UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA

FACULTAD DE INGENIERÍA, ARQUITECTURA Y DISEÑO ENSENADA

DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE ACCESO SEGURO BASADO EN CAOS Y BIOMETRÍA

TESIS

que para cubrir los requisitos necesarios para obtener el grado de

INGENÍERO EN ELECTRÓNICA

presenta:

ERICK ENRIQUE ESPINOZA PERALTA

Ensenada, Baja California, México, Agosto de 2017.

UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA

FACULTAD DE INGENIERÍA, ARQUITECTURA Y DISEÑO ENSENADA

DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE ACCESO SEGURO BASADO EN CAOS Y BIOMETRÍA

TESIS

Que para obtener el grado de Ingeniero en electrónica presenta:

ERICK ENRIQUE ESPINOZA PERALTA

Aprobada por el siguiente comité:

Dr. César Cruz Hernández
Co-director

Dr. Miguel Ángel Murillo Escobar
Co-director

Dra. Rosa Martha López Gutiérrez
Miembro del comité

M.C. José Antonio Michel Macarty
Miembro del comité

Dra. Liliana Cardoza Avendaño
Miembro del comité

Ensenada, Baja California, México. Agosto de 2017

RESUMEN de la tesis de **Erick Enrique Espinoza Peralta**, presentada como requerimiento para obtener el grado de Ingeniero en Electrónica, del programa de licenciatura de la Universidad Autónoma de Baja California. Ensenada, Baja California, México. Mayo de 2017.

DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE ACCESO SEGURO BASADO EN CAOS Y BIOMETRÍA

Resumen aprobado por:

Dr. César Cruz Hernández
Co-director

Dr. Miguel Ángel Murillo Escobar
Co-director

En este trabajo de tesis de licenciatura, se diseña e implementa un sistema de control de acceso seguro biométrico y criptografía caótica, en un sistema embebido basado en un microcontrolador de 32-bits para proteger las plantillas de huellas dactilares de los usuarios y evitar robo de identidad.

Se calcula del exponente de Lyapunov de varios mapas caóticos para determinar cual utilizar en la aplicación criptográfica. Se elige el mapa Lotka-Volterra por presentar dinámica hypercaótica con sólo dos dimensiones, ser de tiempo discreto y tener polinomios de primer orden. Se cambian algunos valores a los parámetros del mapa Lotka-Volterra para escalar el mapa y se utiliza como base el algoritmo de encriptado caótico [15]. Además, se presentan varios análisis de seguridad para validar el algoritmo criptográfico como sensibilidad a la clave, espacio de claves, histogramas, autocorrelación, entropía y frecuencia flotante.

El tiempo de encriptado y desencriptado es aceptable para un sistema de acceso seguro en tiempo real. El sistema embebido construido permite la verificación del usuario y mantiene los templetes de las huellas dactilares encriptadas hasta iniciar el proceso de autenticación.

Palabras clave: caos, biometría, cifrado, análisis de seguridad, microcontrolador, control de acceso.

Abstract of the thesis presented by **Erick Enrique Espinoza Peralta**, as a requirement to obtain the bachelor degree in electronics engineering, of the program of Bachelor's degree of the Autonomous University of Baja California. Ensenada, Baja California, Mexico. May 2017.

DESIGN AND IMPLEMENTATION OF A SECURE ACCESS CONTROL SYSTEM BASED ON CHAOS AN BIOMETRY

Abstract approved by:

Dr. César Cruz Hernández
Co-director

Dr. Miguel Ángel Murillo Escobar
Co-director

In this thesis work, a biometric secure access control system and chaotic cryptography is designed and implemented in an embedded system based on a 32-bit microcontroller to protect the fingerprint templates of users and prevent identity theft.

The calculation of the Lyapunov exponent is applied to several chaotic maps to determine which to use in the criptographic application. The Lotka-Volterra map is chosen because it presents hyper-chaotic dynamics with only two dimensions and it has first-order polynomials. Some values are changed to the Lotka-Volterra map parameters to scale the map and [15] chaotic encryption algorithm is used as the basis. In addition, several statistical security analysis are presented to validate the cryptographic algorithm, such as, sensitivity to the key, key space, histograms, autocorrelation, entropy, and floating frequency.

The encryption and decryption time is acceptable for a real time secure access system. The embedded system allows the verification of the user and maintains the templates of the encrypted fingerprints until the authentication process is started.

Keywords: chaotic cipher, security analysis, microcontroller, embeeded applications.

A mi familia

Agradecimientos

A mi familia por haberme apoyado todo este tiempo a lo largo de mis estudios, por estar siempre atentos, por su cariño y afecto.

Al Dr. César Cruz Hernández, por haberme instruido en este tema tan interesante y por aceptarme como tesista en este proyecto de investigación.

Al Dr. Miguel Ángel Murillo Escobar, por todo el apoyo que me brindó, por estar al tanto de mis estudios, por todos los consejos que me ayudaron a ser una persona más preparada y guiarme en este trabajo.

A la Dra. Rosa Martha López Gutierrez, por la atención que me brindó durante mis estudios, por invitarme a realizar esta tesis y acercarme más a la ciencia.

A mi comite de tesis, Dra. Liliana Cardoza Avendaño, M.C. José Antonio Michel Macarty y Dra. Rosa Martha López Gutierrez por sus consejos y correcciones de tesis.

A la Universidad Autónoma de Baja California (UABC), por brindarme un espacio donde realizar mi formación profesional.

A la Facultad de Ingeniería, Arquitectura y Diseño campus Ensenada, que fue como mi segundo hogar, a mis profesores que siempre tuvieron el gusto por compartir su conocimiento.

Al Consejo Nacional de Ciencia y Tecnología (CONACyT), por el apoyo económico que se me otorgo para el desarrollo de mi trabajo, a través del Proyecto de Grupos de Investigación en Ciencia Básica, Referencia 166654.

Ensenada, B.C., México.
Agosto de 2017

Erick Enrique Espinoza Peralta

Tabla de Contenido

Lista de Figuras

Lista de Tablas

Capítulo 1

Introducción

El los últimos años, el avance tecnológico ha facilitado el desarrollo de sistemas de identificación biométricos, que se postulan para sustituir los métodos tradicionales que se utilizan desde los inicios de la era tecnológica con el nombre de usuario y contraseña. En la antigüedad, el centinela mencionaba el santo y le debía responder con la seña para determinar si la persona tiene permitido el acceso, si no lo conocía se trataba de un intruso [1].

Actualmente se han popularizado los sistemas de control de acceso que utilizan algún tipo de identificador biométrico de forma que se tenga la certeza de que realmente se esta tratando con la persona indicada y. De esta forma, se evita que alguna otra persona tome su lugar ingresando a sitios de suma importancia como puede ser, el acceso a archivos médicos, controles de una planta energética, bancos, aeropuertos, entre otros.

El interés de proteger los datos personales se ha incrementado debido a la facilidad con la que se puede transportar, manipular y distribuir la información. Es por eso que ahora se procede a mantener seguros los datos confidenciales del usuario por medio de la encriptación.

Los sistemas caóticos poseen muchas propiedades que benefician a la encriptación, como alta sensibilidad a condiciones iniciales y a parámetros de control, mezcla de datos, comportamiento tipo aleatorio, no linealidad, entre otros, por lo que un sistema de encriptado caótico puede llegar a ser muy efectivo para el cifrado de la información.

En primera instancia, se utilizó la encriptación caótica analógica, con técnicas de sincronización mediante metodos criptográficos como enmascaramiento caótico (ver figura 1.1.), conmutación caótica, y modulación [2–7].

El uso de *caos digital* tiene la ventaja de que no se requiere establecer la sincronización de un sistema con otro antes de iniciar la transmisión del mensaje. En su lugar, se utilizan las *condiciones iniciales* y *parámetros de control* del mapa caótico para generar la secuencia caótica, las cuales son secretas y solo las podrán recuperar las personas que posean la clave. En la literatura han propuesto aplicaciones de criptografía caótia

Figura 1.1: Esquema de cifrado *analógico* unidireccional empleando *enmascaramiento caótico* con dos sistemas idénticos para lograr establecer la sincronización.

digital para imagenes [8, 9], información biométrica [10–12] y telemedicina [13, 14].

La seguridad de un algoritmo de cifrado basado en caos *digital* depende de el espacio de claves (suficientemente grande para resistir un ataque exhaustivo), en el diseño del algoritmo de cifrado,(que genere texto cifrado con propiedades estadísticas tales que resistan ante los ataques criptoanalíticos conocidos) y que el algoritmo sea factible para aplicaciones en tiempo real [15].

1.1. Motivación

Los identificadores biométricos han ido apareciendo poco a poco en diferentes áreas. La labor de identificar a una persona para permitirle el acceso de algún servicio o acceso, ha ido transicionando al uso de los sistemas embebidos que se encarguen de reconocer a las personas, ya sea por la huella digital, por la voz, por el iris del ojo, reconocimiento del rostro, etc. Estos sistemas permiten una identificación más rápida y con una mayor certidumbre, por la que resulta conveniente contar con este tipo de tecnologías. Sin embargo, el hecho de almacenar la identidad de una persona para poder hacer el proceso de verificación, resulta en un compromiso con la seguridad y la integridad de la información, ya que pudiera ser robada, duplicada o utilizada de alguna manera, para fines fraudulentos.

La gran capacidad de cómputo de los microcontroladores permiten hacer operaciones matemáticas relativamente complejas en cortos periodos de tiempo. Debido a la importancia de proteger la información confidencial que se utilizan en los sistemas biométricos de control de acceso, se propone utilizar cifrado caótico, lo que permite mantener la información biométrica de los usuarios de forma segura en una base de datos o en transmisión por canales de comunicación inseguros.

1.2. Objetivos

Objetivo general: Diseñar e implementar un sistema embebido de acceso seguro basado en caos y biometría.

Objetivos específicos:

1. Determinar el sistema caótico a utilizar para ser implementado en un microcontrolador de 32-bits, con base a su sensibilidad a condiciones iniciales y parámetros de control.

2. Diseñar un algoritmo de encriptado caótico digital para plantillas biométricas con clave simétrica, permutación y difusión.

3. Implementar el algoritmo de encriptado en microcontrolador y evaluar la seguridad y desempeño.

4. Construir el sistema de acceso seguro basado en microcontrolador de 32-bits, lector de huellas dactilares, mando de control, pantalla de cristal líquido y demás componentes.

1.3. Organización del manuscrito

- **Capítulo 1:** Se presenta la introducción de este trabajo de investigación, la motivación y los objetivos.

- **Capítulo 2:** Se presenta una descripción de los sistemas biométricos, el proceso de obtención de minucias y las propiedades de las huellas dactilares.

- **Capítulo 3:** Se presentan los sistemas caóticos utilizados, el mapa caótico elegido en base a las simulaciones de Matlab para usarlo en el algoritmo de encriptado.

- **Capítulo 4:** Se presentan los sistemas de encriptado, algunas propiedades, consideraciones para prevenir ataques y algunas aplicaciones.

- **Capítulo 5:** Se presenta la implementación del algoritmo de encriptado en el microcontrolador y los resultados del sistema de acceso.

- **Capítulo 6:** Se presentan las conclusiones generales y el trabajo futuro.

Capítulo 2

Biometría

En este capítulo se presentan los aspectos fisiológicos y conductuales que son utilizados para identificar a las personas, así como los métodos que han sido utilizados desde sus inicios a la actualidad.

2.1. Introducción

La biometría es la ciencia que analiza los rasgos, distancias y posiciones entre las partes del cuerpo para poder identificar o clasificar a las personas.

En la actualidad, existen distintos rasgos biométricos que se usan para la identificación, pueden ser de tipo *conductual* como la forma de caminar, la forma de teclear, la firma con bolígrafo y la voz. También puede ser de tipo *fisiológica* como la huella digital, el rostro, el iris del ojo, la huella de la mano, al ADN, las venas de la mano, entre otros [16].

Cada una de ellas tiene sus fortalezas y debilidades, por lo que tienen un interés en diferentes áreas según sea el grado de seguridad que se requiera obtener. Por ejemplo, en los *smartphones* es común tener la identificación del rostro para el desbloqueo de pantalla aprovechando la cámara frontal, lo cual resulta más rápido que teclear el PIN. Recientemente han comenzado a utilizar la huella digital, colocando un lector específicamente para esa tarea, con mejores prestaciones al no necesitar iluminación ambiental para que funcione correctamente.

2.2. Sistemas biometricos

Los sistemas de reconocimiento biométricos tienen distintas características que los hacen interesantes para implementarse en múltiples áreas, algunas de ellas son [17]:

- Es intransferible.
- Difícil falsificación.

- Dificil de reproducir, compartir y distribuir.
- Alto grado de certidumbre en la identificación.
- Evita el uso de un nombre o número de usuario.

Para poder utilizar los rasgos biométricos en un sistema electrónico, se requiere disponer del hardware adecuado y realizar un proceso para digitalizar esta información en forma de plantilla, esta se almacena para hacer la verificación cuando así se requiera (ver figura 2.1).

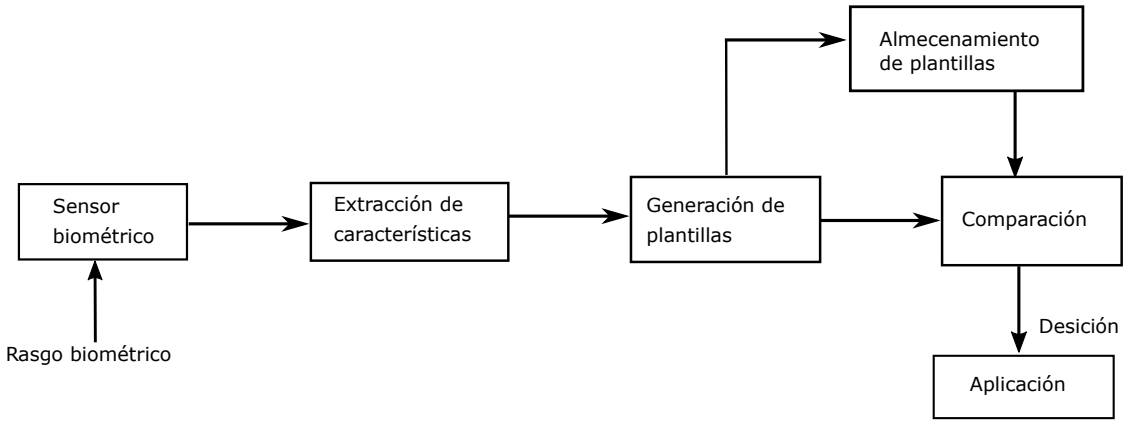


Figura 2.1: Diagrama a bloques de un sistema de *identificación* biométrico [17].

Se pueden usar varios rasgos anatómicos o de comportamiento como identificador, en todos los casos se busca que se cumplan con una serie de requerimientos para elegir el sistema que se debe implementar en la aplicación [11, 12].

- **Universalidad:** Cada persona debe tener ese rasgo biométrico.
- **Particularidad:** Todas las personas tienen que ser suficientemente diferentes en términos del rasgo biométrico.
- **Permanencia:** El rasgo debe ser invariante con el tiempo y a cualquier otro factor desde el punto de vista de la comparación entre rasgos biométricos.
- **Mensurabilidad:** El rasgo biométrico se tiene que poder medir cuantitativamente.
- **Rendimiento:** El rasgo biométrico debe garantizar precisión y robustez en diferentes factores ambientales.
- **Aceptabilidad:** Los usuarios del sistema deben aceptar el uso de ese rasgo biométrico para su identificación.

- **Evitabilidad:** Capacidad de eludir el sistema mediante procedimientos fraudulentos, se puede requerir identificarlo sin la cooperacion de la persona.
- **No falsificable:** El rasgo biométrico tiene que garantizar que su falsificación tenga un alto grado de dificultad.

Existen más características para evaluar los sistemas, no se han colocado todos debido a que algunos pueden ser demasiado específicos, la finalidad de las clasificaciones de estos rasgos es evaluar cuales tienen las propiedades que benefician en mayor grado el sistema de identificación.

Rasgo biométrico	**Característica**			
	Universalidad	**Particularidad**	**Permanencia**	**Mensurabilidad**
Cara	A	B	M	A
Iris	A	A	A	M
Retina	A	A	M	B
Geometría de la mano	M	M	M	A
Huella dactilar	M	A	A	M
Huella de la mano	M	A	A	B
ADN	A	A	A	B
Firma	B	B	B	A
Voz	M	B	B	M

Tabla 2.1: Rendimiento de los sistemas de identificación biométricos, donde B: Bajo; M: Medio; A:Alto. [18].

Rasgo biométrico	**Característica**		
	Rendimiento	**Aceptabilidad**	**No falsificable**
Cara	B	A	A
Iris	A	B	A
Retina	A	B	A
Geometría de la mano	M	M	M
Huella dactilar	A	M	M
Huella de la mano	A	M	M
ADN	A	B	A
Firma	B	A	B
Voz	B	A	B

Tabla 2.2: Continuacion del rendimiento de los sistemas de identificación biométrico, donde B: Bajo; M: Medio; A:Alto. [18].

En la Tabla 2.1 y Tabla 2.2, se observa que la huella dactilar tiene todas las ventajas que se necesitan para un sistema de identificación biométrico y siempre se puede identificar a una persona por este rasgo. Aunque dos huellas se parezcan a simple vista, al extraer las características siempre resultan diferentes. Los sensores actuales captan las huellas a muy alta resolución a un precio asequible y no tienen el problema de tener

que diferenciar el dedo respecto al fondo, como sucede con el reconocimiento del rostro o tomar las muestras a la retina con un mal enfoque del lente.

2.3. Huellas dactilares

La huella dactilar es el patrón de crestas papilares en la yema de los dedos, su formación es siempre única y se determina durante los primeros meses de gestación. El patrón nunca se repite con ninguna otra persona, incluso las huellas de gemelos y las huellas de diferentes dedos de una persona son diferentes [19].

Hoy en día, el reconocimiento de las huellas dactilares es una tecnología fácil de instalar y relativamente de bajo costo. Es una tecnología útil en aplicaciones forenses así como en aplicaciones civiles y de alta seguridad.

La identificación por huellas dactilares no es un descubrimiento reciente. En 1892 se logró identificar a un homicida por el reconocimiento de una huella dactilar ensangrentada en un buzón, fué el caso de Francisca Rojas (Figura 2.2) que es muy conocido en la criminalística [18].

Figura 2.2: Huella del pulgar derecho de *Francisca Rojas*, considerada como la primera persona homicida en haber sido identificada y condenada a partir de sus huellas digitales, junio de 1892.

Después de aquellos acontecimientos, este sistema tomó relevancia en el área forense, de manera que ha sido una de las más utilizadas por los gobiernos, desde el registro para los pasaportes, visas, aeropuertos, cruce fronterizo e incluso en las credenciales de identificación oficial como las tarjetas de circulación y las credenciales para votar.

Se pueden considerar dos tipos de sistemas de reconocimiento, los sistemas de *verificación* y los sistemas de *identificación*.

En el proceso de *identificación*, lo primero que se hace es la extracción de minucias, este proceso se llama *enrolamiento*, en donde se genera el templete digital que se utiliza

para hacer una comparación entre los distintos templetes de otras huellas que se tengan almacenados en una base de datos, el tipo de comparación es denominado *uno a todos*, es decir, se compara la huella actual uno por uno hasta encontrar dos templetes idénticos, el tiempo que se tarda este proceso puede variar según la cantidad de huellas que tenga que comparar antes de encontrar la correcta.

En caso de que el sistema deba dar un resultado negativo, antes debe comparar todas las muestras. En cambio con los sistemas de *verificación*, después de hacer la extracción de minucias (enrolamiento) y obtener el templete de la huella, el usuario debe utilizar un segundo tipo de identificación, por ejemplo un número de usuario o una clave PIN. De esta forma, solo se toma el templete de la base de datos que le corresponde a esa persona y este tipo de comparación es denominado *uno a uno*, es decir, solo hace la comparación entre el templete de la huella actual con el templete que le corresponde a su PIN.

Matriculación: Tanto los sistemas de *verificación* como de *identificación* previamente hacen un registro llamado *matriculación*, este proceso hace la primera captura de la huella por un lector electrónico que extrae las características y las digitaliza. Dependiendo de la aplicación y el grado de seguridad deseado, puede requerir múltiples muestras para minimizar errores y obtener toda la información posible. Es importante que las muestras que se toman para generar el templete o plantilla tengan la mejor calidad posible ya que una vez almacenados en la base de datos, será la identificación de esta persona. Un mal templete pudiera ocasionar un rechazo al sistema hasta no repetir el proceso de matriculación nuevamente [18].

2.4. Extracción de minucias

Existen varias técnicas para el análisis de las huellas dactilares, en la antigüedad se identificaban visualmente en base a las formas del patrón de la figura, pero a medida que se requiere aumentar el número de personas que se deben de identificar, se debe hacer más robusto el sistema de identificación. Actualmente se utilizan los sistemas de minucias para extraer la información de la huella y permite clasificar los templetes por cinco grupos para hacer mas rápida la identificación [19].

Las huellas dactilares son un patrón de crestas papilares que tienen los siguientes elementos:

- Cresta: Es el relieve lineal qu existe en la epidermis de ciertas zonas, que alternando con los surcos, forman el dibujo papilar. Son las rayas negras de una huella impresa en papel.
- Valle: Hendidura entre las crestas de la huella digital.
- Minucia: Una minucia es un punto de interés de la huella digital

Tipos de minucias Las minucias son la información más relevante, hay varios tipos de minucias según la forma del patrón papilar en una sección determinada, en la figura 2.3 se tiene la representación de los tipos de minucias.

- Terminación: Una minucia de terminación es donde la cresta termina y no continua en ningún otro lado.
- Bifurcación: Una minucia de bifurcación es donde una cresta de divide en dos caminos.
- Laguna: Una corta separación que se vuelve a unir inmediatamente.
- Linea independiente: Una linea que aparece, de manera similar a la terminación pero vuelve a continuar.
- Punto: Una minucia de este tipo aparece en medio de dos lineas como si fuera a terminar.
- Spur: Una separación donde una linea continua derecha y la otra se termina rápidamente quedando la forma de un gancho.
- Cruce: Una minucia de cruce consiste en la unión de dos *crestas* en algún punto, en lugar de mantenerse separados por un *valle*.

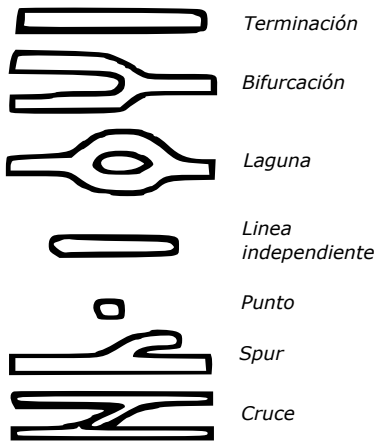


Figura 2.3: Tipos de minucias.

También existen dos atributos especiales que generalmente aparecen una vez en una huella dactilar y sirven para poder clasificar las huellas en grupos (ver figura 2.4).

- **Delta:** Es un área de la huella dactilar donde hay una triangulación o división de las líneas.
- **Core:** Es el centro de la huella dactilar, es donde se genera el inicio del lazo en la huella o el inicio del círculo.

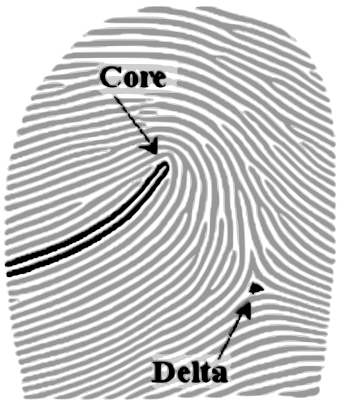


Figura 2.4: Ejemplo de la visualización del *core* y *delta*.

Para hacer una identificación más rápida entre una cantidad relativamente grande de templetes, es conveniente separar por grupos de acuerdo a la forma que tengan, de esta manera se descartan todas las huellas que no sean del grupo antes de iniciar la comparación de todas las minucias de la huella.

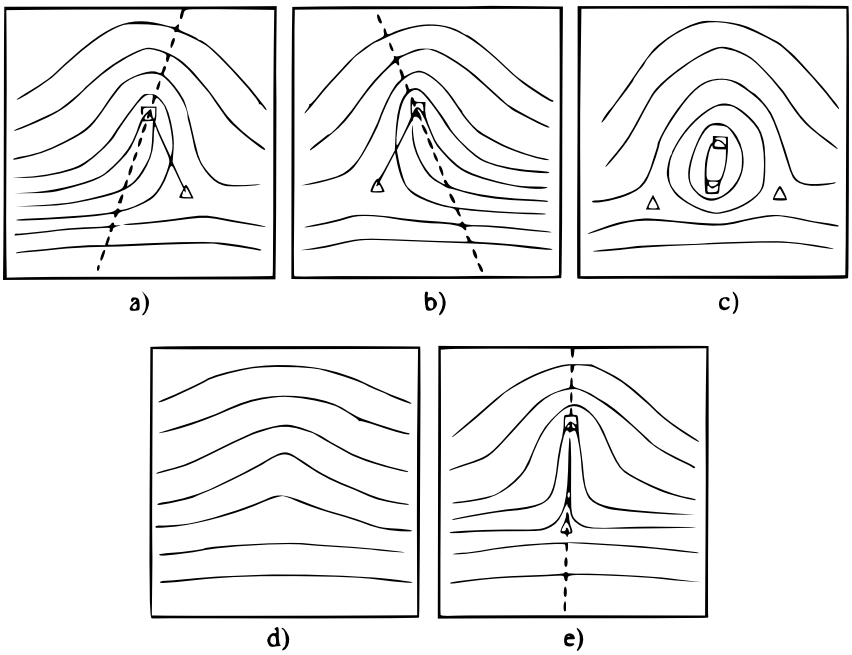


Figura 2.5: Clasificación por grupos de las huellas digitales: a)lazo izquierdo b)lazo derecho, c)espiral d)arco llano e)arco tendido. Los triangulos son las *Deltas* los cuadrados son el *Core*.

Hay varias formas para agrupar las huellas en base a sus patrones. Se pueden considerar cinco grandes grupos de huellas en base al patrón general: lazo izquierdo, lazo derecho, espiral, arco llano y arco tendido. En la figura 2.5 se muestra la representación de estas formas.

Todas las huellas entran en una sola categoría, hay diversas formas que parecieran

no entrar en ninguna de ellas, sin embargo, son variaciones del mismo grupo, son poco frecuentes y es mas fácil considerarlas todas como un grupo grande para no usar un grupo para cada tipo de lazo o cada tipo de espiral.

2.5. Problemas de seguridad

Cada vez son mas las instituciones que están adaptando el uso de la identificación por sistemas biométricos, de manera que toda la seguridad que anteriormente se tenían en distintas credenciales (claves, NIP's, tarjetas magneticas, etc., ahora recaen en la información biométrica. En abril de 2015 se registró un ataque a la oficina de personal para el gobierno de EE.UU (OPM, por sus siglas en ingles). Fueron robados más de 5.6 millones de registros de huellas e información de autorización de seguridad del personal del gobierno de EE.UU. La mayor preocupación acerca de la biometría hasta ahora ha sido la revocación. Es fácil conseguir una nueva contraseña, PIN o tarjeta de crédito después de una violación de seguridad, en cambio, la información biómetrica no se puede cambiar [20].

2.6. Conclusiones

En este capítulo se presentó una introducción a los sistemas de identificación biométrica, se hace una comparación entre diferentes rasgos biométricos y se muestra que en este caso resulta más conveniente el uso de la huella digital. Debido a que se elige utilizar la huella digital como identificador del sistema, se explica la relevancia que ha tenido este sistema de identificación y una explicación a la técnica de detección de minucias, la cual, es usada en los módulos de autentificación por huellas dactilares comercialmente y por el módulo del lector de huella utilizado en este trabajo de tesis.

Capítulo 3

Caos

En este capítulo, se presenta una breve historia del caos y se compara el caos de tiempo continuo con los de tiempo discreto. Se hace la comparación de diferentes mapas caóticos de tiempo discreto con base al exponente de Lyapunov y se elige uno para su implementación en criptografía y sistemas embebidos.

3.1. Introducción

El terminó caos es comúnmente utilizado por su significado lingüístico, puede ser asociado directamente con el desorden, confusión, desorganización, etc. En el área de las matemáticas y física, se adoptó este término para referirse a un comportamiento aparentemente impredecible de los sistemas dinámicos no lineales determinísticos.

La idea del determinismo fue planteado por el filósofo franco-alemán Paul Thiry d'Holbach, contrario a lo que se pensaba a finales del siglo XVIII, Holbach aseguraba que todo lo que ocurría tenía una causa.

> «*En un torbellino de polvo levantado por un viento impetuoso, por confuso que parezca a nuestros ojos, en medio de la tempestad más horrible avivada por vientos opuestos que empujan las corrientes, no hay ni una sola molécula de polvo o de agua que esté colocada al azar, que no tenga una razón suficiente para ocupar el lugar donde se encuentra, ni actúe en rigor como debe hacerlo. Un geómetra que conociera con exactitud las diferentes fuerzas que actúan en ambos casos y las propiedades de las moléculas en movimiento, demostraría que, por motivos fundados, cada molécula actúa precisamente como debe actuar y no podría hacerlo de otra manera.*»

Después en su ensayo filosófico sobre las probabilidades, astronomía y matemáticas de Laplace en 1814.

> «*Podemos mirar el estado presente del universo como el efecto del pasado y la causa de su futuro. Se podría concebir un intelecto que en cualquier momento dado conociera todas las fuerzas que animan la naturaleza y las posiciones de los seres que la componen; si este intelecto fuera lo suficientemente vasto como para someter los datos a análisis, podría condensar en una simple fórmula el movimiento de los grandes cuerpos del universo y del átomo más ligero; para tal intelecto nada podría ser incierto y el futuro, así como el pasado, estarían frente a sus ojos.*»

En términos generales, el caos determinista da lugar a trayectorias asociadas a la evolución temporal de forma muy irregular y aparentemente aleatoria, sin embargo, son totalmente deterministas. La irregularidad de las trayectorias está asociada a la imposibilidad práctica de predecir la evolución futura del sistema, aunque esta evolución sea totalmente determinista [21].

El origen de la teoría del caos sucede por *el problema de los tres cuerpos*, tras la formulación de la ley de gravitación universal por Isaac Newton, que explicaba las fuerzas de atracción entre dos objetos, se estudió el movimiento de tres cuerpos en interacción gravitatoria, ya que el sistema sol, tierra y luna, supestamente estaba aislada del resto del universo. El propósito de aquella investigación era determinar si el sistema solar era *estable* a largo plazo, o si uno de los cuerpos pudiera un día golpear a otro cuerpo, o ser expulsado del sistema solar hasta el infinito. La idea del problema era poder saber, en cualquier instante de tiempo, las posiciones y velocidades de tres cuerpos, de cualquier masa, sometidos a atracción gravitacional mutua y partiendo de posiciones y velocidades dadas [22].

En 1963, Edward Lorenz (1917-2008), quien se interesaba en el problema de la convección en la atmósfera terrestre, simplificó de forma drástica las ecuaciones de Navier-Stokes de la mecánica de fluidos, conocidas por su complejidad. El modelo atmosférico de Lorenz es a lo que los físicos llaman *modelo de juguete*, al reducir el número de dimensiones y variables para formar un sistema de ecuaciones análogo, ilustrar un efecto y hacer el fenómeno más fácil de visualizar [21].

Lorenz utilizaba una de las primeras computadoras del la historia, la Royal McBee LGP-30. Su método consistía en ingresar en la computadora una serie de parámetros de seis cifras decimales para iniciar su algoritmo. El protocolo suponía que se debía hacer dos veces para cada conjunto de parámetros, con fines de verificación. Se dice que un día Lorenz, mientras realizaba sus experimentos decidió hacer una taza de café, así que antes de ir a prepararla ingresó los parámetros con 3 cifras decimales para acelerar el proceso del cálculo, de manera que al volver ya tuviera los resultados. Lorenz esperaba resultados similares a la primera columna de 6 decimales sin embargo eran totalmente diferentes, repitió el experimento con diferentes grados de precisión y siempre eran diferentes [23].

Hoy en día se conoce como *atractor de Lorenz* y debido a la forma se le llamó el *efecto mariposa* (ver figura 3.1).

Así Lorenz descubrió el principio fundamental de la teoría del caos, es decir, una variación de los parámetros muy pequeña en las condiciones iniciales puede variar enormemente el resultado final.

El sistema de ecuaciones de Lorenz se describe como sigue:

$$\frac{dx_1}{dt} = \alpha(y - x), \tag{3.1}$$

$$\frac{dx_2}{dt} = rx - y - xz, \tag{3.2}$$

$$\frac{dx_2}{dt} = xy - bz, \tag{3.3}$$

donde x_1, x_2, x_3 son los estados del sistema y α, r, y b los parámetros de control.

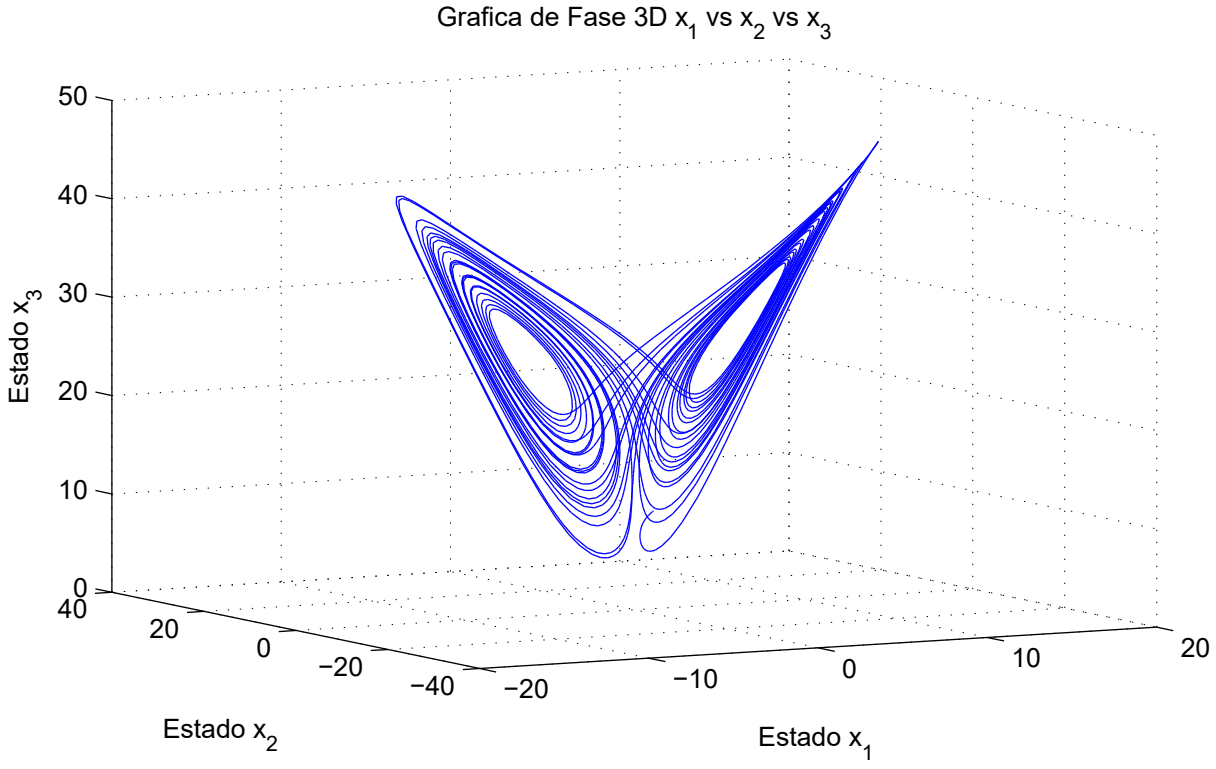


Figura 3.1: Atractor de Lorenz.

3.2. Dinámica caótica

En matemáticas y física, los sistemas dinámicos se pueden clasificar en estables, inestables o caóticos. Un sistema estable es aquel que genera fuerzas de atracción y lo

mantiene confinado en un atractor de punto fijo u órbita periódica, mientras que, un sistema inestable generan fuerzas de repulsión que expulsa a la trayectoria fuera del atractor. Un sistema caótico presenta ambas características, parte de la ecuación hace que la trayectoria se desvíe y otra parte de la ecuación atrae la trayectoria del sistema, como resultado se obtiene una trayectoria impredecible. [15, 21, 22]

Los sistemas lineales son fácilmente predecibles, se puede estimar los próximos valores conociendo muy poca información. Sin embargo algunos sistemas no lineales como las funciones trigonométricas, también se pueden estimar ya que ambos son sistemas deterministas, solo se necesita un poco mas de información como la amplitud y el periodo (figura 3.2). Un sistema se dice que es determinista, si su evolución con respecto al tiempo se puede determinar completamente a partir de las condiciones iniciales y parámetros que describen dicho sistema.

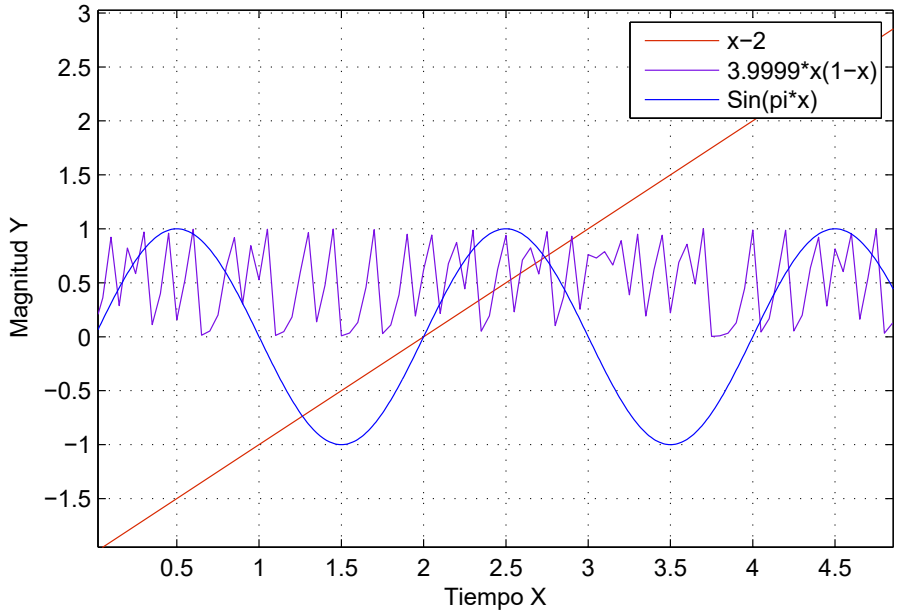


Figura 3.2: Comportamiento de tres funciones matemáticas, la gráfica en morado es un *mapa logístico* en dinámica caótica.

Podemos destacar las siguientes propiedades de un sistema caótico:

- Es sensible a las condiciones iniciales.
- Su serie de tiempo es acotada en amplitud.
- Las iteraciones de tiempo son infinitas.
- El sistema no converge a un punto.
- El sistema es no lineal.

- Proviene de un modelo determínistico.
- La serie es aperiódica.
- Tiene al menos un exponente de Lyapunov positivo.

Para aplicaciones de encriptación, resulta interesante utilizar las propiedades que presentan este tipo de dinámicas tan complejas para el proceso de cifrado. De la lista anterior se puede notar que algunas de esas características pueden ser visualizadas a simple vista si prestamos atención a los detalles de la gráfica temporal, en cambio el *exponente de Lyapunov* y la sensibilidad a *condiciones iniciales* requieren un análisis especial.

3.3. Exponente de Lyapunov

El exponente de Lyapunov es un análisis estadístico sobre la manera en que dos secuencias con condiciones iniciales extremadamente similares se comportan de forma iterativa, es una forma de poder cuantificar cuanto divergen. De esta forma, podemos usar el exponente de Lyapunov como un indicador del grado de sensibilidad a las condiciones iniciales y predictividad de la secuencia caótica [24].

La sensibilidad a las condiciones iniciales puede ser cuantificado con [25]

$$\|\delta x(t)\| \approx e^{\lambda t}\|\delta x_0\| \tag{3.4}$$

que para sistemas de tiempo discreto se usa

$$\|\delta x(n)\| \approx e^{\lambda n}\|\delta x_0\| \tag{3.5}$$

El exponente de Lyapunov se calcula por lo tanto con la siguiente expresión

$$\lambda = \frac{1}{T} ln \| \frac{f^n(x_n - \delta_0) - f^n(x_n)}{\delta_0} \| \tag{3.6}$$

donde λ es el exponente de Lyapunov, T el número de iteraciones, x_n el estado actual del mapa y $x_n - \delta$ es la secuencia con una perturbación infinitesimal.

Este análisis es aplicado a cada mapa caótico para elegir uno que sea caótico y sea factible para ser utilizado en una aplicación de tiempo real en un microcontrolador. Si el mapa caótico es de n-dimensión, se tendrá n exponentes de Lyapunov.

3.4. Mapas caóticos

Para todos los mapas caóticos se calcula el exponente de Lyapunov con tres mil iteraciones, es un estimado de la cantidad de iteraciones que se usará en el microcontrolador para el cifrado caótico.

Mapa logístico

El sistema no lineal más simple que puede tener comportamiento caótico y también uno de los más estudiados es el modelo demográfico conocido como *mapa logístico* [26].

$$x_{n+1} = \alpha x_n(1 - x_n), \tag{3.7}$$

donde $x_n \in (0, 1)$ y α es el parámetro de control, el cual debe ser $3.57 < \alpha < 4$ para obtener la dinámica caótica (ver figura 3.3).

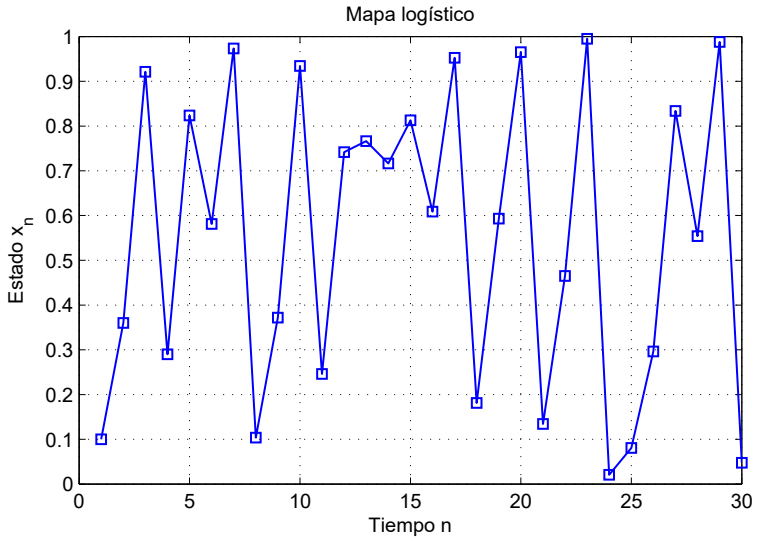


Figura 3.3: Estados del mapa logístico con dinámica caótica.

Se usan los parámetros $\alpha = 3.9999$ y $x_0 = 0.1$. El exponente de Lyapunov en matlab arroja un resultado de $\lambda = 0.6772$, podemos afirmar que para estos parámetros se genera siempre una secuencia caótica.

Mapa Hénon

El mapa de Hénon es un sistema dinámico discreto en el tiempo (figura 3.4 y figura 3.5), fue introducido por Michel Hénon como modelo simplificado de la sección de Poincaré del modelo de *Lorenz*. El mapa de Hénon toma un punto (x_n, y_n) en el plano y lo mapea a un punto nuevo [27].

$$x_{n+1} = 1 - \alpha x_n^2 + y_n, \tag{3.8}$$

$$y_{n+1} = \beta x_n, \tag{3.9}$$

donde $x_n \in (-1.29, 1.28)$ de acuerdo a los resultados en Matlab, con α y β como parámetros de control, los cuales deben ser reales. $1.34 < \alpha < 1.426$ para obtener la dinámica caótica cuando $\beta = 0.3$.

Para calcular el exponente de Lyapunov se usan los parámetros $\alpha = 1.399$, $\beta = 0.3$ y $(x_0, y_0) = (0.1, 0.1)$. Al tener dos estados se tienen dos resultados, $\lambda_1 = 0.119674$ y

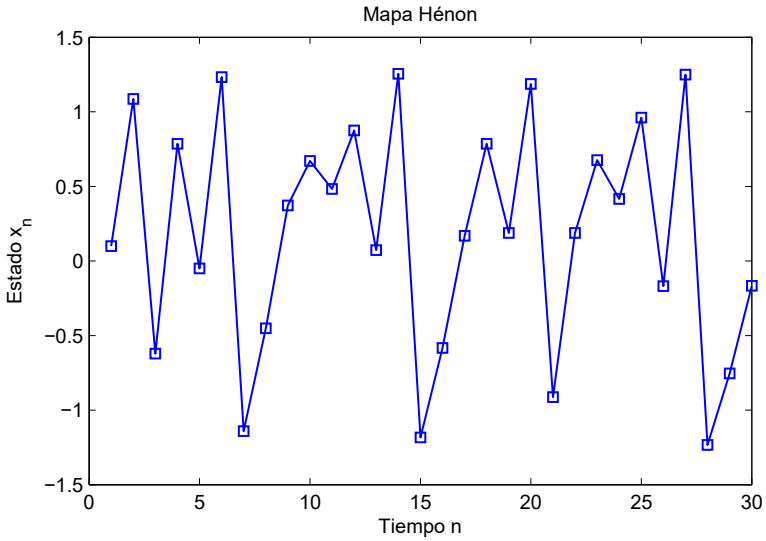


Figura 3.4: Estados x del mapa Hénon con dinámica caótica.

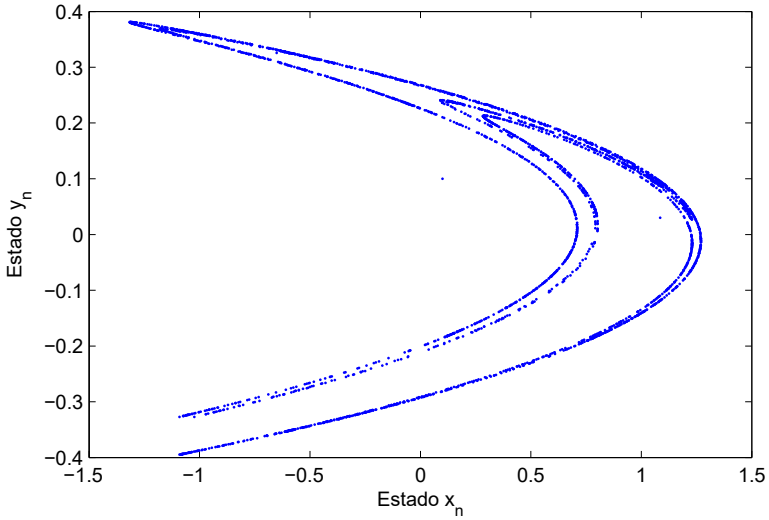


Figura 3.5: Atractor caótico del mapa Hénon.

λ_2 = -1.203972, lo cual indica que las secuencias tienen dinámica caótica.

Mapa de Bernoulli generalizado

Este mapa tiene la siguiente ecuación [28, 29]

$$x_{n+1} = \frac{x_n}{\alpha}, mod1, \tag{3.10}$$

donde $x_n \in (0, 1)$, mod es la operación módulo y α es el parámetro de control, el cual debe ser $0 < \alpha < 1$ para obtener la dinámica caótica (figura 3.6).

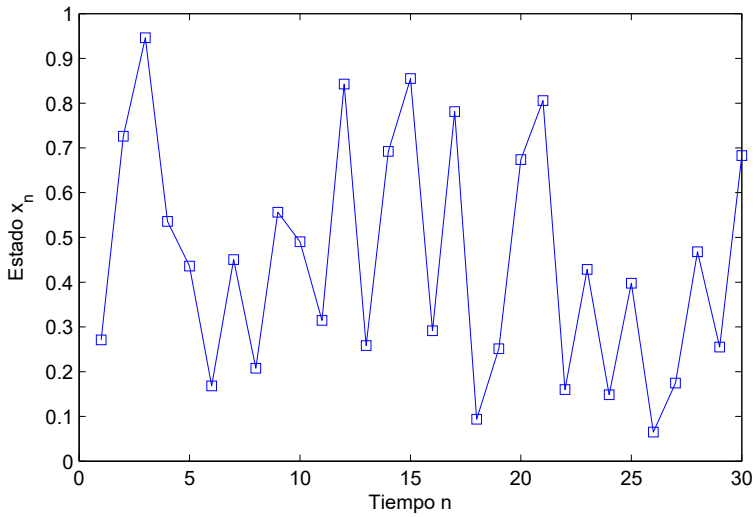


Figura 3.6: Estados x del mapa Bernoulli con dinámica caótica.

Se usan los parámetros $\alpha = 0.3731$ y condición inicial $x_0 = 0.2709$, lo que da como resultado un exponente de Lyapunov de $\lambda = 0.99859$.

Mapa Chebyshev

Los polinomios de Chebyschev son curvas de coseno con una perturvación en la escala horizontal, pero la escala vertical se mantiene constante [30].

$$x_{n+1} = cos(\alpha cos^{-1}(x_n)), \tag{3.11}$$

donde $x_n \in (-1, 1)$ y α es el parámetro de control, el cual debe ser $1 < \alpha$ para obtener dinámica caótica (figura 3.7).

Para calcular el exponente de Lyapunov se usa $\alpha = 3.7$ y $x_0 = 0.6$. Como resultado se obtiene $\lambda_1 = 1.3084$, lo que indica que la dinámica es caótica.

Mapa logistico 2D

Este mapa usa dos ecuaciones del mapa logístico, esta modificación agrega un polinomio de segundo orden con el que ambas ecuaciones interaccionan entre si [31].

$$x_{n+1} = \alpha_1 x_n(1 - x_n) + \beta_1(y_n^2), \tag{3.12}$$

$$y_{n+1} = \alpha_2 y_n(1 - y_n) + \beta_2((x_n^2) + x_n y_n), \tag{3.13}$$

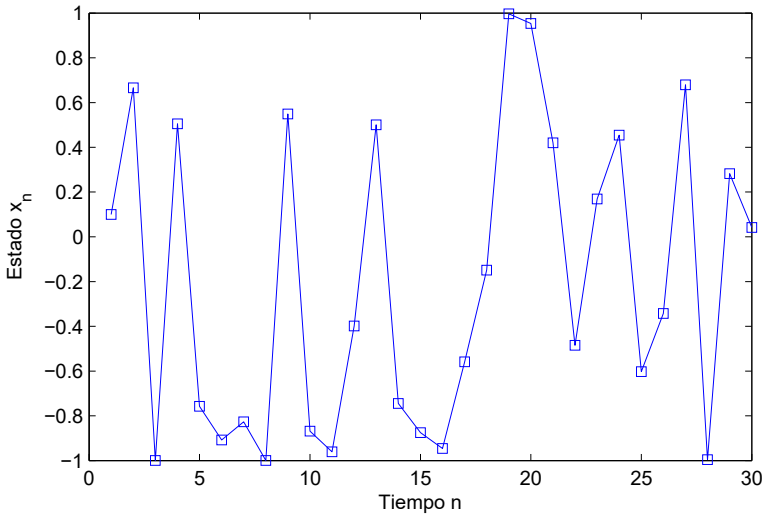


Figura 3.7: Estados x del mapa Chebyshev con dinámica caótica.

donde $x_n \in (0.26, 0.90)$, $y_n \in (0.19, 0.96)$ y α_1, α_2 β_1 y β_2 son los parámetros de control, los cuales deben ser $2.75 < \alpha_1 < 3.4$, $2.75 < \alpha_2 < 3.45$, $0.15 < \beta_1 < 0.21$, $0.13 < \beta_2 < 0.15$ para obtener la dinámica caótica (ver figura 3.8).

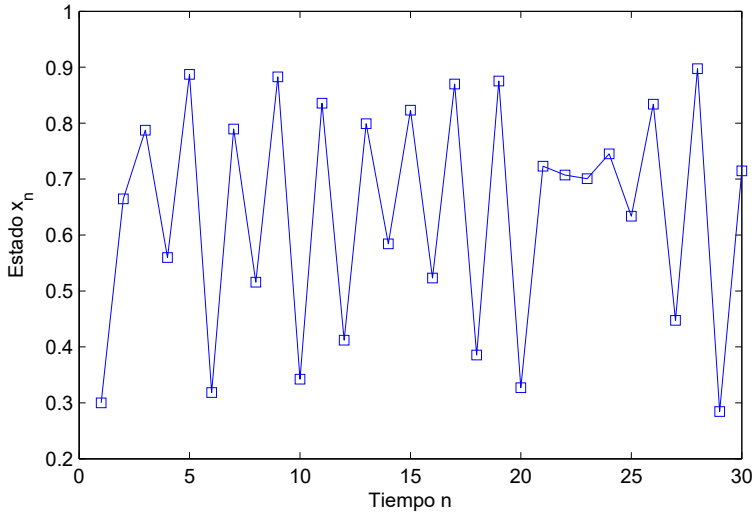


Figura 3.8: Estados x del mapa logístico 2D con dinámica caótica.

Se usan los siguientes parámetros y condiciones iniciales para calcular el exponente de Lyapunov: $\alpha_1 = 2.93$, $\beta_1 = 0.197$, $\alpha_2 = 3.17$, $\beta_2 = 0.139$, $x_0 = 0.3$ y $y_0 = 0.5$. Dando como resultado $\lambda_1 = -0.005439$ y $\lambda_2 = 0.19145$.

Mapa Lotka-Volterra

Este mapa es un modelo demográfico presa-depredador, ha sido reportado como hyper-

caótico en [32], con las siguientes ecuaciones.

$$x_{n+1} = \alpha x_n(1 - x_n) - \beta x_n y_n, \tag{3.14}$$
$$y_{n+1} = \delta x_n y_n, \tag{3.15}$$

donde $x_n \in (0, 0.87)$, $y_n \in (0, 14.5)$ de acuerdo a los resultados en Matlab, α, β y δ son los parámetros de control, los cuales deben ser $\alpha = 3.5$, $\beta > 1.4$ y $3.7 < \delta < 4$ para obtener la dinámica hypercaótica (figura 3.9 y figura 3.10).

Para el calculo de los exponentes de Lyapunov se usan $\alpha = 3.5$, $\beta = 3.2$, $\delta = 3.8$, $x_0 = 0.1$, $y_0 = 0.1$. Dando como resultados $\lambda_1 = 0.1589$ y $\lambda_2 = 0.07558$. Lo cual indica hypercaos.

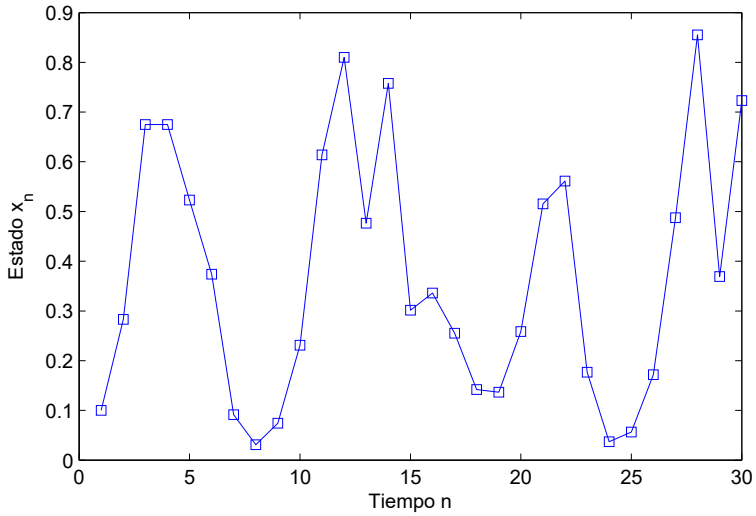


Figura 3.9: Estados de x del mapa Lotka-Volterra con dinámica hypercaótica.

3.5. Aplicaciones del caos

El caos es una caracteristica que se puede presentar en los sistemas dinámicos no lineales y debido a su comportamiento, se tienen aplicaciones en comunicaciones seguras [33]; generadores de números pseudoaleatorios [34], en el area de finanzas, se ha buscado entender los comportamientos de los valores del mercado [35], en la robótica se ha utilizado para programar trayectorias de drones con aplicaciones de búsqueda y secate [36] y como en esta tesis de licenciatura, tambien se han hecho otras investigaciones para el encriptado de información biométrica [17].

3.6. Conclusiones

En este capítulo, se presentó una introducción a los sistemas caóticos y sus propiedades. Se implementan seis mapas caóticos de una y dos dimensiones y se determinó

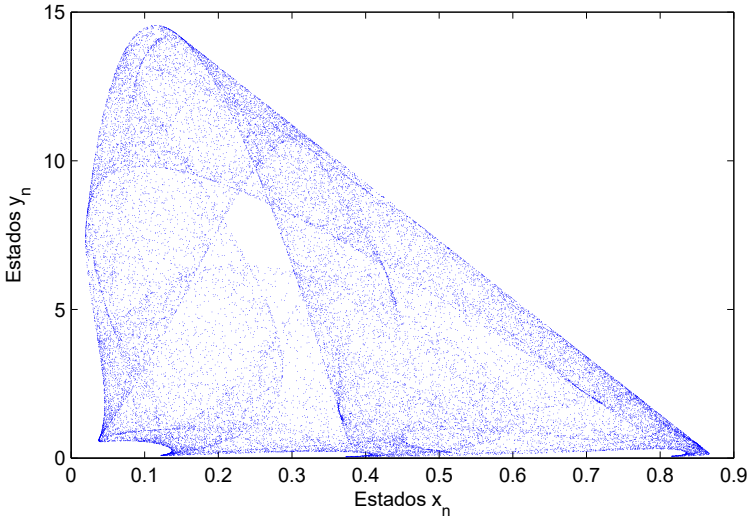


Figura 3.10: Atractor hypercaótico de Lotka-Volterra.

su exponente de Lyapunov, el cual, nos indica caos (ver Tabla 3.1). Para la aplicación criptográfica en esta tesis, se utiliza el mapa de 2-dimensiones Lotka-Volterra por presentar dinámica hipercaótica y pocas operaciones aritméticas. Además, la literatura no ha presentado algoritmos criptográficos basadas en dinámicas hipercaóticas del mapa Lotka-Volterra.

Mapa caótico	**Exponente de Lyapunov**	
	$\boldsymbol{\lambda_1}$	$\boldsymbol{\lambda_2}$
Logístico	0.6772	x
Hénon	0.11967	-1.20397
Bernoulli generalizado	0.9985	x
Chebyshev	1.3084	x
logistico 2D	-0.005439	0.19145
Lotka-Volterra	0.29742	0.82299

Tabla 3.1: Exponente de Lyapunov para los mapas estudiados.

Capítulo 4

Criptografía

En este capítulo, se presenta una breve historia de la criptografía, se presenta el funcionamiento de los sistemas de encriptado y se describen algunas pruebas de seguridad.

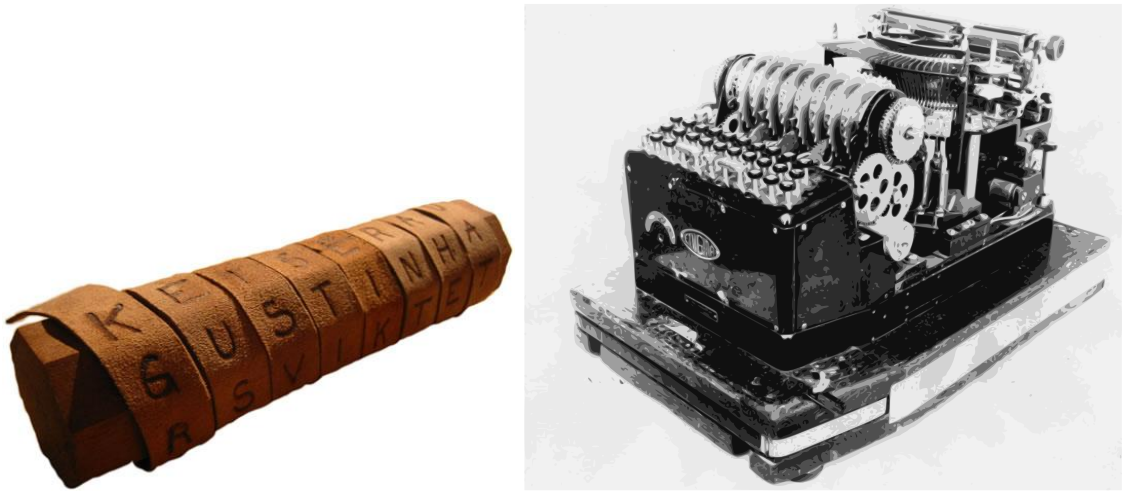
4.1. Introducción

La criptografía clásica data desde el año 1500 a.C. y fueron creados distintos métodos y artilugios para esconder información. El hombre ha tenido, por diversas razones, la necesidad de transmitir la información de manera confidencial, ya sea por motivos militares, comerciales, diplomáticos o personales.

Uno de los primeros artilugios para transmitir un mensaje fue la *escítala*, utilizada por los griegos en el siglo V a.C. que consistía en una tira de cuero, tela o papiro con el grabado del mensaje, se enrollaba en un cilindro de madera y que utilizaba la técnica de cambio de lugar. En el siglo I a.C. aparece un nuevo procedimiento de cifrado, el cual es conocido como *cifrado de César*, debido a que era usado por el militar y político romano Julio César. Este método consistía en sustituir cada carácter del mensaje original por otro situado tres posiciones después de el en un determinado alfabeto. En 1790, *Thomas Jefferson* creó un cilindro formado por varios discos coaxiales en donde cada uno tenía escrito en la parte exterior un alfabeto [37].

La criptografía moderna involucra máquinas mecánicas y electromecánicas. En 1923, Arthur Sherbius un ingeniero alemán dio a conocer una máquina llamada Enigma, se trataba de una máquina que en su exterior parecía una máquina de escribir común, pero en su interior estaba compuesta por un mecanismo que trasformaba la letra tecleada en otra. *Máquina Purple* en 1940 construida por los japoneses similar a la Enigma. *Máquina Sigaba* en 1940 desarrollada por los estadounidenses, similar a la Enigma;

Conforme pasaba el tiempo, la criptografía tomaba un carácter matemático y gracias a los estudios realizados por Claude Shannon sobre la Teoría de la Información (1948) y posteriormente con la publicación de su trabajo “La Teoría de las Comunicaciones



a) b)

Figura 4.1: Artilugios criptográficos históricos: a) *Escítala*, utilizada por los griegos en el siglo V a.C. b) *Enigma*, máquina criptográfica utilizada por los alemanes durante la II Guerra Mundial.

Secretas" en 1949, en donde sugería utilizar operaciones múltiples que mezclaran transposiciones y sustituciones, la criptografía dejó de considerarse un arte y fue reconocida como una ciencia [37].

A partir de entonces, la invención de la computadora y el desarrollo de las matemáticas, permitieron que los nuevos algoritmos de cifrado fueran cada día más complejos, como el *DES* (del inglés, *Data Encryption Algorithm*), importante por ser el primer algoritmo de cifrado estándar en los años 70's y con ello el nacimiento de la criptografía moderna, *AES* (del inglés, *Advanced Encryption Standard*), *RSA* (del inglés, *Rivest, Shamir y Adleman*), *IDEA* (del inglés, *International Data Encryption Algorithm*), entre otros [37].

La criptografía es la ciencia que estudia las técnicas y métodos para transformar la información de manera que sea ilegible a personas no autorizadas, a esta cualidad se le llama *confidencialidad*. Por lo contrario el criptoanálisis es la ciencia que se ocupa de analizar el algoritmo y texto cifrado, para encontrar el texto claro o la clave secreta.

4.2. Sistemas de encriptado

Un *sistema criptográfico* consiste en un algoritmo usualmente de dominio público, que depende de un parámetro llamado clave secreta para cifrar (codificar) y descifrar (decodificar) información entre emisor y receptor [15]:

- m: mensaje.
- c: criptograma.

- K: clave secreta.
- E: representa la función de cifrado.
- D: representa la función de descifrado.

En todo sistema criptográfico, se debe cumplir la siguiente condición

$$D_K(E_K(m)) = m \tag{4.1}$$

Esto significa que si un mensaje m se cifra con una función E y una clave K y después se descifra con la misma clave K, se obtiene el mensaje original m (figura 4.2).

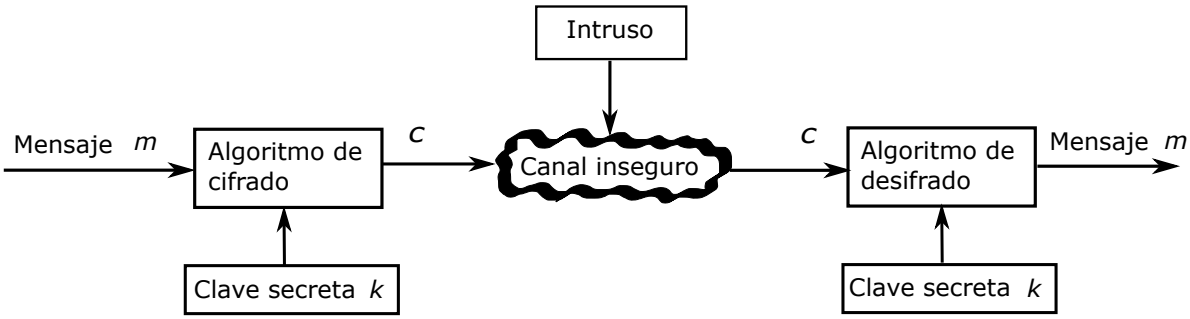


Figura 4.2: Esquema de cifrado con clave simétrica.

Se consideran cinco formas de clasificar los sistemas criptográficos [15]:

1. **Distribución de la clave secreta.**
   - *Simétricos o de clave privada.*
   - *Asimétricos o de clave pública.*
   - *Criptografía híbrida.*
2. **Operaciones realizadas en el algoritmo:**
   - *Confusión:* Consiste en permutar cada elemento del mensaje de manera desordenada en función de la clave secreta.
   - *Difusión:* Consiste en cambiar el valor a cada elemento del texto claro de manera desordenada en función a la clave secreta, para transformarlo en otro elemento del mismo tipo.
3. **Estructura del algoritmo de cifrado:**
   - *Cifrado de flujo:* Cifran el mensaje bit a bit (o byte a byte) hasta terminarlo.
   - *Cifrado de bloque:* Cifran el mensaje en bloques de n bits simultáneamente hasta terminarlo. Algunos ejemplos son el DES, 3DES, RC5, AES.

4. **Método para construir el algoritmo criptográfico:**
    - *Convencionales:* Utiliza la criptografía tradicional para diseñar los algoritmos, como la teoría de números, álgebra, curvas elípticas, números primos, operación módulo, etc. por ejemplo, 3DES, AES e IDEA.
    - *No convencionales:* En este caso, se utilizan herramientas matemáticas en estado de investigación como la criptografía cuántica, ADN y caótica.

4.3. Ataques criptográficos

El principio de Keckhoff indica que la seguridad de un sistema criptográfico debe recaer en la clave secreta y no sobre el algoritmo de cifrado. El algoritmo de cifrado se considera de dominio público y la clave debe ser confidencial.

Un sistema criptográfico se considera vulnerado si un criptoanalista encuentra la forma de determinar la clave secreta y en consecuencia el texto claro. Básicamente, existen tres formas de vulnerar un sistema criptográfico [15]:

1. **Ataques teóricos (lógicos):** Aplicar teoría de la información y criptoanálisis para quebrantar el algoritmo.
2. **Ataques físicos:** La información de tiempo, el consumo de energía, fugas electromagnéticas o incluso sonido, pueden proporcionar fuentes adicionales de información, que puede aprovecharse para romper el sistema criptográfico.
3. **Ataques humanos:** Obtener información importante para desencriptar como, numeros de usuarios, nombre de usuario o tarjetas de acceso, por medio del robo, soborno, extorsión u otro método.

En esta tésis se propone un algoritmo caótico para ser implementado en un microcontrolador, para comprobar que sea viable, se evalúa mediante análisis teóricos para resistir distintos métodos de ataques criptoanalíticos.

4.4. Cifrado caótico

El cifrado de información consiste en transformar un mensaje claro en un mensaje ilegible mediante un algoritmo y una clave secreta, de tal manera que sólo la persona autorizada pueda decodificar y entender.

En [38] fueron presentadas unas series de reglas que se deben considerar para formar un sistema criptográfico:

- Se debe describir qué sistema caótico utiliza el cifrado.

- La degradación digital sebe ser evaluada.
- El sistema criptográfico debe ser fácil de implementar.
- La clave secreta debe ser claramente definida.
- El espacio de claves debe ser especificada sólo para generar secuencias caóticas.
- El efecto avalancha debe producirse para cualquier clave secreta.
- Información parcial de la clave secreta, no debe revelar información parcial del mensaje, tampoco de la parte de la clave desconocida.
- El proceso para generar secuencias caóticas a partir de la clave secreta debe estar claramente definido.
- El cifrado debe tener alta sensibilidad al texto claro.
- El cifrado debe generar un texto cifrado con distribución de probabilidad uniforme.

Recientemente se ha propuesto criptografía caótia para imagenes [8, 17], biometría [39], texto alfanumérico [40] y telemedicina [41]. Para la aplicación en telemedicina, se encriptan las señales obtenidas de los electrocardiogramas y electroencefalogramas para mantener la privacidad y seguridad en el intercambio de información entre paciente y doctor (o su almacenamiento en las bases de datos); se han presentado algoritmos de encriptado basados en caos para el encriptado de imágenes y texto alfanumérico, utilizado secuencias pseudoaleatorias para el proceso de confusión y difusión, donde muestran tener excelentes resultados estadísticos; debido al incremento que ha tenido el uso de sistemas de verificación biométricos, ha crecido también la necesidad de proteger esta información, para proteger la integridad de todo el sistema y de sus usuarios, para solucionar estos problemas se han utilizado algoritmos de encriptado basados en caos.

4.5. Conclusiones

A lo largo de la historia se han empleado diferentes métodos para mantener la información importante segura y confidencial, en este trabajo de tesis de licenciatura se propone un sistema criptográfico no convencional, utilizando caos para el *cifrado de flujo*, el encriptado caótico tiene un grado de aleatoriedad bastante alto al cumplir las pruebas estadísticas de aleatoriedad, por estas razones resulta conveniente implementar este tipo de encriptado en los sistemas embebidos que guarden información confidencial.

Capítulo 5

Sistema de acceso seguro propuesto

En este capítulo se presenta la implementación y la estructura del sistema de acceso seguro propuesto. Se presenta un sistema de autenticación basado en la huella dactilar y su cifrado caótico para incrementar la seguridad del sistema embebido y proteger la identidad del usuario. El sistema de cifrado se basa en el algoritmo de encriptado presentado por [15]; se adapta el mapa hypercaótico Lotka-Volterra reportado en [32] para que sea utilizado en este algoritmo de encriptado en lugar del mapa logístico; se presenta un análisis de seguridad como el espacio de la clave, sensibilidad a la clave, sensibilidad a la plantilla biométrica, correlación, entropía, análisis de aleatoriedad y tiempos de procesamiento.

5.1. Introducción

Se ha demostrado que todas las personas pueden ser identificadas, a través de las características fisiológicas o conductuales que hacen a las personas únicas. Estas características se conocen como identificadores biométricos. En el Capítulo 2, se da una explicación de este tema. Comparado con técnicas de identificación tradicional como tarjetas de identificación (ID), claves, número de identificación personal (NIP), tarjeta inteligente, etc, todas ellas son susceptibles al robo, difíciles de recordar, fácil de determinar, o copias no autorizadas de los identificadores, que a diferencia de los identificadores biométricos requieren una verificación presencial para autenticar el acceso al sistema y evita muchas desventajas de identificación tradicional. Actualmente, la huella dactilar es la característica biométrica más utilizada para identificar personas ya que es práctica, muy aceptada, es la más estudiada y la más desarrollada.

5.2. Sistema embebido

Un sistema embebido es un sistema basado en un microprocesador o un microcontrolador que está desarrollado para controlar una aplicación o un rango de aplicaciones, tienen una arquitectura completa de computadora para ejecutar todas sus funciones de

forma independiente: CPU, memoria interna del sistema (RAM, ROM y flash), puertos de entrada y salida (GPIO) y módulos de comunicación.

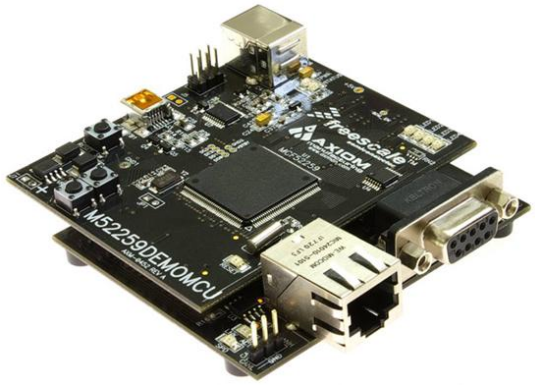


Figura 5.1: Tarjeta de desarrollo de 32-bits M52259DEMOKIT utilizado en el trabajo experimental de esta tesis.

La implementación del sistema propuesto está basada en el microcontrolador ColdFire MCF22259 de Freescale (figura 5.1), con una frecuencia de reloj de 80 MHz, esta tarjeta de desarrollo es capaz de ejecutar todas sus funciones de forma independiente: CPU, memoria flash, memoria RAM, memoria ROM, puertos de entrada y salida (GPIO) y módulos de comunicación como USB, ethernet, I2C, Serial. Estas y otras características hacen del microcontrolador MCF5225X ideal para aplicaciones en control industrial, redes de internet industriales, sector salud, seguridad, donde se requiere un amplio rango de conectividad y alto desempeño [42].

Para capturar las huellas dactilares se utiliza el módulo Futronic modelo FS83 con comunicación RS-232. Para desplegar los menús de la interfaz de usuario se utiliza una pantalla LCD 20X4 matrix orbital modelo LK204-25 [43]. Para guardar los datos se utiliza la memoria flash interna del microcontrolador con capacidad de 512KB. Para las salidas digitales del control de acceso se utilizan los puertos GPIO de la panantalla LCD. El algoritmo de cifrado se programa con lenguaje C en el entorno de desarrollo integrado (IDE) CodeWarrior de Freescale 7.1 y MQX 3.5.1.

En la figura 5.2, se muestra el diagrama a bloques del funcionamiento del sistema embebido propuesto para la parte experimental de este trabajo, se indica la forma en que los cinco módulos se conectan y comunican. En la figura 5.3, se tiene una fotografía del sistema funcionando y se ubican los módulos del sistema.

El sistema embebido está formado por cinco módulos independientes, cada una con funciones especificas y controladas directa o indirectamente por la tarjeta de desarrollo MCF5229DEMOKIT:

1. Microcontrolador: El programa del sistema se ejecuta en este módulo, al iniciar, se establece la comunicación I2C, RS-232, configura los puerto de entrada y salida.

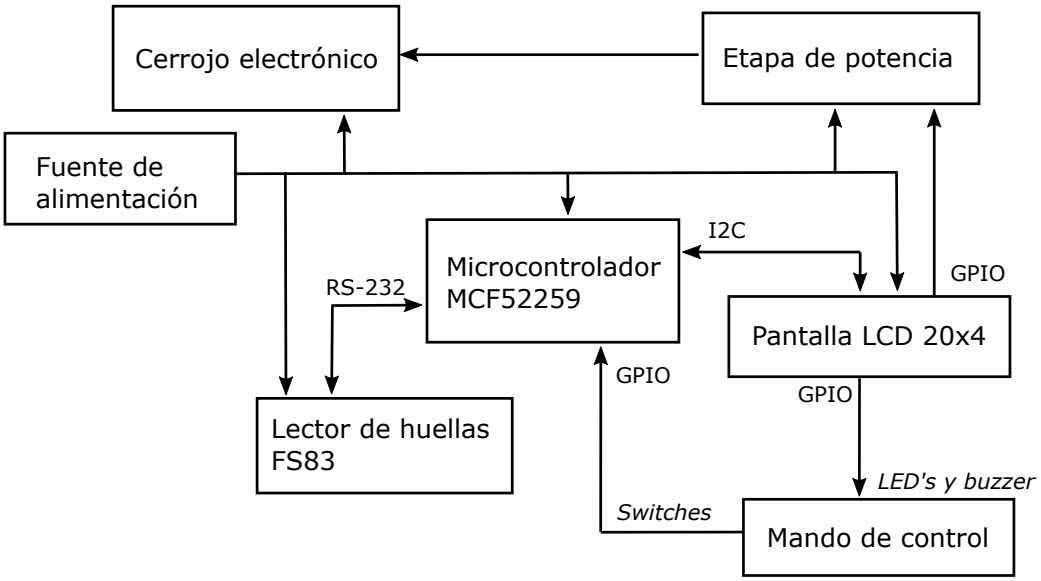


Figura 5.2: Esquema del sistema embebido de autentificación por huella dactilar seguro y propuesto en esta tesis.

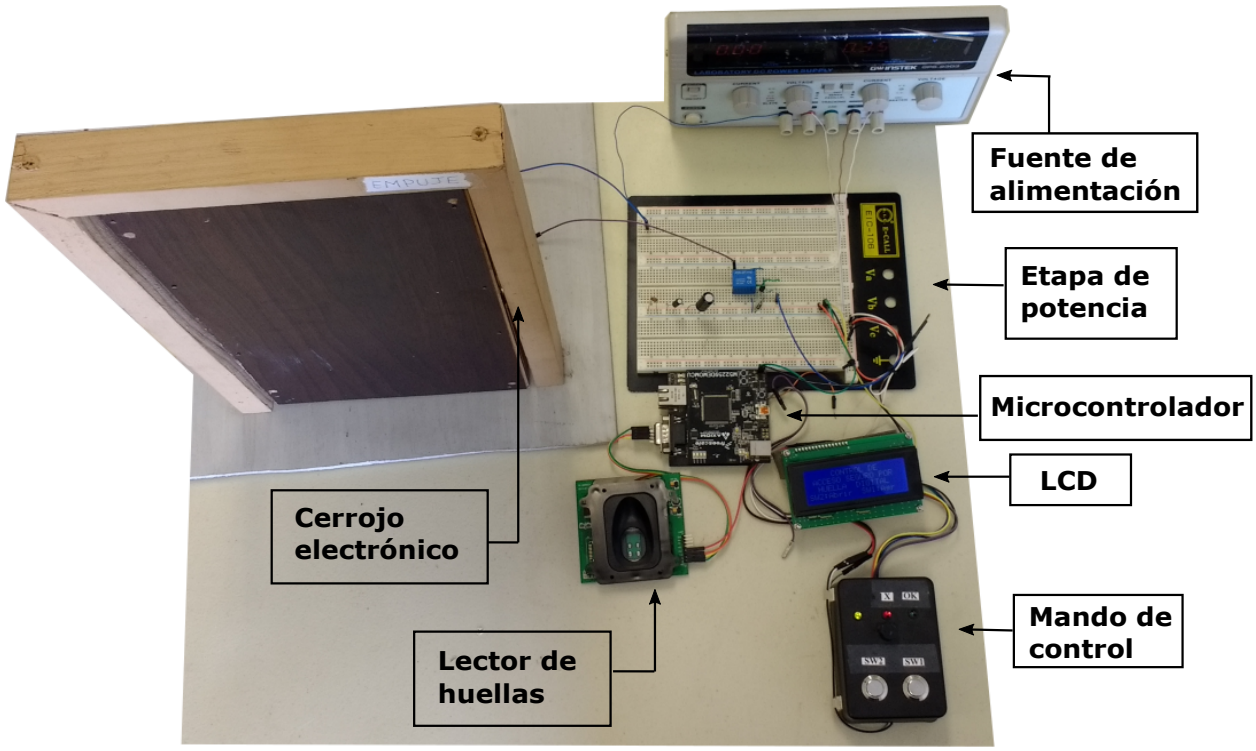


Figura 5.3: Foto del sistema de acceso seguro con encriptación caótica para las plantillas de los usuarios.

Despliega los menús y reconoce los comandos que solicita el usuario por el mando de control, ya sea el registro de un nuevo usuario o el acceso para abrir el cerrojo electrónico.

2. Lector de huellas FS83: Este módulo puede ejecutar ciertas funciones usando algoritmos internos del fabricante, puede tomar muestras, hacer el enrolamiento, comparar dos huellas entre otros, todo esto por medio de comandos que recibe del microcontrolador a través de comunicación RS-232.

3. Mando de control: En este módulo se tienen los switches para avanzar por el menú de la pantalla LCD, están conectados directamente a uno de los puertos del microcontrolador. Además del GPIO de la pantalla LCD están conectados tres LED's de distintos colores y un buzzer que indican el estado del sistema, ya sea si está procesando algo, si la ultima instrucción fue correcta, si se permite o deniega el acceso al usuario.

4. Pantalla LCD 20x4: La pantalla es un módulo de doble circuito, en el segundo circuito se encuentran los pines de alimentación, comunicaciones y un GPIO; este módulo recibe ordenes del microcontrolador por comunicación I2C, en la pantalla se despliegan los mensajes del menú, de la actividad que está realizando el programa principal, en la figura 5.4 se muestra el esquemático del menú y para cada caso se muestra información adicional para el usuario, en la figura 5.5 se muestra una fotografía del menú de inicio. Se usan los GPIO como salidas para los leds indicadores y el buzzer, ubicados en el mando de control. También se usa una salida a la etapa de potencia para abrir el cerrojo electrónico al aceptar a un usuario.

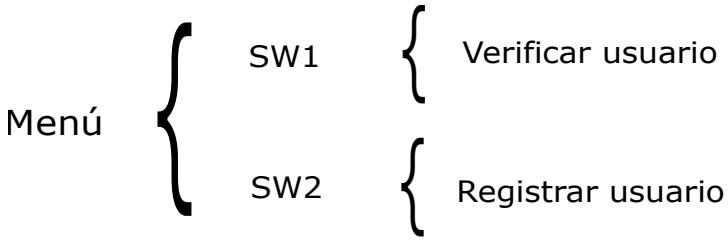


Figura 5.4: Esquemático del menú.

5. Etapa de potencia: Este módulo es un circuito que activa o desactiva un cerrojo electrónico, a la entrada de este módulo se recibe una señal TTL, esta señal polariza un transistor NPN que se utiliza como switch para activar un relevador de 5V, este relevador se utiliza para energiza el selenoide de 12V del cerrojo, permitiendo el acceso al usuario verificado, en la figura 5.6 se presenta el diagrama del circuito.

Figura 5.5: Menú de inicio del sistema embebido.

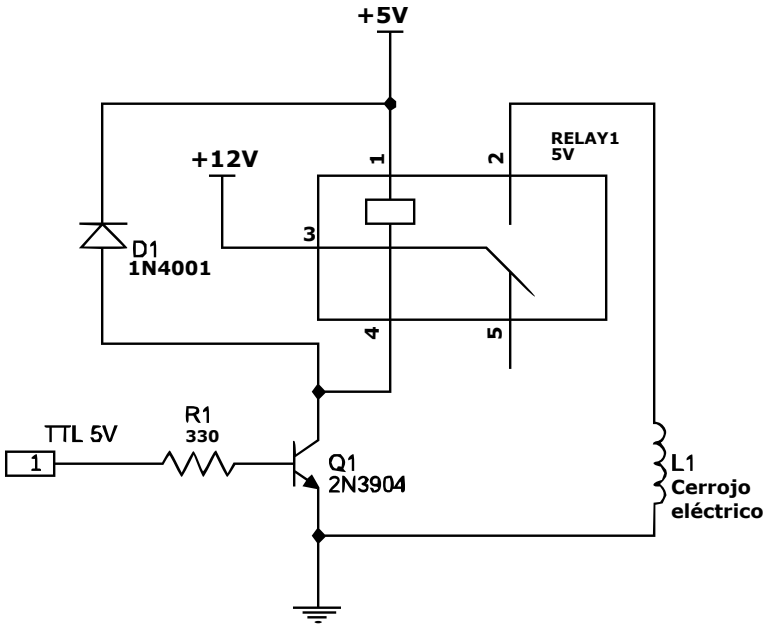


Figura 5.6: Diagrama esquemático del circuito de potencia.

Interfaz de usuario

Los sistemas de acceso tienen tres tipos de reconocimiento, identificación, verificación y matriculación. En la etapa de *enrolamiento*, se toman tres muestras del identificador biométrico mediante un método de extracción de minucias para generar una plantilla, esta plantilla se almacena en una base de datos en el microcontrolador para futuras comparaciones. En esta tesis se propone un método de encriptado usando caos digital, para proteger esta valiosa información. En la etapa de *verificación*, se toma una muestra del módulo identificador biométrico en tiempo real y esta muestra se compara con una plantilla previamente encriptada y almacenada en la base de datos (en este momento se debe desencriptar antes de ser enviada para autenticar al usuario). El proceso de comparación se realiza con los algoritmos del módulo lector de huellas futronic FS83 [44].

Registrar nuevo usuario

En la etapa de *enrolamiento* (figura 5.7), la plantilla del usuario se lee por el módulo FS83 que funciona como esclavo y los datos se envían al microcontrolador vía comunicación RS-232; una vez que la plantilla esta en el microcontrolador, esta se encripta con una clave secreta de 32 números hexadecimales y se almacena en la memoria flash del microcontrolador. En la figura 5.8, se ilustra la interfaz de usuario de este proceso.

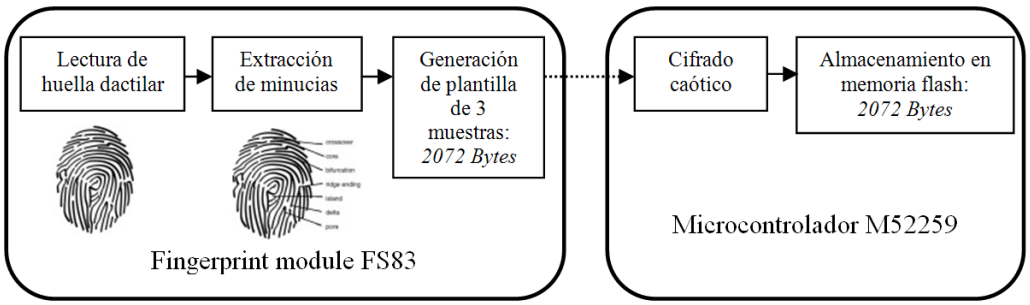


Figura 5.7: Esquema del proceso de enrolamiento del sistema embebido [3].

Abrir cerrojo

En el proceso de *autenticación* (figura 5.9), el usuario solicita su plantilla cifrada al microcontrolador y la plantilla original se recupera mediante el descifrado con la misma clave secreta de 32 números hexadecimales (se plantea utilizar otra huella como clave secreta a trabajo futuro). Después, el microcontrolador envía la plantilla descifrada al módulo lector de huella y se lleva a cabo el proceso de comparación de plantillas con la plantilla actual del usuario para autenticarlo. En la Figura 5.10 se ilustra la interfaz de usuario de este proceso.

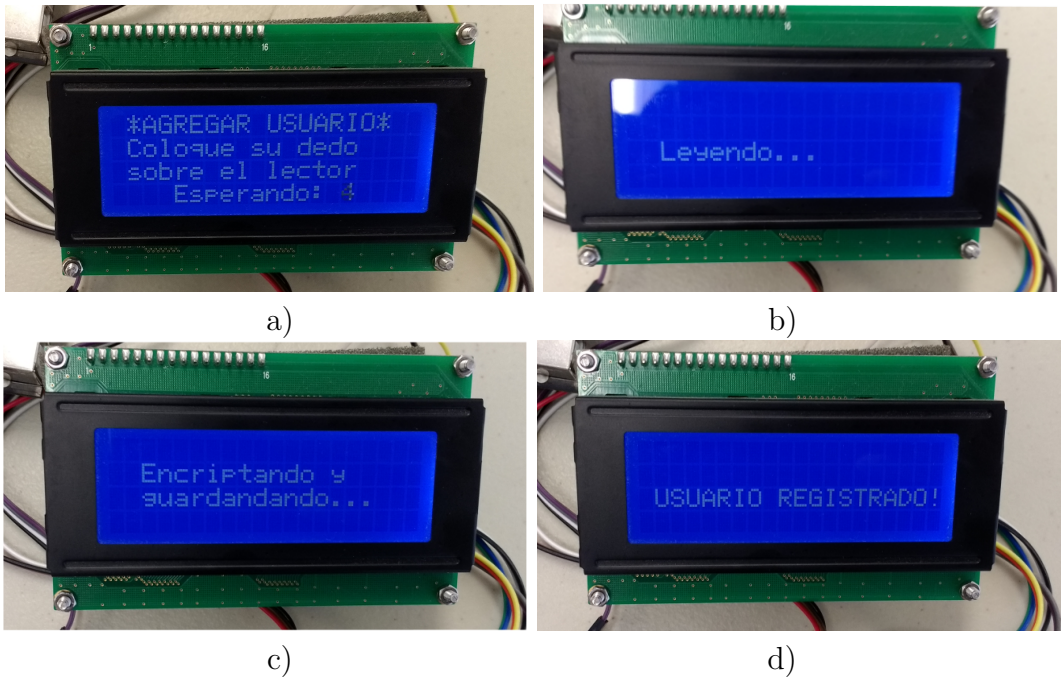


Figura 5.8: Registro de usuario: a) al precionar sw1 se debe colocar el dedo índice en el lector, b) al colocar el dedo, se extrae el templete biométrico y se envía al microcontrolador, c) cifrado caótico del templete y almacenamiento en microcontrolador y d) confirmación de usuario registrado.

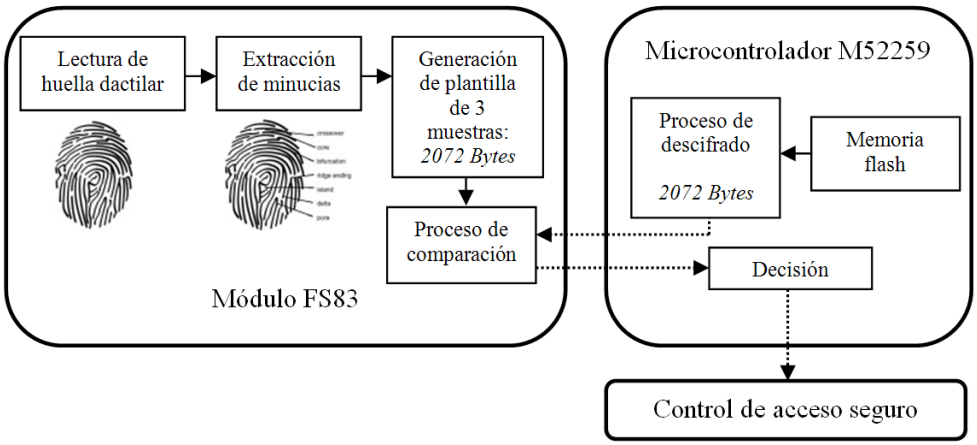


Figura 5.9: Esquema del proceso de autenticación del sistema embebido [3].

a) b) c)

Figura 5.10: Verificación al usuario: a) al presionar el sw2 se debe colocar el dedo registrado del usuario en lector de huellas, b) el microcontrolador desencripta la plantilla biométrica y envía templete a lector de huellas y c) se realiza la verificación del usuario y permite el acceso al sitio restringido.

Recursos del sistema

El algoritmo de cifrado y configuración del microcontrolador requiere de 148 KB (28 %) de un total de 512 KB; la memoria libre restante (364KB) es utilizada para almacenar plantillas cifradas de los usuarios autorizados. Por tanto, el sistema embebido de autentificación puede registrar hasta 179 usuarios con la memoria flash del microcontrolador sin necesidad de usar una memoria externa para la base de datos.

El sistema de autentificación utiliza el módulo FS83, este módulo tiene una tasa de falsa aceptación (FAR) de 10^6 (uno en un millón acepta un falso usuario) y una tasa de falso rechazo (FRR) de 10^2 (una de cada cien rechaza a un usuario autorizado).

El tiempo de registro de un nuevo usuario requiere de 5.5 segundos aproximadamente, lo que incluye lectura de tres muestras, procesar la plantilla, enviar el templete al microcontrolador, cifrado caótico de la plantilla en microcontrolador y almacenamiento en memoria flash. El proceso de autenticación requiere de 4.9 segundos que incluye tomar tres muestra del solicitante, leer la memoria flash del microcontrolador (base de datos), descifrar el criptograma, enviar el templete a comparar contra la muestra del solicitante y el proceso de comparación. El costo del sistema es menor a $250 usd. Por tanto, el esquema propuesto puede ser implementado en aplicaciones en tiempo real a bajo costo.

5.3. Cifrado

Se asume un templete claro $P \in [0, 255]$ de longitud $\ell = 2072$ basado en datos leídos del módulo FS83 que están representados por 8 bits (0-255 en decimal). La figura 5.11 muestra el diagrama a bloques del proceso de cifrado. Se obtienen 2,072 bytes del módulo FS83 como templete claro cada vez que se desea registrar un nuevo usuario.

En el trabajo presentado en [32], se obtiene el atractor hypercaótico de Lotka-Volterra con un rango de estados muy desigual entre x_n y y_n (ver figura 3.10), en base a múltiples pruebas en Matlab se encontró que un mayor balance entre el rango de

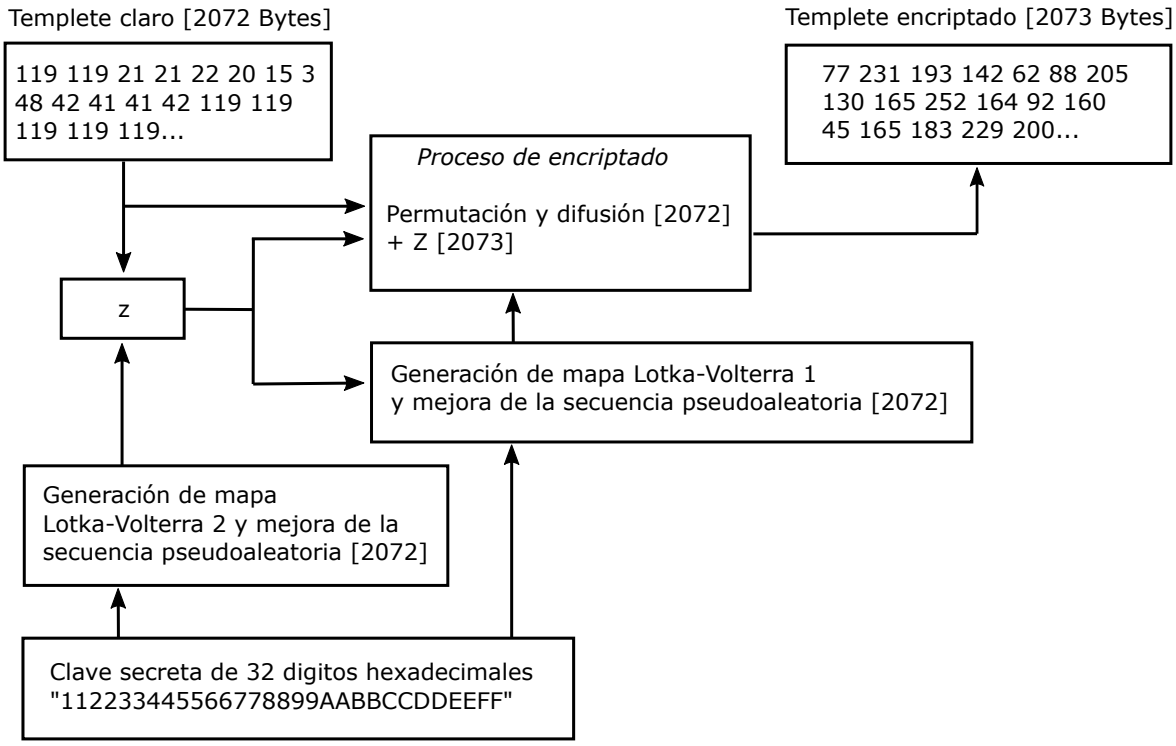


Figura 5.11: Esquema de cifrado del templete de huella dactilar.

ambos estados genera mejores resultados en la prueba de sensibilidad a condiciones iniciales. Para obtener un rango de valores de $0 < y_n < 0.9$ (ver figura 5.12) sin añadir operaciones adicionales se incrementa el parámetro β, en el cifrado se utiliza $\beta = 3.2$.

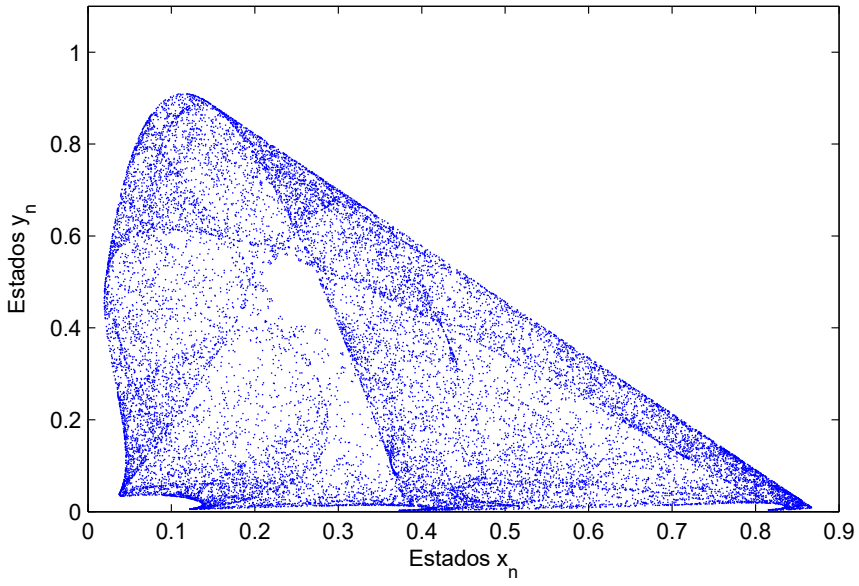


Figura 5.12: Atractor hypercaótico de Lotka-Volterra con rango mejorado.

En este trabajo se utiliza el método presentado por [15] para que el criptograma dependa en parte por el mensaje que se quiere enriptar, para que siempre se generen criptogramas diferentes y mejorar la seguridad. Se utiliza el formato IEEE-754 para núumeros flotantes, para evitar degradación deigital del caos y generar alta sensibilidad de hasta 10^{15} decimales de presición.

La clave de 32 números hexadecimales se separa en cuatro grupos de 8 como sigue

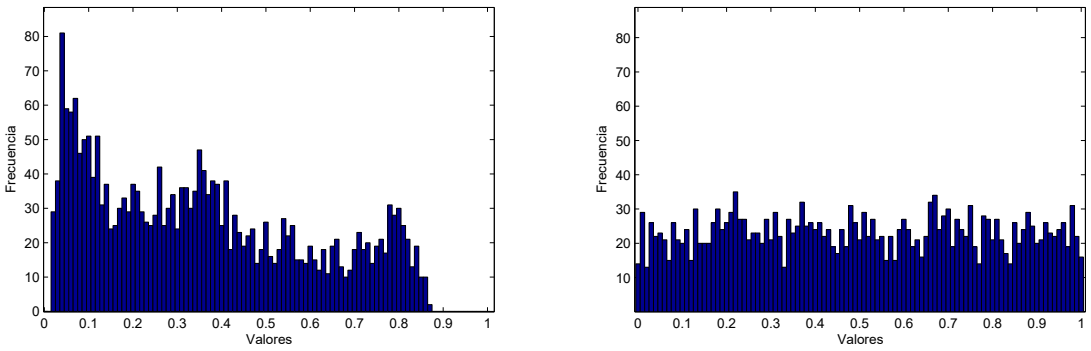
$$CLAVE = 11223344556677889AABBCCDDEEFF \tag{5.1}$$

$$A = \frac{CLAVE_{1-8}}{2^{32}+1}; B = \frac{CLAVE_{9-16}}{2^{32}+1}; C = \frac{CLAVE_{17-24}}{2^{32}+1}; D = \frac{CLAVE_{25-32}}{2^{32}+1} \tag{5.2}$$

El mapa Lotka-Volterra 2 se itera $I_2 = 2,072$ con valores de $\alpha_2 = 3.5$, $\beta = 3.2$, $\delta = 3.9 + \{(A+B)*0.1\}$ (mód 1), $x_2(0) = \{(C+D)*0.1\}$ (mod 1) y $y_2(0) = 0.2$, donde mód es la operación de módulo. Para generar la secuencia caótica de $y_2^{LV2} \in (0,1)$ y precisión 10^{-15}, y_2^{LV2} se mejora pseudoaleatoriamente de la siguiente manera:

$$y_i^{LV2} = \left\{ \left[y_i^{LV2} \right] * 100 \right\} \quad (\text{mód } 1), \quad para\ i = 1, 2, 3, \ldots, 2072 \tag{5.3}$$

donde $y^{LV2} \in (0,1)$ y *mód* es la operación de módulo. En la figura 5.13(a) y 5.13(b) se muestran los histogramas de la secuencia original y mejorada respectivamente.



a) b)

Figura 5.13: Histogramas de la secuencia y: a) secuencia original, b) secuencia mejorada.

Después, los elementos de la plantilla clara se suman con la secuencia caótica y^{LV2}

$$S = \left\{ S + [P_i * y_i^{LV2}] + y_i^{LV2} \right\} \quad (\text{mód } 1), \quad para\ i = 1, 2, 3, \ldots, 2072 \tag{5.4}$$

donde P_i representa el elemento i de la plantilla clara, S es una variable inicializada en cero y *mód* es la operación de módulo. El valor de Z debe ser un valor entero entre 1-255 para evitar la cancelación de Z, por tanto se calcula lo siguiente

$$V = \ round(S * 254) + 1, \tag{5.5}$$

donde $V \in [1, 255]$ y *round* es la operación de redondeo al valor más cercano. Finalmente, el valor de Z está dado por

$$Z = V/256, \tag{5.6}$$

donde $Z \in (0, 1)$ con precisión de 10^{-15}.

El mapa Lotka-Volterra 1 se itera $I_1 = 3000$ con valores de $\alpha_1 = 3.5$, $\beta = 3.2$, $\delta = 3.8 + \{(A + B + Z) * 0.1\}$ (mód 1), $x_1(0) = \{(C + D + Z) * 0.1\}$ (mod 1) y $y_1(0) = 0.1$ para generar la secuencia caótica de datos y^{LV1} con $y^{LV1} \in (0, 1)$ y una precisión de 10^{-15}. De esta secuencia caótica se determinan dos subsecuencias, una para el proceso de confusión y otra para difusión como sigue

$$Q_i = \ round\left(y^{LV1}_{3000-i} * 2071\right) + 1, \quad para\ i = 1, 2, 3, \ldots, 2072 \tag{5.7}$$

donde $Q_i \in [1, 2072]$ es el vector de confusión pseudoaleatorio. Se considera que el vector Q_i esta mejorado pseudoaleatoriamente, es decir, posee todos los valores de 1-2072.

La segunda subsecuencia para el proceso de difusión se determina como sigue

$$F_i = \left\{\left(x^{LV1}_{3000-i} * 100\right) + Z\right\} \quad (\text{mód } 1), \quad para\ i = 1, 2, 3, \ldots, 2072 \tag{5.8}$$

donde $F_i \in (0, 1)$ es un vector de longitud de 2072 elementos con precisión de 10^{-15}. Después, F_i se transforma de $(0, 1)$ a $[0, 255]$ como sigue

$$Y_i = \ round\left(M_i * 255\right), \quad para\ i = 1, 2, 3, \ldots, 2072 \tag{5.9}$$

donde $Y_i \in [0, 255]$ es un vector de longitud 2072. Para el proceso de cifrado se permutan los elementos en orden caótico (Q_i) mientras se difuminan los elementos al sumar valores discretos de caos (Y_i):

$$E_i = (P(Q_i) + Y_i) \ \ mod\ 256, \quad donde\ i = 1, 2, 3, \ldots, 2072 \tag{5.10}$$

donde $E_i \in [0, 255]$ es la plantilla cifrada y P es la plantilla clara. Finalmente, el valor de Z se agrega al criptograma en la posición 2,073

$$E_{2073} = V. \tag{5.11}$$

El **proceso de descifrado** consiste básicamente en invertir los pasos del cifrado y se considera que el criptograma no fue modificado durante su transmisión y almacenamiento. Primero, Z se recupera del criptograma como sigue

$$Z = E_{2073}/256. \tag{5.12}$$

Después, y^{L1} se genera con la misma clave secreta utilizada en el cifrado; posteriormente, se determinan Q_i y Y_i de la misma forma que el proceso de cifrado. Finalmente, la plantilla descifrada se calcula con la siguiente expresión

$$D(Q_i) = (E_i - Y_i) \ \ mod\ 256, \quad para\ i = 1, 2, 3, \ldots, 2072 \tag{5.13}$$

donde Q_i es el vector de confusión, E_i es el criptograma, Y_i es el vector de difusión y D es la plantilla recuperada.

5.4. Análisis de seguridad

Para los análisis de seguridad, se extrae del microcontrolador el templete de una huella, un mapa Lotka-Volterra y un criptograma. Se usan las librerías de MQX 3.5.1 para guardar estos datos en una memoria USB conectada a la tarjeta de desarrollo y poder realizar los análisis de seguridad en MatLab. Primero se simularon en Matlab usando texto claro y se encriptaron los caracteres de codigo ASCII con el algoritmo propuesto para microcontrolador.

5.4.1. Espacio de clave secreta

El sistema propuesto utiliza una clave secreta de 32 números hexadecimales, esto equivale a un espacio de claves de 2^{128} posibilidades, por lo que puede resistir un ataque exhaustivo.

5.4.2. Sensibilidad a clave secreta

En el Capítulo 3, se menciona que una de las características fundamentales del caos, es que una pequeña variación en las condiciones iniciales representan un enorme cambio en la trayectoria de los mapas caóticos, esto se aprovecha en el algoritmo de encriptado utilizando la clave secreta en parte como una de las condiciones iniciales x_0, y_0 y otra parte se usa en el parámetro de control δ en las ecuaciones, de manera que el cifrado responda de manera drástica a pequeños cambios en la clave secreta. La sensibilidad a la clave secreta se prueba con dos claves secretas con un cambio de una unidad en el byte menos significativo (ver Tabla 5.1). Para realizar este análisis, se utiliza la prueba de NPCR y UACI para demostrar la sensibilidad de la clave secreta al proceso de cifrado, Se aprecia que cada criptograma tiene su propia ruta, por tanto el algoritmo de cifrado es altamente sensible a la clave secreta.

No. clave	Clave secreta
Clave 1	11223344556677889900AABBCCDDEEFF
Clave 2	11223344556677889900AABBCCDDEEF**E**

Tabla 5.1: Claves secretas utilizadas para análisis de sensibilidad a la clave en cifrado en el templete de una huella dactilar.

El valor de *NPCR* se determina con

$$NPCR = \frac{\sum_{i=1}^{i=2073} W(i)}{2073} \times 100 \tag{5.14}$$

donde

$$W(i) = \begin{cases} 0 & \text{if } E_1(i) = E_2(i) \\ 1 & \text{if } E_1(i) \neq E_2(i) \end{cases} \tag{5.15}$$

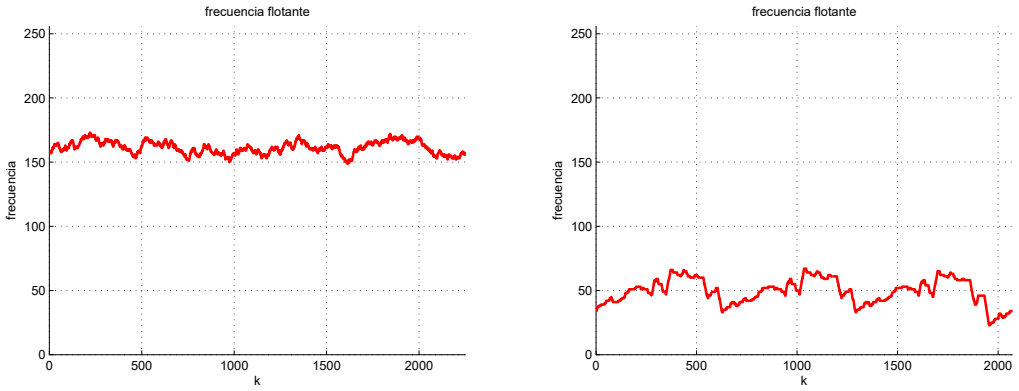
y el valor de $UACI$ se determina con

$$UACI = \frac{100}{2073} \sum_{i=1}^{i=2073} |E_1(i) - E_2(i)| \tag{5.16}$$

donde E_1 y E_2 son los dos criptogramas e i es el byte en el vector del criptograma, dando como resultado NPCR = 99.5561 % y UACI = 88.6785 %.

5.4.3. Frecuencia flotante

La *frecuencia flotante* determina cuantos símbolos son diferentes en una sección del mensaje. Esta información puede utilizarse para determinar si la plantilla cifrada se divide en varias secciones, cada sección debe tener todos los posibles símbolos de forma uniforme. La prueba consiste en seleccionar los primeros 256 elementos del templete y después, la ventana se recorre una posición a la derecha y la prueba se repite sucesivamente hasta formar un barrido de todo el mensaje y termina en la posición inicial. En la figura 5.14, se observa que la frecuencia flotante de la plantilla cifrada es uniforme, por tanto, el cifrado propuesto no tiene secciones débiles. Además, el templete cifrado tiene hasta 62.9948 % de todos los posibles elementos contra 29.3721 % del templete claro.

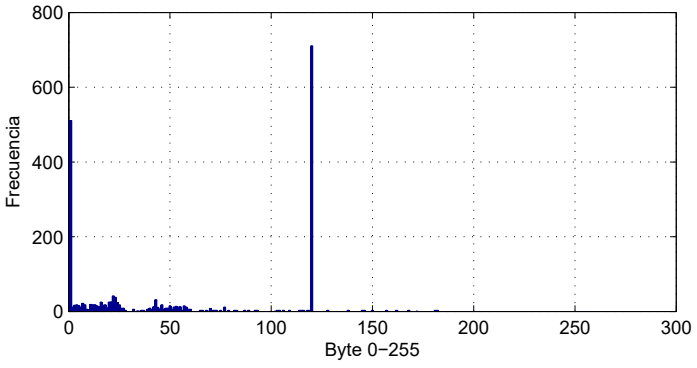


a) b)

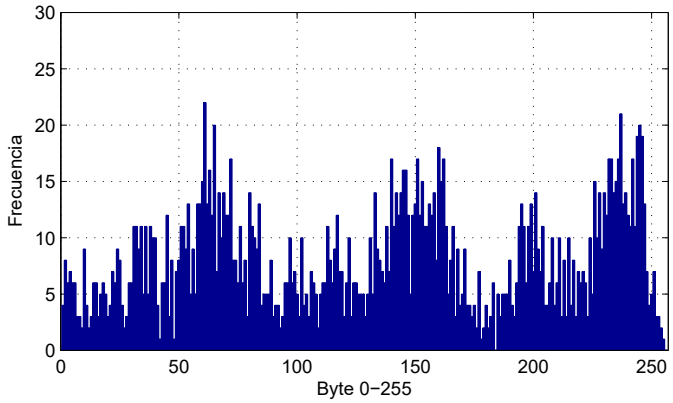
Figura 5.14: Análisis de frecuencia flotante: a) templete encriptado y b) templete claro.

5.4.4. Histogramas

El histograma de la plantilla cifrada debe ser uniforme para resistir un ataque estadístico. Como se puede notar en la figura 5.15(a), una parte importante de los datos son ceros y otra gran cantidad es el valor 119. En cambio el histograma del templete enriptado en la figura 5.15(b) se tiene un rango de valores notablemente distribuido.



a)



b)

Figura 5.15: Histogramas del templete de la huella digital: a) templete claro y b) templete encriptado.

5.4.5. Autocorrelación

Con esta prueba se puede determinar si la plantilla cifrada tiene patrones repetitivos, periodicidad o alguna dependencia. Esta se calcula con

$$AC(k) = \frac{A - D}{T}, \tag{5.17}$$

donde $AC \in [1 - 1]$ es la autocorrelación del mensaje desplazado k posiciones, A es el número de elementos que concuerdan entre el mensaje original y el mensaje desplazado, D es el número de elementos que no concuerdan y T es la longitud del mensaje.

En este análisis, se calcula la autocorrelación a nivel bit y se utilizan 8 bits para representar cada elemento de la plantilla. Primero, la autocorrelación de la plantilla clara se calcula como sigue: los 2,072 elementos se transforman a un renglón de 16,576 bits: *1001001110*...; después, se determina la autocorrelación del mensaje con un valor de hasta $k = 20000$ hacia la derecha.

La figura 5.16(a) y 5.16(b) muestra la autocorrelación de la plantilla clara y de la plantilla cifrada, respectivamente. La autocorrelación del templete claro tiene patrones repetitivos con altos valores positivos AC, en cambio el templete encriptado tiene una autocorrelación cercana a 0. Cabe destacar que el pico máximo en las dos gráficas ocurre cuando el barrido de k alcanza los 16,576 de desplazamiento, es decir cuando ha vuelto a la posición inicial. Podemos decir entonces que el proceso de cifrado genera números pseudoaleatorios con un alto grado de distribución en el vector del criptograma de 2073 bytes.

5.4.6. Entropía de la información

La *entropía* determina qué tan impredecible es cierto mensaje. A mayor entropía, más impredecible es un mensaje. El cifrado de plantilla utiliza 256 símbolos diferentes con una entropía máxima de $H = 8$. La entropía de la plantilla clara es de $H(P) = 4.0218$, mientras que la plantilla cifrada tiene $H(E) = 7.7996$. Podemos decir entonces que se logra obtener todos los símbolos con una probabilidad uniforme. La entropía $H(m)$ de un mensaje m puede calcularse como sigue

$$H(m) = \sum_{i=0}^{2^N - 1} p(m_i)\ log_2(1/p(m_i)), \tag{5.18}$$

donde N es el número de bits que representan la unidad básica del mensaje m, 2^N son todas las combinaciones de la unidad básica, $p(m_i)$ representa una probabilidad de m_i, $\log_2$ es el logaritmo base 2 y la entropía esta expresada en bits, donde la máxima entropía es N.

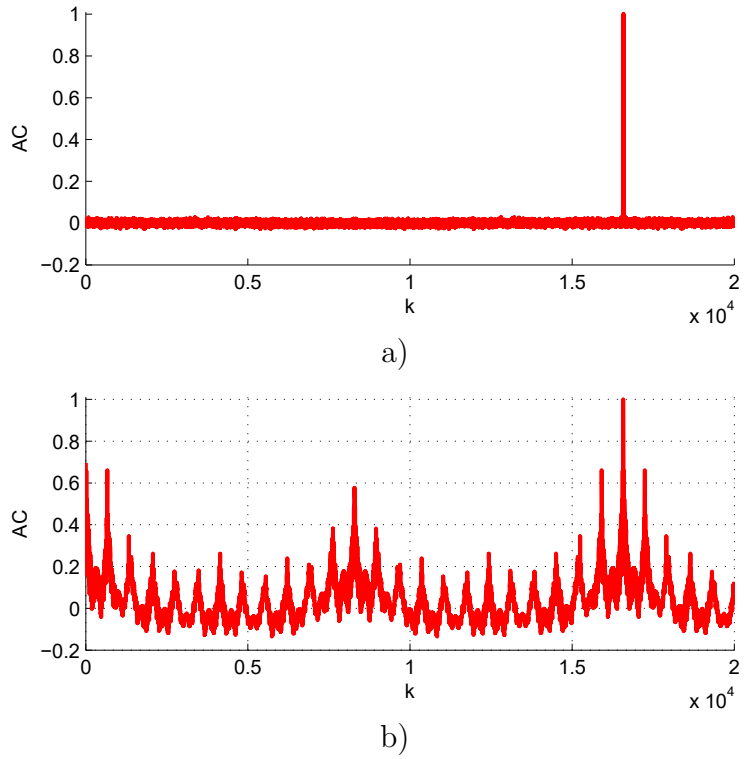


Figura 5.16: Autocorrelación del templete: a) templete claro, b) templete encriptado.

5.4.7. Análisis de aleatoriedad FIPS-140-2

El análisis de aleatoriedad esta basado en las siguientes pruebas de acuerdo con el estándar FIPS-140-2 [45]: *monobit*, *poker*, y *serial*. Los resultados de estas pruebas están basados en la distribución Chi-cuadrada χ^2 con un nivel de significancia de $\alpha = 0.05$ y un grado de libertad variable.

El criptograma debe tener el mismo número de 0's y 1's idealmente. La prueba de *monobit* determina cuantos 0's y 1's tiene un criptograma. La prueba *monobit* se calcula con la siguiente expresión

$$Mt = (zeros - ones)^2 / (2073 * 8) \tag{5.19}$$

donde Mt es la prueba de *monobit* del templete cifrado, *zeros* es el número de 0's en el criptograma y *ones* es el número de 1's en el criptograma. Se considera una distribución de chi-cuadrada χ^2 de un grado de libertad y el rango critico para Mt debe ser menor a 3.814.

La prueba *poker* verifica si una secuencia de longitud m y todas sus combinaciones se reproduce con la misma probabilidad; esta prueba se determina como sigue

$$Pt = \left(\frac{2^m}{k}\right)\left(\sum_{i=1}^{i=c} sum(i)\right) - k, \tag{5.20}$$

donde Pt es la prueba *poker*, $sum(i)$ es la concurrencia de la secuencia i en el criptograma, $m = 3$ es la longitud de la secuencia, $c = 2^3$ son todas las posibles combinaciones de $m \in (000, 001, ...111)$ y $k = (2073 * 8)/3$ son los trigramas no traslapados del criptograma; de acuerdo con la tabla de χ^2 y el grado de libertad definido por $2^m - 1 = 7$, el umbral de Pt máximo es 14.07.

La prueba *serial* revisa si un bit es dependiente de su bit predecesor, calculado mediante

$$St = \left(\frac{4}{L}\right)\left((N_{00})^2 + (N_{01})^2 + (N_{10})^2 + (N_{11})^2\right) - \left(\frac{2}{L}\right)\left((N_0)^2 + (N_1)^2\right) + 1, \quad (5.21)$$

donde St es la prueba *serial*, $L = (2073 * 8) - 1$ es la longitud, N_0 y N_1 son los números de 0's y 1's en el criptograma y $N_{00,01,10,11}$ son los números importantes de la subsecuencia con traslape; la distribución χ^2 con dos grados de libertad es de 5.991. En la Tabla 5.2 se recopilan los resultados de estas pruebas y se obserba que pasa la prueba monobit y serial, mientras que la prueba póker sobrepasa el umbral.

Prueba	Resultado	Limite por FIPS-2
Monobit	Mt = 0.078147	3.814
Póker	Pt = 25.4616	14.07
Serial	St = 1.2858	5.991

Tabla 5.2: Resultados para el criptograma de un templete de huella dactilar.

5.5. Conclusiones

El sistema embebido propuesto cifra la plantilla de huella dactilar mediante el algoritmo propuesto en [15], adaptando el mapa hypercaótico de Lotka-Volterra para generar la secuencia caótica (utilizando por primera vez en criptografía caótica), se utilizan los algoritmos internos del lector FS83 [44] para generar los templetes y el proceso de verificación por su sencillez y la alta presición en FRR-FAR; la etapa de potencia del sistema es controlada mediante uno de los puertos de salida de la pantalla Futronic el criptograma generado con este algoritmo puede ser considerado seguro, de acuerdo con las pruebas estadísticas que se presentaron. Los tiempos de registro y de acceso son relativamente cortos, con 4.9 segundos para acceder. Los recursos de implementación son de bajo costo ya que no se requieren muchos elementos de hardware para implementar un sistema de acceso seguro con la protección de las identidades de los usuarios aseguradas gracias al encriptado caótico. Además este algoritmo de encriptado incrementa el grado de aleatoriedad cuando se utilizan bloques más grandes de información como en el caso de texto.

Capítulo 6

Conclusiones

6.1. Conclusiones generales

En este trabajo de tesis, se diseñó e implementó un algoritmo criptográfico basado en caos y biometría, en un sistema embebido para el control de acceso seguro basado en el algoritmo [15]. Primeramente, se analizó el exponente de Lyapunov a los mapas caóticos: lógistico, Hénon, Bernolli generalizado, Chebyshev, lógistico 2D y Lotka-Volterra hypercaótico siendo este último elegido por tener dinámica hypercaótica con sólo dos dimensiones y usar polinomios de primer orden. Sin embargo, para poder usar este mapa en el algoritmo de encriptado, se hace un ajuste a uno de los parámetros reportados en la literatura y se hace una operación con módulo 1, de manera que los estados del mapa cubran todo el rango de valores de 0 a 1.

El algoritmo de encriptado caótico es de clave simétrica, cifrado de flujo, y arquitectura de *confusión y difusión*; el sistema embebido de acceso seguro presentado en los experimentos utiliza una clave secreta de 32 números hexadecimales programada en el IDE de Freescale *CodeWarrior*; para leer, generar y verificar los templetes de las huellas dactilares se usa el lector Fultronic FS83 que lo hace de forma independiente con sus algoritmos internos, lo que facilita el desarrollo de este tipo de sistemas; el espacio de claves puede resistir un ataque exhaustivo y los resultados de las pruebas estadísticas demuestran un alto grado de aleatoriedad, sensibilidad a la clave secreta, con buena distribución de histogramas y no presenta periodicidad en la prueba de autocorrelación.

6.2. Trabajo a futuro

Como trabajo a futuro se plantea lo siguiente:

- Criptografía caótica y ADN para sistemas biométricos
- Protección de plantillas de más de un identificador biométrico.
- Sistema de mayor seguridad: encriptado de plantillas biométricas y reconocimiento en tiempo real de identificador conductual.

- Implementaión en FPGA's.
- Clave secreta basado en algún identificador biométrico.

Bibliografía

[1] Ortega Morán A. (2013). De dónde viene. *Editorial Algarabía y Lectorum*, pp. 98-100.

[2] Pecora L.M. y Carroll T.L. (1990). Synchronization in chaotic systems. *Physical Review Letters*, **64**(8): 821-824.

[3] Cuomo, K.M., Oppenheim, A.V., y Strogatz, S.H. (1993). Synchronization of Lorenz-Based Chaotic Circuits with Applications to Communications. *IEEE T. Circuits-II*, **40**(10): 626-633.

[4] Cruz-Hernández, C. y Nijmeijer, H. (2000). Synchronization Through Filtering. *International Journal of Bifurcation and Chaos.*, **10**(4): 763-775.

[5] Posadas-Castillo C., López-Gutiéerrez R.M., y Cruz-Hernández C. (2007). Synchronization of chaotic solid-state Nd:YAG lasers: Application to secure communication. *ELSEVIER.*, Communications in Nonlinear Science and Numerical Simulation **13**(8): 1655-1667.

[6] López-Gutiérrez R.M., Posadas-Castillo C., López-Mancilla D., y Cruz-Hernández C. (2009). Communicating via robust synchronization of chaotic lasers. *Science Direct*, Chaos, Solitons y Fractals. **42**(1): 277-285.

[7] Arellano-Delgado A., López-Gutiérrez R.M., Cruz-Hernández C., Posadas-Castillo C., Cardoza-Avendaño L., y Serrano-Guerrero H. (2012). Experimental network synchronization via plastic optical fiber. *Science Direct*, Optical Fiber Technology. **19**(12): 93-108.

[8] Murillo Escobar M.A., Cruz-Hernández C., Abundiz-Pérez F., López-Gutiérrez R.M. y Acosta Del Campo O.R. (2015). A RGB image encryption algorithm based on total plain image characteristics and chaos. *Signal Processing*, **109**: 119–131.

[9] Fridrich J. (1998). Symmetric ciphers cased on two-dimensional chaotic maps. *International Journal of Bifurcation and Chaos*, **8**(6): 1259–1284.

[10] Murillo-Escobar M.A., Cruz-Hernández C., Abundiz-Pérez F., y López-Gutiérrez R.M. (2015). A robust embedded biometric authentication system based on fingerprint and chaotic encryption. *Expert Systems with Applications*, **42**(21): 8198-8211.

[11] Abúndiz-Pérez F., Cruz-Hernández C., Murillo-Escobar M.A., López-Gutiérrez R., y Arellano-Delgado A. (2016). A Fingerprint Image Encryption Scheme Based on Hyperchaotic Rössler Map. *Mathematical Problems in Engineering*, **2016**(15): 8198-8211.

[12] Abundiz-Pérez F. (2016). Seguridad en un sistema biométrico basado en huellas dactilares aplicando caos. Tesis doctoral, *Universidad Autónoma de Baja California*.

[13] Lin, C.F. (2016). Chaotic visual cryptosystem using empirical mode decomposition algorithm for clinical EEG signals. *Journal of Medical Systems*,**40**(52): 1-10.

[14] Murillo-Escobar M.A., Cardoza-Avendaño L., López-Gutiérrez R.M.(2017). A double chaotic layer encryption algorithm for clinical signals in telemedicine. *Journal of Medical Systems*. **41**(59): 1-17.

[15] Murillo Escobar M. A. (2015). Diseño de un algoritmo de cifrado caótico y su implementación para aplicaciones embebidas. Tesis doctoral, *Universidad Autónoma de Baja California*.

[16] Forbes México (2015). Biometría: acceso seguro a la movilidad. consultado en: `https://www.forbes.com.mx/biometria-acceso-seguro-la-movilidad/` el 25 de Mayo de 2017.

[17] Inzunza-González M.A. (2012). Encriptado caótico en sistemas biométricos. Tesis doctoral, *Universidad Autónoma de Baja California*.

[18] Serratosa F. (2015), PID: 00195448. La biometría para la identificación de las personas. *Universidad Oberta de Catalunya*, pp 50.

[19] Rosales Cruz A. (2009). Clasificación de huellas digitales mediante minucias. *Instituto Nacional de Astrofísica, Óptica y Electrónica*, pp 50.

[20] UNAM Coordinación de Seguridad de la Información (2015). Roban millones de huellas dactilares a gobierno de EE.UU. consultado en: `https://www.seguridad.unam.mx/historico/noticia/index.html-noti=2511` el 20 de mayo de 2017.

[21] Leys J., Ghys É. y Alvarez A. (2013). Movimiento y determinísmo. *Creative Commons*, `http://www.chaos-math.org/es` (2017-05-10).

[22] Gómez de la Cruz J. A. (2016). Transmisión óptica en red utilizando fibra óptica de plástico. Tesis de licenciatura, *Universidad Autónoma de Baja California*.

[23] Lorenz E. N. (1963). Deterministic non periodic. *Journal of the Atmoferic Sciences*, volume 20 **109**: 130-142.

[24] Wolf, A. (1986). Quantifying chaos with Lyapunov exponents. Princeton. *University Press*,Ch. 13, pp. 273–289.

[25] Cvitanovicé P., Artuso R., Mainieri R., Tanner G. y Vattay G. (2016). Chaos: Classical and Quantum. *ChaosBook.org*, **115**: 998-1017.

[26] May R. M. (1976). Simple mathematical models with very complicated dynamics. *Nature*, **261**: 459–467.

[27] Hernández de la Sota C. (2004). Control inteligente de sistemas dinámicos caóticos. *Universidad Politécnica de Madrid*, Tesis doctoral.

[28] Ye R. y Ma Y.(2013). A Secure and Robust Image Encryption Scheme Based on Mixture of Multiple Generalized Bernoulli Shift Maps and Arnold Maps. *MECS*,I. J. Computer Network and Information Security.

[29] Ye R. (2011). An Image Encryption Scheme with Efficient Permutation and Diffusion Processes. *Springer*, In: Zhou M., Tan H. (eds) Advances in Computer Science and Education Applications. Communications in Computer and Information Science, Berlin, Heidelberg **202**: pp 32-39 613.

[30] K. Prasadh y R. Gnanajeyaraman (2009). Analysis of Chaotic-Chabyshev polinomials using on public key cryptosystems. *Geogian Electronic Scientific Journal*, Computer science and Telecommunications.

[31] Ahmad M., Farooq O. (2010). A Multi-Level Blocks Scrambling Based Chaotic Image Cipher. *Springer, Berlin, Heidelberg*, In: Ranka S. et al. (eds) Contemporary Computing. IC3 2010. Communications in Computer and Information Science **94**: pp 171-182.

[32] Abrego F.J. y Moiola J.L. (2013). Lyapunov Exponent Analysis Applied to a Hyperchaotic Prey-predator Model. *IEEE Latin America Transactions*, Vol 1 .

[33] Pecora L. M. y Carroll T. L. (1991). Synchronizing chaotic systems. *IEEE Transactions in Circuits Systems*, **38**: 454-456.

[34] Volos C. K. (2010). Chaotic Random Bit Generator Realized with a Microcontroller. *Journal of Computations and Modelling*, pp. 115-136.

[35] Borjón Nieto J. J. (2002). Caos, orden y desorden: en el sistema monetario y financiero internacional, el caso de México. *Plaza y Valdez*

[36] Crespo Chávez R.(2016). Trayectorias programadas en Drones. Tesis de licenciatura, *Universidad Autónoma de Baja California*.

[37] UNAM Facultad de Ingeniería (2012). Historia de la Criptografía. `http://redyseguridad.fi-p.unam.mx/proyectos/criptografia/criptografia/index.php/1-panorama-general/12-historia-de-la-criptografia` (2017-05-22).

[38] Alvarez G. y Li S. (2006). Some Basic Cryptographic Requirements for Chaos-Based Cryptosystems. *International Journal of Bifurcation and Chaos.* **16**(8): pp. 2129-2151.

[39] Khan M. K., Zhang J., y Tian L. (2007). Chaotic secure content-based hidden transmission of biometric templates. *Chaos, Solitons and Fractals*, **32**: 1749–1759.

[40] Stanciu M. y Datcu O. (2012). Atmel AVR microcontroller implementation of a new enciphering algorithm based on a chaotic generalized Hénon map. *9th International Conference on Communications, Bucharest, Rumania,*, 21-23 junio de 2012, 319-322.

[41] Murillo Escobar M.A., Cardoza Avendaño L., López Gutiérrez R.M. y Cruz Hernández C. (2016). Cifrado caótico simétrico de ECG y EEG para aplicaciones en telemedicina. *Congreso Latinoamericano de Control automático (CLCA 2016)*, Medellin, Colombia, 612-617.

[42] MCF52259 (2008). MCF52259RM ColdFire Integrated reference manual. *MCF52259RM*, Rev 2 8/2009.

[43] Matrix Orbital LK204-25 (2015). Manual for PCB revision 3.0 of the LK204-25, LK204-25-422, and LK204-25-USB displays.. *Matrix Orbital* `https://www.matrixorbital.ca/manuals/lk-series/lk204-25-manual?download=141:manual-for-pcb-revision-3-0`.

[44] Futronic (2014). Futronic FS83C PIV serial Fingerprint Authentication Module(sFAM). *Futronic Technology Company Limited,* `http://www.futronictech.com/download/FS83C/brochure.pdf`.

[45] FIPS 140-2 U.S. Department of Commerce / National Institute of Standards and Technology (2001). Security requirements for cryptographic modules FIPS 140-2. *Federal Information Processing Standards Publication*, FIPS 140-2.