert à identifier des motifs récurrents, des répétitions ou des dépendances dans un modèle chiffré.[14]

Elle est décrite par la formule suivante :

$$AC(k) = \frac{A - D}{T} \tag{4.3}$$

où :

— **A** est le nombre d'éléments identiques entre le message initial et sa version décalée de k positions.

— **D** est le nombre d'éléments différents.

— **T** est la longueur totale du message.

Ce test permet de vérifier si certaines parties se répètent à intervalles réguliers, ce qui peut indiquer une faiblesse dans le chiffrement. La formule utilisée compare le message à lui-même, mais décalé de quelques positions. Si le message est bien chiffré, il ne doit pas présenter de régularité.

La figure 4.22 illustre l'autocorrélation pour le premier utilisateur, où le modèle non chiffré (ligne en pointillés) présente des motifs récurrents, alors que le modèle chiffré (ligne continue) ne révèle aucun motif récurrent. Ainsi,le template chiffré est fortement pseudo-aléatoire.

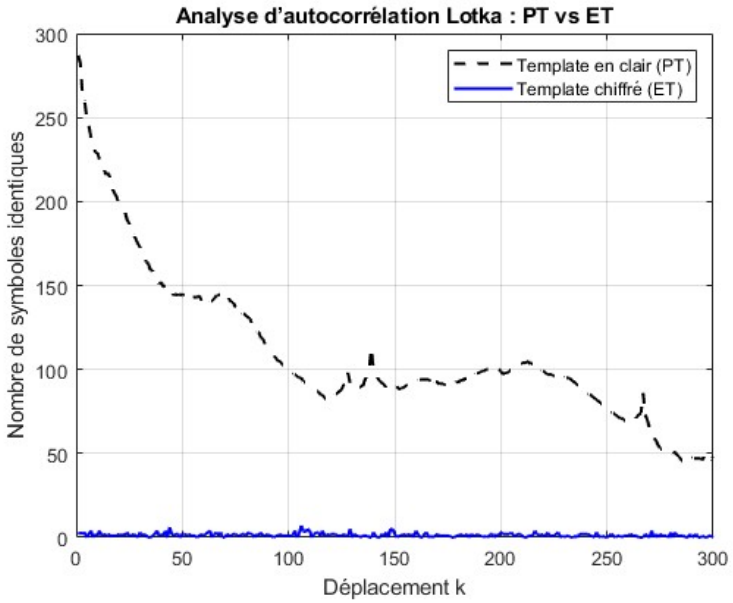


FIGURE 4.22 – Analyse de l'autocorrélation de l'utilisateur 1

4.6.6 Entropie

Entropie permet d'évaluer le niveau d'imprévisibilité d'un message [14], peut être calculée comme suit :

$$H(m) = \sum_{i=0}^{2^N-1} p(m_i) \log_2 \left(\frac{1}{p(m_i)} \right) \tag{4.4}$$

d'où :

— N est le nombre de bits utilisés pour représenter l'unité de base du message m,

— 2^N correspond au nombre total de combinaisons possibles de cette unité,

— $p(m_i)$ désigne la probabilité d'apparition de la valeur m_i,

— $\log_2$ est le logarithme en base 2.

Plus l'entropie est élevée, plus le contenu est difficile à analyser ou à prédire, ce qui est souhaitable en cryptographie.

L'entropie de modéle non chiffré (PT) d'utilisateur 1 est H(P)=3.9273 bits , tandis que de modéle chiffré (ET) est H(E) = 7.6172 bits. Ainsi, le processus de diffusion produit un degré élevé de désordre pour se défendre contre une attaque par entropie.

4.7 Etude comparative

Nous avons mené une étude comparative entre les systèmes chaotiques de Lotka-Volterra et de Lorenz. L'algorithme suivant a été utilisé pour étudier le comportement du système de Lorenz :

Algorithm 2: Algorithme de chiffrement basé sur Lorenz

Input: PT (template d'empreinte), PK (clé personnelle)

Output: ET (template chiffré)

$HK \leftarrow$ SHA-1(PT) ;
$PHK \leftarrow PK \oplus HK$;
$A \leftarrow$ PHK$(1{:}8)/2^{32}$;
$B \leftarrow$ PHK$(8{:}16)/2^{32}$;
$C \leftarrow$ PHK$(16{:}24)/2^{32}$;
$D \leftarrow$ PHK$(25{:}32)/2^{32}$;
$E \leftarrow$ PHK$(33{:}40)/2^{32}$;
$x(1) \leftarrow \text{mod}(D + E, 1) + 1$;
$y(1) \leftarrow \text{mod}(A + C, 1) + 1$;
$z(1) \leftarrow \text{mod}(B + E, 1) + 1$;
$\sigma \leftarrow 10, \quad r \leftarrow 28, \quad \beta \leftarrow \frac{8}{3}$;
for $n = 1$ **to** 511 **do**
    $x(n+1) \leftarrow x(n) + \sigma \cdot (y(n) - x(n))$;
    $y(n+1) \leftarrow y(n) + r \cdot x(n) - y(n) - x(n) \cdot z(n)$;
    $z(n+1) \leftarrow z(n) + x(n) \cdot y(n) - \beta \cdot z(n)$;
    $X(n) \leftarrow \text{mod}(x(n), 1)$;
    $Y(n) \leftarrow \text{mod}(y(n), 1)$;
: $permvect \leftarrow$ argsort$(X(1:512))$
: $CV(1:512) \leftarrow permvect[1:512]$
: $DV(1:512) \leftarrow$ round$(255Y(1:512))$
: $ET(1:512) \leftarrow PT(CV(1:512)) + DV(1:512) \mod 256$
: $DT(CV(1:512)) \leftarrow ET(1:512) - DV(1:512) \mod 256$

Le système chaotique de Lorenz utilisé dans notre méthode de chiffrement est défini par les équations suivantes [24] :

$$\begin{cases} \dfrac{dx}{dt} = \sigma(y - x) \\ \dfrac{dy}{dt} = x(r - z) - y \\ \dfrac{dz}{dt} = xy - \beta z \end{cases} \tag{4.5}$$

avec les paramètres standards [24] : $\sigma = 10$, $r = 28$, et $\beta = \frac{8}{3}$.

Les conditions initiales (x_1, y_1, z_1) sont dérivées de deux clés (Hks, Pks) par une opéra-

tion XOR, puis converties en réels A, B, C, D, $E \in [0, 1]$. Elles sont fixées comme suit :

$$\begin{cases} x(1) = \mathrm{mod}(D + E, 1) + 1 \\ y(1) = \mathrm{mod}(A + C, 1) + 1 \\ z(1) = \mathrm{mod}(B + E, 1) + 1 \end{cases} \tag{4.6}$$

Cette formulation permet de garantir des valeurs initiales strictement positives, évitant ainsi les points fixes ou comportements non chaotiques.

Le système de Lorenz est chaotique mais pas partout. Certaines valeurs comme x(1) = 0, y(1) = 0, ou z(1) = 0 peuvent amener le système à des zones de comportement régulier ou peu intéressant.

Contrairement aux premières versions on a fait mod(100*x(1),1), la normalisation est appliquée à toute la trajectoire :

$$X_n = mod(xn, 1) \tag{4.7}$$

$$Y_n = mod(yn, 1) \tag{4.8}$$

Ce qui permet d'extraire des séquences de permutation (CV) et de diffusion (DV) plus riches et plus aléatoires, renforçant ainsi la sécurité du chiffrement.

Les résultats de comparaisons obtenus sont présentés comme suit :

4.7.1 Sensibilité de la clé

Cette figure illustre la sensibilité au changement de la clé personnelle (PK) pour les systèmes chaotiques de Lorenz et de Lotka-Volterra.

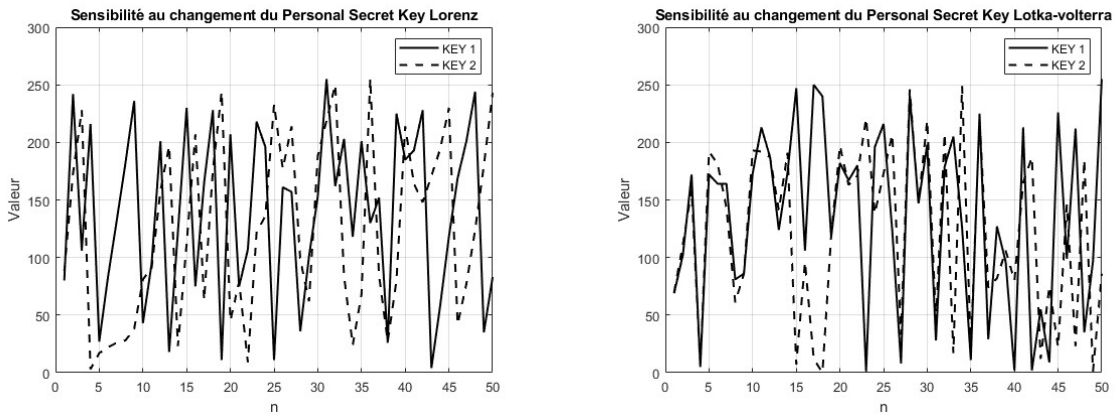


FIGURE 4.23 – Comparaison de la sensibilité au changement de la clé personnelle (PK)

Lorenz montre une plus grande sensibilité au changement de la clé personnelle, ce qui est positif pour la sécurité cryptographique .

Lotka-Volterra est aussi sensible, mais la divergence est un peu moins violente visuellement.

4.7.2 Fréquence flottante

Dans l'analyse de fréquence flottante, le template en clair montre environ 26,5% de symboles différents dans une fenêtre de 256 octets, ce qui est identique pour Lorenz et Lotka-Volterra, car le même template est utilisé.

Après chiffrement, Lotka-Volterra atteint 63,35% de symboles uniques contre 61,25% pour Lorenz. Cela montre que Lotka-Volterra disperse mieux les octets, améliorant la complexité locale des données.

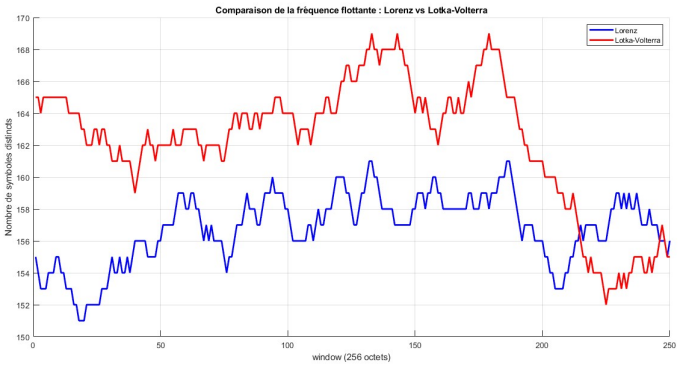


FIGURE 4.24 – Comparaison de la fréquence flottante

4.7.3 Entropie

Dans l'analyse d'entropie, le template en clair présente une valeur identique de 3,9273 bits pour les deux systèmes, ce qui est attendu puisqu'il s'agit du même template. Après chiffrement, l'entropie passe à 7.6172 bits avec Lotka-Volterra, contre 7,5334 bits avec Lorenz. Cela indique que Lotka-Volterra produit un chiffrement légèrement plus aléatoire, avec une meilleure répartition des symboles.

4.7.4 Autocorrélation

Pour l'analyse d'autocorrélation les deux systèmes (Lotka et Lorenz) assurent un très faible niveau d'autocorrélation après chiffrement, ce qui indique une bonne diffusion de l'information et une résistance aux attaques statistiques(Voir la figure 4.25).

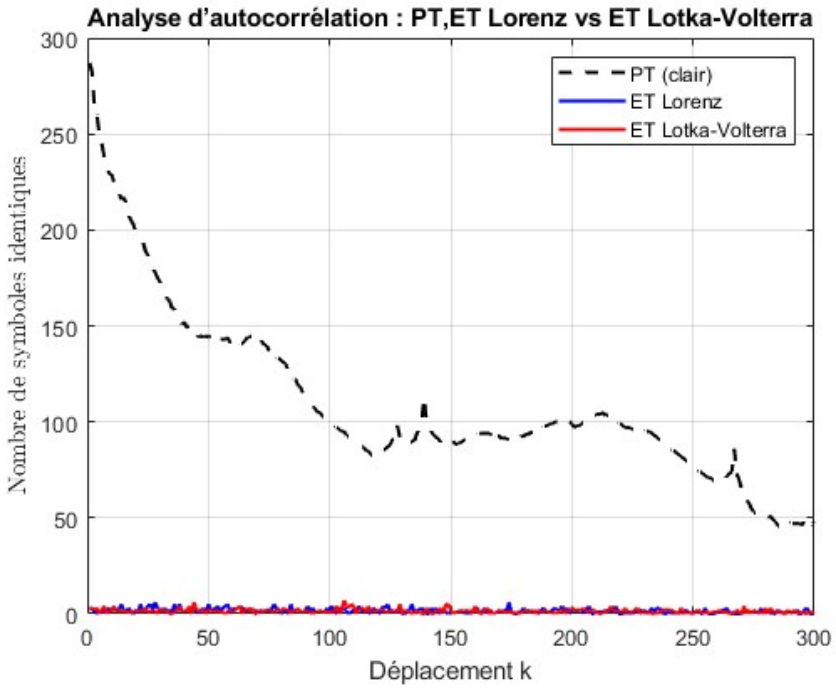


FIGURE 4.25 – Comparaison de l'autocorrélation

La figure 4.26 illustre un agrandissement sur la séquence chiffrée ET, permettant de mieux visualiser la différence de comportement entre les systèmes de Lorenz et de Lotka-Volterra dans l'analyse d'autocorrélation.

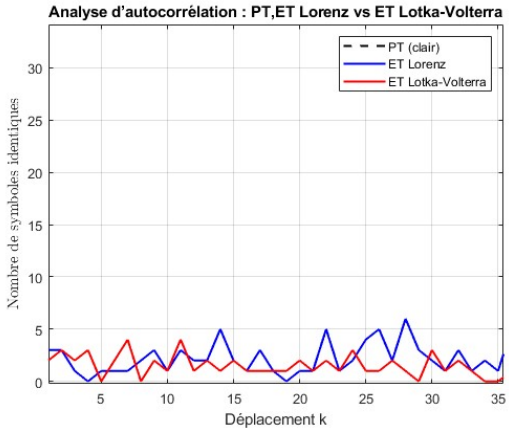


FIGURE 4.26 – Agrandissement sur la séquence chiffrée ET

4.7.5 Sensibilité du template PT

Tableau 4.6 montre Les résultats d'analyse de sensibilité du template non chiffré de Lotka-volerra et Lorenz :

Lotka-Volterra présente une meilleure réactivité binaire (NPCR), idéale pour détecter

Algorithme	NPCR moyen (%)	UACI moyen (%)
Lotka	99.80	32.17
Lorenz	94.53	34.15

TABLE 4.6 – La sensibilité des algorithmes Lotka et Lorenz selon NPCR et UACI

toute variation, même minime, dans les données en entrée.Lorenz est plus performant en termes de diffusion des valeurs (UACI), ce qui augmente la complexité de prédiction pour un attaquant.

4.7.6 Résumé comparatif

Le tableau 4.7 présente un résumé comparatif des résultats obtenus, mettant en évidence les points forts de chaque méthode selon les indicateurs de sécurité choisis :

Critère analysé	Conclusion comparative
Entropie du template en clair (PT)	Identique pour les deux systèmes (template fixe)
Entropie du template chiffré (ET)	Lotka-Volterra → meilleure entropie (7.6172 bits vs 7.5334 bits)
Autocorrélation (visuelle)	Lotka → meilleure rupture de régularité, structure chaotique plus marquée
Analyse de la fréquence flottante du PT	Identique (26.51%) pour les deux, car le template source est le même
Analyse de la fréquence flottante du ET	Lotka → meilleure diffusion locale (63.35% vs 61.25%)
Test NPCR	Lotka → plus grande sensibilité (99.80% vs 94.53%)
Test UACI	Lorenz → variations d'intensité plus marquées (34.15% vs 32.17%)

TABLE 4.7 – Résumé comparatif entre les algorithmes Lotka-Volterra et Lorenz

4.8 Conclusion

Dans ce chapitre, nous avons présenté l'environnement matériel et logiciel utilisé, ainsi que le système de contrôle d'accès biométrique développé. Les résultats de simulation du système chaotique, notamment Lotka-Volterra, ont démontré une bonne diffusion et une grande sensibilité aux clés. Les tests pratiques ont validé le bon fonctionnement du système : de la détection de la clé USB, au chargement du template d'empreinte, jusqu'à la comparaison finale. L'analyse de sécurité a confirmé la robustesse du système. La comparaison entre les résultats simulés et pratiques montre une bonne cohérence et confirme les performances du système proposé.

Conclusion générale et perspectives

Dans ce mémoire nous avons étudié et réalisé un système d'accès par empreinte digitale intégrant un chiffrement chaotique basé sur le modèle de Lotka-Volterra et en associant des composants accessibles et en garantissant un haut niveau de sécurité dans un environnement embarqué.

Le chiffrement repose sur la combinaison d'un clé personnelle PK et d'un jeton HK combinés pour produire une clé secréte PHK qui servira pour la génération de séquences chaotiques.

Le système assure un chiffrement réversible pour les utilisateurs autorisés, tout en étant résistant aux attaques. Les tests pratiques ont confirmé sa robustesse à travers des les tests de l'entropie, l'autocorrélation, NPCR, UACI et la distribution des élements du template chiffré.

Le temps d'exécution global du système a été estimé à environ 2,5 secondes. Ce temps a été calculé à partir du moment où l'utilisateur insére son jeton jusqu'à ce que l'accés lui soit accordé. Cela le rend adapté à des applications embarquées.

Une comparaison entre les systèmes chaotiques de Lotka-Volterra et celui de Lorenz a été menée pour évaluer leur efficacité respective dans le processus de chiffrement. Cette analyse a renforcé la pertinence du choix du modèle chaotique Lotka-Voltera.

En conclusion, ce travail a contribué à la promotion de solutions de sécurité biométrique avancées, fiables et économiquement viables. Il montre que la chiffrement chaotique n'est pas seulement une théorie mais un outil pratique et efficace pour répondre aux protection des données personnelles.

De plus, ce travail offre plusieurs perspectives d'amélioration :

— L'intégration d'autres modalités biométriques (iris, visage).
— L'ajout de la technologie RFID pour une double authentification.
— Le stockage sécurisé en cloud, l'amélioration de l'interface utilisateur.
— L'emploi de modèles chaotiques plus complexes.

Bibliographie

[1] M.El-Abed,"Évaluation de systèmes biométriques", Thèse de doctorat, Université de Caen, 2011.

[2] N. Galy,"Étude d'un système complet de reconnaissance d'empreintes digitales pour un capteur microsystème à balayage", Thèse de doctorat,Institut National Polytechnique de Grenoble, 2005.

[3] S. Ramisetty et R. B.,"Implementation of AES using biometric", International Journal of Electrical and Computer Engineering (IJECE), vol. 9, no. 5, pp. 4266–4276, octobre 2019

[4] N. Kumar, S. Mallick, S. K. Sood, and Y. Singh, "Energy-Efficient Data Dissemination for Healthcare Systems in Sustainable Smart Cities", Sustainability, vol. 10, no. 10, p. 3588, octobre 2018.

[5] S. Akrouf,"Une approche multimodale pour l'identification du locuteur" , Thèse de doctorat, Université Sétif, 2014.

[6] D. Petrovska-Delacrétaz, C. Garcia, D. B. Dorizzi, "Guide to Biometric Reference Systems and Performance Evaluation", Londres : Springer-Verlag London Limited, 2009.

[7] Handson Technology,"AS608 Optical Finger Print Sensor Module", Handson Technology, 2020.

[8] J. Jallouli,"Sécurité des systèmes embarqués : contribution à la modélisation des attaques par canaux auxiliaires et à la conception de contre-mesures",Thèse de doctorat, Rennes, France, 2017

[9] A. J. Menezes, P. C. van Oorschot, et S. A. Vanstone,"Handbook of Applied Cryptography", CRC Press, 2018, p. 4.

[10] M. Videau, "Critères de sécurité des algorithmes de chiffrement à clé secrète", Thèse de doctorat, Université Pierre et Marie Curie (Paris 6), France, 2005.

[11] Y.C. Lin et C.H. Wang,"A Study on AES-based Encryption Scheme with Face Recognition Mechanism", Ministry of National Defense, Taiwan, Technical Report, juin 2024.

[12] E. Cherrier, "Estimation de l'état et des entrées inconnues pour une classe de systèmes non linéaires", Thèse de doctorat, Institut National Polytechnique de Lorraine, Vandoeuvre-lès-Nancy,page91.France, 2006.

[13] A. Yahi,"Développement d'Algorithmes de chiffrement d'Images à Base des Suites Chaotiques",Thèse de doctorat, Université Mohamed El Bachir El Ibrahimi de Bordj Bou Arréridj, 2023

[14] M. A. Murillo-Escobar, R. M. López-Gutiérrez, C. Cruz-Hernández, E. E. Espinoza-Peralta, et D. Murillo-Escobar,"Secure access microcontroller system based on fingerprint template with hyperchaotic encryption", Integration, vol. 90, pp. 27–39, 2023.

[15] H. Agaguena et M. Tifouti, "Étude et simulation d'un système de chiffrement d'images à base de chaos", Mémoire de Master, Université 8 Mai 1945 – Guelma, Algérie, juin 2023.

[16] M.Wang, M.Li, Y.Zhang, Q.Liu, and J.Zhang,"A novel 3D chaotic map and its application in image encryption",Communications in Nonlinear Science and Numerical Simulation, vol 8, pp. 216145-216157, 2020.

[17] S. Manuel, "Analyse et conception de fonctions de hachage cryptographiques", Thèse de doctorat, École Polytechnique, Palaiseau, France, 2010.

[18] G. Leurent and T. Peyrin, "SHA-1 is a Shambles,in Proceedings of the 29th USENIX Security Symposium", Boston / Virtual, USA, Aug. 2020.

[19] A. E. Belfedhal, "Étude et implémentation des fonctions de hachage cryptographiques basées sur les automates cellulaires ", Thèse de doctorat, Université Djillali Liabès de Sidi-Bel-Abbès, Sidi-Bel-Abbès, Algérie, 2015.

[20] Z. Qiao, "Nonlinear dynamics, applications to chaos-based encryption", Thèse de doctorat, École Centrale de Nantes, Nantes, France, 2021.

[21] Hangzhou Zhian Technologies Co., Ltd., "ZFM-20 Series Fingerprint Identification Module ", version 1.4, septembre 2008.

[22] A.Naqeel, "Introduction to Raspberry Pi3B+", *The Engineering Projects*, 24 juillet 2018. [En ligne]. Disponible sur : https ://www.theengineeringprojects.com/2018/07/introduction-to-raspberry-pi-3-b-plus.html. [Consulté le 30 mai 2025].

[23] Components101, "Components101 – Electronic Components Pinouts, Datasheets, and Tutorials", *Components101*, datasheet/tutorial.[En ligne]. Disponible sur : https ://components101.com/. [Consulté le 30 mai 2025].

[24] D. Viswanath, "The fractal property of the Lorenz attractor", Physica D : Nonlinear Phenomena, vol.190, pp.115–128, 2004.