UNIVERSIDAD DE COSTA RICA
SISTEMA DE ESTUDIOS DE POSGRADO

CIFRADO DE LLAVE PÚBLICA UTILIZANDO MAPAS ACOPLADOS PRESENTES EN LA NATURALEZA PARA GENERAR CIFRADOS RESISTENTES A ATAQUES CUÁNTICOS

Tesis sometida a la consideración de la Comisión del Programa
de Estudios de Posgrado en Computación e Informática
para optar al grado y título de
Doctorado Académico en Computación e Informática

HUGO SOLÍS SÁNCHEZ

Ciudad Universitaria Rodrigo Facio, Costa Rica

2020

A la más linda

Agradecimientos

A Dios por todas sus bendiciones, gracias por cada uno de tus regalos.

A mi profesora consejera, Gabriela Barrantes, por cada uno de sus geniales aportes a este trabajo y toda la ayuda brindada para hacer esto posible. A los profesores lectores Manuel Ortega y Arturo Camacho por la atención prestada a este trabajo y las horas dedicadas a su corrección.

A Leda Roldán, Daniel Azofeifa, Neville Clark y Roberto Magaña, mis profesores y amigos en este camino recorrido en la Ciencia.

Para mis increíbles amigos Gerardo Lacy, José David Cojal, Laura Rojas, Allan Lacy, Rodrigo Castillo, Ana Segura y Pamela Saborio, con los que he compartido el caminar y el entusiasmo en la Física ya por muchos años.

A Helena Oses Alvarado, por su cariño y esperanza en la vida.

A mi hermano Hary, que tiene la paciencia y el amor para un hermano poco común.

A mi padre Hugo Solís Ríos, que sin su ayuda y sin su apoyo ninguno de mis logros serían posibles. Gracias por todo, gracias por estar para mi por siempre.

A mi madre Katty Sánchez Marín por el regalo de la vida, y por cada uno de sus consejos, por su visión y sus pensamientos de vanguardia. Gracias Mamá por ser la mejor del mundo.

Y a cada uno de mis amigos, familiares, compañeros, profesores y estudiantes por aportar a mi carrera y mi vida, algo de las suyas. Gracias de todo corazón ♡.

"Esta tesis fue aceptada por la Comisión del Programa de Estudios de Posgrado en Computación e Informática de la Universidad de Costa Rica, como requisito parcial para optar al grado de Doctorado Académico en Computación e Informática"

Dr. Luis Guerrero Blanco
Representante del Decano del Sistema de Estudios de Posgrado

Dra. Gabriela Barrantes Sliesarieva
Profesora Guía

Dr. Manuel Ortega Rodríguez
Lector

Dr. Arturo Camacho Lozano
Lector

Dr. Ricardo Villalón Fonseca
Representante de la Directora del Programa de Posgrado en Computación e Informática

---

Hugo Solís Sánchez
Sustentante

Índice general

Resumen

Hoy en día, existe una gran necesidad de crear criptosistemas nuevos y robustos, debido a la amenaza que representa las arquitecturas cuánticas. Los sistemas dinámicos son prometedores en el desarrollo de sistemas criptográficos debido a la estrecha relación entre ellos y los requisitos criptográficos. La encriptación de dinámica distribuida (EDD) representa el primer método matemático para generar un criptosistema de llave pública basado en dinámicas caóticas. Sin embargo, se ha descrito que la propuesta EDD tiene un punto débil en el proceso de descifrado relacionado con la eficiencia y la practicidad. En este trabajo, adaptamos el EDD a un sistema caótico de baja dimensión para evaluar la debilidad y seguridad de la adaptación en un ejemplo realista. Específicamente, utilizamos un mapa acoplado logístico no simétrico, que tiene múltiples atractores caóticos y uno derivado de un sistema físico real, como son las oscilaciones alrededor de un agujero negro. Se creó una implementación completa con un costo computacional y una velocidad aceptables para EDD, lo cual es esencial porque proporciona un requisito criptográfico clave para los criptosistemas basados en el caos.

Abstract

Nowadays, there is a high necessity to create new and robust cryptosystems due to the threat that represents the quantum architectures. Dynamical systems have promised to develop crypto-systems due to the close relationship between them and the cryptographic requirements. Distributed dynamic encryption (DDE) represents the first mathematical method to generate a public-key cryptosystem based on chaotic dynamics. However, it has been described that the DDE proposal has a weak point in the decryption process related to efficiency and practicality. In this work, we adapted the DDE to a low-dimensional chaotic system to evaluate the weakness and security of the adaption in a realistic example. Specifically, we used a non-symmetric logistic coupled map, which is known to have multiple chaotic attractors and one obtained from a real physical system, like the oscillations around a black hole. We created a full implementation with acceptable computational cost and speed for DDE, which it is essential because it provides a key cryptographic requirement for chaos-based cryptosystems.

Lista de Figuras

Lista de Tablas

Lista de Abreviaturas

ELP Encriptación de llave pública.

EDD Encriptación de dinámica distribuida.

EQR Encriptación cuántico resistente.

NL No lineal.

RMA Redes de mapas acoplados.

RSA Sistema criptográfico desarrollado por Rivest, Shamir, Adleman

MCD Máximo común divisor

UNIVERSIDAD DE COSTA RICA

SEP Sistema de Estudios de Posgrado

Autorización para digitalización y comunicación pública de Trabajos Finales de Graduación del Sistema de Estudios de Posgrado en el Repositorio Institucional de la Universidad de Costa Rica.

Yo, Hugo Solís Sánchez, con cédula de identidad 112090519, en mi condición de autor del TFG titulado Cifrado de llave pública utilizando mapas acoplados presentes en la naturaleza para generar cifrados resistentes a ataques cuánticos

Autorizo a la Universidad de Costa Rica para digitalizar y hacer divulgación pública de forma gratuita de dicho TFG a través del Repositorio Institucional u otro medio electrónico, para ser puesto a disposición del público según lo que establezca el Sistema de Estudios de Posgrado. **SI** [X] **NO** *[ ]

***En caso de la negativa favor indicar el tiempo de restricción: _____ año (s).**

Este Trabajo Final de Graduación será publicado en formato PDF, o en el formato que en el momento se establezca, de tal forma que el acceso al mismo sea libre, con el fin de permitir la consulta e impresión, pero no su modificación.

Manifiesto que mi Trabajo Final de Graduación fue debidamente subido al sistema digital Kerwá y su contenido corresponde al documento original que sirvió para la obtención de mi título, y que su información no infringe ni violenta ningún derecho a terceros. El TFG además cuenta con el visto bueno de mi Director (a) de Tesis o Tutor (a) y cumplió con lo establecido en la revisión del Formato por parte del Sistema de Estudios de Posgrado.

INFORMACIÓN DEL ESTUDIANTE:

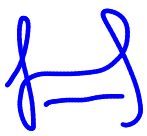
Nombre Completo: Hugo Solís Sánchez.

Número de Carné: A24971 Número de cédula: 112090519.

Correo Electrónico: hugo.solis@ucr.ac.cr.

Fecha: 27 de Abril de 2020. Número de teléfono: 88337917.

Nombre del Director (a) de Tesis o Tutor (a): Gabriela Barrantes Sliesarieva.



FIRMA ESTUDIANTE

Nota: El presente documento constituye una declaración jurada, cuyos alcances aseguran a la Universidad, que su contenido sea tomado como cierto. Su importancia radica en que permite abreviar procedimientos administrativos, y al mismo tiempo genera una responsabilidad legal para que quien declare contrario a la verdad de lo que manifiesta, puede como consecuencia, enfrentar un proceso penal por delito de perjurio, tipificado en el artículo 318 de nuestro Código Penal. Lo anterior implica que el estudiante se vea forzado a realizar su mayor esfuerzo para que no sólo incluya información veraz en la Licencia de Publicación, sino que también realice diligentemente la gestión de subir el documento correcto en la plataforma digital Kerwá.

Capítulo 1

Introducción

En la actualidad, el cifrado de datos es de suma importancia. Mucho más allá del uso en ambientes tradicionales que requieren de un alto grado de confidencialidad, tales como los militares o financieros, hoy es una herramienta utilizada por el público general para llevar a cabo la gran mayoría de las transacciones electrónicas vía web, usualmente sin tan siquiera enterarse de que los datos han sido cifrados y descifrados varias veces por transacción [1].

En el desarrollo de la computación actual se ha tomado el sistema criptográfico RSA (por los apellidos de sus proponentes Rivest, Shamir y Adleman) y sus variantes como base para realizar el intercambio de información de forma segura [2]. Su seguridad reside en que la factorización de números grandes en números primos es ineficiente para las arquitecturas computacionales actuales. De hecho, se usan llaves tan grandes que hacen que a estas arquitecturas tradicionales les tome cientos o miles de años lograr la factorización. No obstante en 1994, Peter Shor creó un algoritmo diseñado para un computador cuántico que en un tiempo polinomial es capaz de factorizar números de

cualquier tamaño en números primos, lo cual hace práctico un ataque contra RSA, en cuanto se disponga de un computador cuántico [3].

El algoritmo de Shor consiste de dos partes: la primera, una reducción del problema de descomponer en factores primos al problema de encontrar el orden, que se puede hacer en una computadora clásica y la segunda, un algoritmo cuántico para solucionar el problema de encontrar el período, donde ahora en el computador cuántico los bits son ondas, de todos los posibles períodos, por lo que se puede, en un tiempo muy corto, probar todos los casos de una sola vez.

Si bien el ataque a RSA mediante el algoritmo de Shor había sido una curiosidad hasta hoy en día, la creciente disponibilidad de máquinas cuánticas reales hace que la utilidad de RSA, y por ende de todos los algoritmos asimétricos, se vea comprometida [4].

Hoy en día, empiezan a existir desarrollos importantes en el mundo de la computación cuántica, como el desarrollo de los primeros circuitos cuánticos, que crean una necesidad del desarrollo de mejores sistemas de cifrado resistentes a este ataque cuántico [5]. Por computación cuántica se entiende un nuevo paradigma computacional distinto al tradicional. En este nuevo paradigma se tienen nuevas compuertas lógicas y bits cuánticos que almacenan más estados que su contraparte clásica [6].

La primera idea que se le ocurre a muchos al ver comprometido RSA por el ataque basado en el algoritmo de Shor es desarrollar un cifrado cuántico, donde todas las propuestas a la fecha están basadas en el concepto de enredo cuántico, que es un

fenómeno donde dos paquetes de información se encuentran entrelazados sin importar cuán separados estén [7]. Aunque, los cifrados cuánticos en principio son una buena idea, queda a la merced del precio de los computadores cuánticos, que al principio no serán fácilmente accesibles, y los temas de logística de la migración de la información del computador clásico o tradicional al nuevo, que será un problema tan grande como el desarrollo de la nueva arquitectura, por lo que tener un cifrado clásico (no cuántico, es decir que puede corre en el computador tradicional) resistente al ataque cuántico sería más conveniente [8].

Para la discusión a seguir es importante definir el concepto de *sistema*. Cuando se habla de sistema dentro de las ciencias exactas, se entiende como un fenómeno de la naturaleza que se caracteriza en su totalidad por un conjunto único de ecuaciones matemáticas. Por mucho tiempo las ecuaciones de Newton fueron suficientes para describir la mayoría de sistemas de la naturaleza, por lo que a estos sistemas se les llamó *clásicos* [9]. A inicios del siglo XX se empezaron a encontrar sistemas que no se podían describir con el trabajo de Newton. Dos de los casos más conocidos son los sistemas cuánticos, que usan las ecuaciones de la mecánica cuántica, y los sistemas caóticos, que usan versiones no lineales de las ecuaciones de Newton.

Desde el descubrimiento de los sistemas caóticos, Claude Shanon, padre de la teoría de la información, resaltó su relación estrecha con los requerimientos deseados para un buen sistema criptográfico [10]. Aunque el desarrollo de algoritmos de cifrado de llave pública siempre ha sido complicado por el paralelismo entre eficiencia y seguridad [11], actualmente juegan un papel importante por la estrecha relación que existen entre el caos y el enredo cuántico [12], dándole a este tipo de cifrado una fortaleza ante el al-

goritmo de Shor.

En esta tesis aprovecha el conocimiento de sistemas caóticos que presentan características similares a los efectos conocidos resistentes al ataque descrito para RSA y se muestra un sistema computacionalmente eficiente para lograr criptografía clásica segura a los ataques cuánticos. Esto es un tema de mucho interés y desarrollo, pues la seguridad de los datos es fundamental en nuestra sociedad [13].

El objetivo general de esta tesis es construir un esquema de cifrado no lineal que reproduce de manera clásica una propiedad similar a las presentes en el enredo cuántico, para la búsqueda de un esquema clásico resistente a ataques cuánticos.

Los objetivos específicos para lograr el objetivo general de esta investigación son los siguientes:

1. Caracterizar un sistema clásico que simule al menos una propiedad del enredo cuántico.

2. Elaborar un algoritmo de cifrado que utilice el sistema clásico caracterizado.

3. Someter el algoritmo propuesto al menos a un ataque conocido para la familia de cifrados a la que pertenezca con el fin de determinar su resistencia.

4. Evaluar la eficiencia de cifrado y descifrado del algoritmo.

1.1. Antecedentes y trabajo relacionado

En esta sección se presentan los antecedentes de la temática. En la sección 1.1.1 se describen los algoritmos de cifrado. En la sección 1.1.2 se revisan los esquemas de cifrado que usan sistemas complejos y en la sección 1.1.3 se profundiza en los elementos que hacen viables a estos cifrados. En la sección 1.1.4 se describen los antecedentes en la temática de los algoritmos resistentes a los ataques cuánticos conocidos, destacando en la sección 1.1.5 el único caso para cifrados de llave pública.

1.1.1. Algoritmos de cifrado

Aunque existen algoritmos de cifrado desde hace al menos cinco mil años, con el advenimiento de los computadores han surgido algoritmos más complejos y con contraseñas (llaves) mucho más grandes [1]. Sin embargo, hasta la década de 1970, todos los algoritmos requerían que la persona que cifraba y la que descifraba conocieran la contraseña completa. Aunque a pequeña escala esto no es problemático, sí implica dificultades logísticas significativas cuando se trata de distribuir una contraseña de forma segura a múltiples entidades o lidiar con la expiración de vigencia de las contraseñas de forma segura y eficiente. Estos algoritmos, en los que las entidades que comparten el secreto deben conocer de forma completa las llaves de cifrado, son denominados algoritmos de llave privada y de ellos existen múltiples modelos [1].

Existe otra categoría de algoritmos de cifrado que no requiere que los participantes conozcan la llave de antemano, sino que la entidad que cifra lo puede realizar con una llave parcial, denominada *llave pública*, y la única entidad que posee la llave completa es la que descifra. La porción que debe estar segura es denominada *llave privada* y el

único que requiere conocerla es el que debe leer el mensaje cifrado. Las matemáticas necesarias para crear este esquema fueron estudiadas en la década de 1960 en Gran Bretaña, pero no fueron publicadas por ser consideradas confidenciales y para uso exclusivamente [14]. Fue un equipo independiente formado por Ron Rivest, Adi Shamir, y Leonard Adleman, quienes desarrollaron en 1978 un algoritmo de cifrado de llave asimétrica, denominado RSA (por sus apellidos) [14]. Los esquemas de este tipo se denominan asimétricos, aunque en la actualidad existen más algoritmos, son todos equivalentes a RSA [15].

En general los algoritmos de llave pública funcionan bajo este esquema. Supongamos que Bob quiere enviar a Alicia un mensaje secreto que solo ella pueda leer. Alicia envía a Bob una caja con un candado abierto, del que solo Alicia tiene la llave. Bob recibe la caja, escribe el mensaje, lo pone en la caja y la cierra con el candado (ahora Bob no puede leer el mensaje). Bob envía la caja a Alicia y ella la abre con la llave. En este ejemplo, la caja con el candado es la llave pública de Alicia y la llave del candado es su llave privada.

En el caso de RSA, las llaves son dos cadenas de números muy grandes, donde la llave privada es un número primo y la pública es otro número primo grande multiplicado por el de la llave privada. Bob debe multiplicar su mensaje por la llave pública y enviarlo a Alicia. Como Alicia conoce las dos llaves puede factorizar el mensaje de Bob. A cualquier otra persona que intercepte el mensaje le tomaría cientos o miles de años en hacer la factorización [15].

Una de las razones por las que son importantes los algoritmos asimétricos es que

mejoraron la seguridad de las transacciones web. Por ejemplo, cuando uno se conecta a un sitio HTTPS, el navegador solicita la porción pública de la llave al sitio, luego cifra los datos, y por último los envía al servidor web. Una vez en el servidor, los datos son descifrados por medio de su llave privada. Así, el cliente y el servidor logran intercambiar los datos de forma segura, sin tener un intercambio de una llave simétrica de por medio [16]. Esto resuelve la dificultad de los esquemas simétricos, donde el intercambio de la llave simétrica solo es seguro si se hace de forma presencial, pues cifrarla lleva al problema original, en el que para cifrar es necesario compartir otra llave de forma presencial.

Si bien los algoritmos asimétricos permiten un contacto inicial seguro entre entidades que no se conocen mutuamente, su complejidad computacional y la generación de nuevas llaves no son triviales. Es por ello que típicamente se utilizan únicamente para cifrar el intercambio de llaves simétricas entre las entidades, pero los datos en sí son cifrados y descifrados utilizando alguno de los múltiples algoritmos de llave simétrica, que continúan siendo muy usados [1].

Tanto los algoritmos simétricos como los asimétricos poseen vulnerabilidades que potencialmente le permitirían a un atacante descifrar mensajes a los cuales no tiene derecho. No todos los ataques sobre estas vulnerabilidades tienen utilidad práctica, ya sea por el tiempo de ejecución o por su complejidad. Sin embargo, si se descubre un ataque nuevo que se puede ejecutar facilmente, el algoritmo de cifrado pierde utilidad.

1.1.2. Cifrado por sistemas dinámicos

Para los tres objetos criptográficos más comunes, los algoritmos de cifrado de bloque (un tipo de algoritmos de llave privada), los generadores de números seudoaleatorios (cifras de la corriente de aditivos) y los algoritmos de llave pública, existen implementaciones que usan sistemas dinámicos en estados caóticos [17].

Los sistemas de cifrado de bloque transforman una cadena relativamente corta (normalmente de 64, 128 o 256 bits) en una cadena de la misma longitud bajo el control de una llave secreta. Se han propuesto varios de estos sistemas de cifrado de bloque basados en mapas caóticos, en los que una discretización (un proceso que describe la forma en que un mapa caótico se implementa en la computadora) no se realiza mediante el redondeo del mapa caótico de acuerdo con la aritmética computacional (caso inseguro conocido como un problema de seudocaos) sino que más bien se construye explícitamente dentro del algoritmo [18].

Pichler y Scharinger proponen sistemas criptográficos basados en permutaciones caóticas construidos al discretizar explícitamente el mapa del panadero de dos dimensiones, idea que resuelve la problemática de seudocaos [19]. Fridrich extendió estas ideas para permutaciones caóticas en cualquier tamaño de ventanas bidimensionales [20].

Un generador de números seudoaleatorio es un método determinista. Por lo general se describe como un mapeo, que partiendo de un pequeño conjunto de números al azar llamado la semilla, produce un conjunto más grande de números de aspecto aleatorio, llamados números seudoaleatorios. Los sistemas caóticos se pueden utilizar para gene-

rar números seudoaleatorios. Por ejemplo, en una serie de documentos [21], se propuso un generador de números seudoaleatorios derivado del caos.

Los algoritmos de llave pública [22], también llamado algoritmos asimétricos, tienen las siguientes características:

1. La llave de cifrado es diferente de la llave de descifrado.
2. La llave de cifrado se puede publicar.
3. La llave de descifrado no puede ser calculada a partir de la llave de cifrado en un tiempo razonable.

Existen muchos algoritmos de llave pública y los tres más utilizados son RSA, ElGamal y Rabin [22]. Relacionados con estos están los algoritmos de cifrado de llave pública que usan mapas de Chebyshev [17, 23]. Estos están definidos en el conjunto de los números entre -1 y 1, y usan la aritmética de punto flotante. Es relevante enfatizar que el uso de estos mapas no es seguro [17].

Existe un tipo de híbrido de algoritmos de llave pública similares a ElGamal y RSA, en los que se usan los mapas de Chebyshev. Este tipo de criptografía caótica es segura y se puede utilizar para el cifrado y firma digital [23].

A pesar de los trabajos descritos en este capítulo para el campo de la criptografía basada en el caos, el impacto de estos en las investigaciones en la criptografía convencional es marginal. Esto se debe a varias razones [24]:

1. Casi todos los algoritmos criptográficos basados en el caos utilizan sistemas dinámicos definidos en el conjunto de los números reales, por lo que su realización práctica es difícil, tanto en algoritmos que como en circuitos eléctricos.

2. La seguridad y el rendimiento de casi todos los métodos basados en el caos no han sido analizados en términos de las técnicas desarrolladas en la criptografía usual.

3. La mayor parte de los métodos propuestos generan algoritmos criptográficamente débiles y los algoritmos mejorados son relativamente lentos.

Eso puede parecer poco esperanzador, el estudio de algoritmos de cifrado caóticos no se ha abandonado por la similitud del caos con la criptografía y por representar un camino a seguir en búsqueda de un algoritmo resistente a los ataques cuánticos.

1.1.3. Algoritmos criptográficos y su viabilidad

Existen muchos factores que deben tenerse en cuenta al examinar la viabilidad de un algoritmo criptográfico de llave pública. Entre los más importantes se encuentran la longitud de las llaves públicas, los mensajes de intercambio de las llaves y las firmas. Para los algoritmos criptográficos de llave pública más comunes, como RSA, estos tamaños son todos similares, y van de unos pocos cientos de bits a unos pocos miles, dependiendo del algoritmo, esto no sucede en el caso de los algoritmos candidatos a ser cuántico resistentes. Si las llaves públicas, los mensajes de intercambio de las llaves o las firmas son mayores que unos pocos miles de bits, se crea un problema para los dispositivos utilizados, que son de memoria o ancho de banda limitado [25].

Otro aspecto a considerar es la duración de la llave privada. Una transcripción de los mensajes firmados, a menudo, revela información acerca de la llave privada del firmante. Esto limita el número de mensajes que, con seguridad, se pueden firmar con la misma llave. El ejemplo más extremo de esto es el esquema de firmas de Lamport, que requiere una nueva llave para cada mensaje firmado. Se han desarrollado métodos para la creación de un esquema de firma a largo plazo, pero estos a menudo requieren memoria adicional para la gestión y el almacenamiento de llaves temporales, que tienden a aumentar la longitud efectiva de las firmas.

La llave privada utilizada para el descifrado, por lo general, tienen vida ilimitada, ya que no se utiliza en el cifrado y, por tanto, no se puede violar. Además, los protocolos de comunicación casi siempre pueden ser diseñados para evitar que el descifrador revele información sobre la llave privada. Esto se puede hacer mediante el cifrado de llaves simétricas y no mediante el contenido en sí, el uso de protección de la integridad o la presentación de informes de fracasos de descifrado de una manera que los hace indistinguibles de los códigos de autenticación de mensajes. Este tipo de comportamiento es usado para protocolos seguros con viejos esquemas de relleno RSA, y, a menudo, se considera una buena práctica, independientemente del mecanismo de transferencia de la llave [5].

Por último, se analiza el costo computacional. Hay cuatro operaciones básicas de llave pública: el cifrado, el descifrado, la firma y la verificación de firma. Los algoritmos utilizados para estas operaciones suelen tomar unos pocos milisegundos, excepto para el cifrado RSA y la verificación de firmas, que puede ser aproximadamente 100 veces más rápido debido a la utilización de llaves públicas pequeñas. El tiempo para generar

las llaves también puede ser una preocupación, si es significativamente más largo que las operaciones criptográficas básicas. Esquemas basados en factorización, como RSA o Rabin y Williams, presentan este problema, ya que la generación de factores primos de alta entropía requiere un cálculo que dura varios segundos [26].

1.1.4. Algoritmos resistentes a ataques cuánticos

Los algoritmos resistentes a los ataques descritos en la sección 2.2, pueden tener problemas bajo algunas de las consideraciones descritas en la sección anterior. Entre los algoritmos resistentes a estos ataques se conocen tres familias: las firmas de Lamport, las llaves de largo término para esquemas de una firma y la criptografía basada en redes [26]. De estos, solo el último es relevante para el cifrado de llave pública, tema de esta tesis.

1.1.5. Criptografía basada en redes

Una red de n dimensiones es un conjunto de vectores que se pueden expresar como la suma de múltiplos enteros de un conjunto específico de n vectores, llamado la *base* de la red. Existe una infinita cantidad de bases distintas que generaran la misma red. Dos problemas NP-completos (los problemas NP son el conjunto de problemas que pueden ser resueltos en tiempo polinómico por una máquina de Turing no determinista) relacionados con el concepto de red son el problema del vector más corto (SVP, por sus siglas en inglés) y el problema del vector más cercano (CVP, por sus siglas en inglés) [27].

Dada una base arbitraria para una red, pedir encontrar el vector más corto en la red (SVP) o encontrar el vector de la red más cercano (CVP) a un vector de retícula arbitrario, tanto en la mecánica cuántica y como en los modelos de cálculos clásicos, son problemas difícil de resolver para redes de alta dimensionalidad, ya que se tiene un gran número de posibles vectores cercanos al vector de la red más corto [27].

1.2. Estructura del documento

La tesis se estructura en 7 capítulos. El capítulo 1 introduce el tema y explica su relevancia. El capítulo 2 presenta el marco teórico, el cual detalla los conceptos importantes de poco uso de uso poco común en la comunidad académica de interés para entender el desarrollo de este trabajo. En el capítulo 3 describe la metodología a seguir para completar los objetivos que persigue el trabajo. El capítulo 4 describe el esquema de cifrado propuesto. El capítulo 5 se da el criptoanálisis. El capítulo 6 expone un escenario de comunicación que usa el esquema de cifrado propuesto en este trabajo. Para finalizar, el capítulo 7, presenta las conclusiones.

Capítulo 2

Marco Teórico

Este capítulo sienta las bases conceptuales necesarias para entender el contexto y los fundamentos en los que se enmarca la investigación. Primeramente, se trata el tema del algoritmo de cifrado RSA con el afán de profundizar en su realización matemática. Como ya fue descrito, el ataque conocido para RSA es el algoritmo de Shor, por lo que se dedica la sección 2.3 para explicarlo, debido a que este necesita de una computadora cuántica para su ejecución, en la sección 2.4 se conceptualiza lo que representa. La defensa al algoritmo de Shor está basada en el fenómeno de enredo cuántico, la sección 2.5 describe tal defensa. En la sección 2.7 se estudian los cifrados basados en sistemas complejos y sus ventajas. La sección 2.8 se presenta los mapas caóticos acoplados, que sabemos son similares clásicos al enredo cuántico. Para terminar, la sección 2.9 explica la encriptación de dinámica distribuida, que es el único esquema de llave pública caótico resistente a ataques cuánticos.

2.1. Algoritmo RSA

Los algoritmos de llave pública funcionan bajo el siguiente esquema en general. Bob quiere enviar a Alicia un mensaje ultra secreto que solo ella debe leer. Alicia envía a Bob una caja con un candado abierto (llave pública), del que solo Alicia tiene la llave (llave privada). Bob escribe el mensaje, lo pone en la caja que recibió de Alicia y la cierra con el candado (ahora nadie puede leer el mensaje). Bob envía la caja a Alicia y ella la abre con su llave privada.

El algoritmo RSA se basa en la multiplicación de dos funciones exponenciales con sus argumentos como llaves. Se usan dos llaves que suelen ser números muy grandes, la llave privada es un número primo y la pública cualquier otro número multiplicado por la privada. Bob debe multiplicar su mensaje por la llave pública y enviarlo a Alicia. Ella conoce las dos llaves con las que puede factorizar el mensaje de Bob. A cualquier otra persona que intercepte el mensaje le tomaría cientos o miles de años hacer la factorización [15].

La idea general de RSA es detallada en la sección 2.1.1. El algoritmo consta de tres pasos: generación de llaves, cifrado y descifrado, que se explican en las secciones 2.1.2, 2.1.3 y 2.1.3.

2.1.1. Idea del algoritmo

Bob quiere enviar a Alicia un mensaje secreto que solo ella pueda leer. Al mensaje original o *mensaje plano* (es decir, sin cifrar), lo llamaremos M. Bob lo envía en forma de un número m menor que otro número n, mediante un protocolo reversible conocido

como patrón de relleno (*padding scheme*, en inglés). A continuación genera el *mensaje cifrado* c mediante la siguiente operación:

$$c = m^e \pmod{n} , \tag{2.1}$$

donde e es la llave pública de Alicia. Para descifra el mensaje codificado en c, lo hace mediante la operación inversa dada por

$$m = c^d \pmod{n} , \tag{2.2}$$

donde d es la llave privada que solo Alicia conoce.

2.1.2. Generación de llaves

Para generar las llaves, se eligen dos números primos distintos p y q. Por motivos de seguridad, estos números deben escogerse de forma aleatoria y deben tener una longitud en bits parecida. Se pueden hallar números primos fácilmente mediante un test de primalidad como el de Miller-Rabin [28]. Luego, se calcula

$$n = p \cdot q , \tag{2.3}$$

y n se usa como el módulo para ambas llaves, pública y privada. A continuación se calcula

$$\varphi(n) = (p-1) \cdot (q-1), \tag{2.4}$$

donde φ es la función φ de Euler [29], que se calcula aprovechando dos de sus propiedades: si p es primo, entonces

$$\varphi(p) = p - 1, \tag{2.5}$$

y si m y n son primos entre sí, entonces

$$\varphi(mn) = \varphi(m)\varphi(n). \tag{2.6}$$

Se escoge un entero positivo e menor que $\varphi(n)$, que sea coprimo con $\varphi(n)$. A e se le conoce como el exponente de la llave pública. Si se escoge un e con una suma encadenada corta, el cifrado será más efectivo. Un exponente e muy pequeño (por ejemplo $e \leq 7$) podría suponer un riesgo para la seguridad. Se determina un d (mediante aritmética modular) que satisfaga la congruencia:

$$e \cdot d \equiv 1 \pmod{\varphi(n)}, \tag{2.7}$$

es decir, que d sea el multiplicador modular inverso de $e \pmod{\varphi(n)}$. Expresado de otra manera,

$$d \cdot e - 1 \tag{2.8}$$

es dividido exactamente por la ecuación (2.4). Esto suele calcularse mediante el algoritmo de Euclides extendido [30]. Ahora d se guarda como el exponente de la llave privada.

Con los resultado del párrafo anterior tenemos que la llave pública es (n, e), esto es, el módulo y el exponente de cifrado. La llave privada es (n, d), esto es, el módulo y el exponente de descifrado, que debe mantenerse en secreto.

2.1.3. Cifrado

Para cifrar los datos, Alicia comunica su llave pública (n, e) a Bob y guarda la llave privada en secreto. Ahora Bob desea enviar un mensaje M a Alicia. Primero, Bob convierte M en un número entero $m < n$ mediante un protocolo reversible acordado de antemano y que garantiza que m y n son coprimos (en caso contrario, no se puede aplicar el teorema de Euler en el descifrado y, por tanto, no se tendría la seguridad de recuperar el mensaje original a partir del mensaje cifrado). Luego, calcula el texto cifrado c mediante la operación

$$c \equiv m^e \pmod{n}. \tag{2.9}$$

Esto puede hacerse rápidamente mediante el método de exponenciación binaria. Ahora Bob transmite c a Alicia.

2.1.4. Descifrado

Alicia puede recuperar m a partir de c usando su exponente d de la llave privada mediante el siguiente cálculo:

$$m \equiv c^d \pmod{n}\ . \tag{2.10}$$

Ahora que tiene m en su poder, puede recuperar el mensaje original M invirtiendo el esquema de relleno. El procedimiento anterior funciona porque

$$c^d = (m^e)^d \equiv m^{ed} \pmod{n} \tag{2.11}$$

y como hemos elegido d y e de forma que $ed = 1 + k\varphi(n)$, se cumple que

$$m^{ed} = m^{1+k\varphi(n)} = m(m^{\varphi(n)})^k \equiv m \pmod{n}. \tag{2.12}$$

La última congruencia se sigue directamente del teorema de Euler cuando m es coprimo con n. Puede demostrarse que las ecuaciones se cumplen para todo m usando congruencias y el teorema chino del resto [31]. Esto demuestra cómo se obtiene el mensaje original, sustituyendo la ecuación (2.11) en la ecuación (2.12), encontrando que

$$m = c^d \pmod{n}\ . \tag{2.13}$$

La sección 2.1 muestra la seguridad y simplicidad de RSA. Sin embargo, el panorama ha venido cambiando desde el año 1994, cuando aparece el renombrado algoritmo de Shor [16], que es capaz de factorizar números grandes en tiempos polinomiales, usando una arquitectura especial conocida como arquitectura cuántica, que aprovecha fenómenos únicos del mundo cuántico (de átomos y electrones) para ejecutar algoritmos [32], aunque en el momento de su aparición sus consecuencias no eran apreciables debido a que esta arquitectura daba en sus primeros pasos. En la siguiente sección profundizamos en dicho algoritmo.

2.2. Ataques cuánticos

La seguridad de los sistemas criptográficos de llave pública más usados RSA (Rivest-Shamir-Adleman), DSA (algoritmo de firma digital) y ECC (criptografía de curva elíptica) se basa en la dificultad de problemas relevantes de la teoría de números. El problema de factorización de enteros (IFP) da la seguridad de RSA, el problema

del logaritmo discreto (DLP) da la de DSA y el problema del logaritmo discreto de la curva elíptica (ECDLP) da la de ECC [25]. Dado que hasta ahora no se han encontrado algoritmos de tiempo polinómico para resolver estos tres problemas, los sistemas criptográficos basados en ellos son seguros. Sin embargo, los algoritmos cuánticos, debido a Shor y otros, resuelven estos tres problemas difíciles para los computadores clásicos en tiempos polinómicos, siempre y cuando un computador cuántico práctico pueda ser construido [5].

2.3. Algoritmo de Shor

El algoritmo de Shor busca, para un número entero N, encontrar otro número entero p entre 1 y N que divida N. El algoritmo consiste de dos partes: una reducción del problema de descomponer en factores al problema de encontrar el período (el entero positivo r más pequeño tal que para una función f se obtenga que $f(x) = f(x + r)$). Este algoritmo puede ejecutarse en una computadora clásica y un algoritmo cuántico para solucionar el problema de encontrar el período.

2.3.1. Parte clásica

La parte clásica del algoritmo de Shor busca descomponer en factores el problema de encontrar el período, para lo cual se tienen los siguientes pasos:

1. Se escoge un número aleatorio $a < N$.

2. Se calcula el mcd(a, N). Esto se puede hacer usando el algoritmo de Euclides.

3. Si el mcd(a, N) $\neq 1$, entonces es un factor no trivial de N, así que terminamos.

4. Si no, encuentre el período r, usando el algoritmo en la sección 2.3.2, de la siguiente función:

$$f(x) = a^x \bmod N,$$

es decir, el número entero más pequeño r para el cual $f(x+r) = f(x)$.

5. Si r es impar, vaya al paso 1.

6. Si $a^{r/2} = -1$ (mod N), vaya al paso 1.

7. Los factores de N son el mcd$(a^{r/2} \pm 1,\ N)$. Terminamos.

2.3.2. Parte cuántica

Este algoritmo busca encontrar el período. Se le conoce como subprograma cuántico debido a que complementa al algoritmo de la sección 2.3.1 en el paso 4. Este subprograma tiene los siguentes pasos:

1. Comience con un par de registros qubits de entrada y salida con $\log_2 N$ qubits cada uno, e inicialícelos en

$$N^{-1/2} \sum_x |x\rangle \, |0\rangle \,, \tag{2.14}$$

donde x va de 0 a $N-1$.

2. Construya $f(x)$ como función cuántica y aplíquela al estado antedicho, para obtener

$$N^{-1/2} \sum_x |x\rangle \, |f(x)\rangle \,. \tag{2.15}$$

3. Aplique la transformada cuántica de Fourier al registro de entrada. La transformada cuántica de Fourier en N puntos se define como

$$U_{QFT} \left|x\right\rangle = N^{-1/2} \sum_{y} e^{2\pi i x y/N} \left|y\right\rangle , \tag{2.16}$$

lo que nos deja en el estado siguiente:

$$N^{-1} \sum_{x} \sum_{y} e^{2\pi i x y/N} \left|y\right\rangle \left|f(x)\right\rangle . \tag{2.17}$$

4. Realice una medición. Obtenga un cierto resultado y en el registro de entrada y $f(x_0)$ en el registro de salida. Aunque este paso es innecesario, ya que, de acuerdo con el principio de medición en diferido, el resultado es el mismo al final del algoritmo, independientemente de que se realice una medición, se incluye por razones de simplificación a la hora de entender el algoritmo. Puesto que f es periódica, la probabilidad de medir cierto valor de y viene dada por

$$N^{-1} \left| \sum_{x:\, f(x)=f(x_0)} e^{2\pi i x y/N} \right|^2 = N^{-1} \left| \sum_{b} e^{2\pi i (x_0 + rb) y/N} \right|^2 . \tag{2.18}$$

Un análisis de la ecuación muestra que cuanto más alta es esta probabilidad, tanto más el factor yr/N es cercano a un número entero.

5. Convierta y/N en una fracción irreducible y extraiga el denominador r', que es un candidato a r.

6. Compruebe si $f(x) = f(x + r')$. Si es así termine.

7. Sino, obtenga más candidatos para r usando valores cercanos a y, o múltiplos de

r'. Si cualquier candidato cumple las condiciones, termine.

8. Sino, repita este subprograma.

Como se mencionó, es necesario un computador cuántico para ejecutar algoritmo, por lo que en la siguiente sección introducimos qué se conoce sobre dicha arquitectura.

2.4. Computación cuántica

Un computador cuántico aprovecha el fenómeno cuántico para poder realizar operaciones con datos más rápidas y eficientes que un computador clásico. Este computador se caracteriza por que las dimensiones de sus componentes se encuentran en escalas que se salen de la mecánica de Newton, que son regidas por la mecánica cuántica. Cabe aclarar que un computador cuántico no es simplemente una versión más poderosa de un computador clásico, sino que posee características sin contraparte en la computación tradicional. Por ejemplo, en 1936, Garrett Birkhoff y John von Neumann, estudiando el artículo clásico de Alan Turing [33], en un intento fallido de probar que la máquina de Turing funciona para el caso de la mecánica cuántica, encontraron que el álgebra booleana no se puede aplicar en condiciones cuánticas [34].

Una particularidad importante es el comportamiento de los bits en un computador cuántico. En un arreglo de bits clásicos (registro) en un momento determinado se puede almacenar un único estado. En un registro de 3 bits binarios, en el momento t se puede almacenar únicamente un número, por ejemplo, el número dos (010). En el caso de que los bits fueran hexadecimales en tres bits podría contenerse nuevamente un solo

número: el número dos en hexadecimal (002) no gana capacidad al variar el comportamiento del bit. Al igual que en el caso clásico, los bits cuánticos pueden estar en cualquier base. La diferencia es que en tres bits cuánticos binarios todos los estados de interés almacenables en tres bits binarios pueden guardarse simultáneamente en el mismo registro (por ejemplo, en el tiempo t, podrían estar almacenados 010, 011 y 100). En el hipotético registro de 3 bits hexadecimales podrían estar almacenados simultáneamente los 4096 estados que puede tomar el registro de 3 bits hexadecimales. Un hecho que no se puede lograr con los bits clásicos es que cada bit cuántico puede representar un número complejo. Debido a esto y otros aspectos, en un computador cuántico se ve comprometida el álgebra booleana, que es elevante para las operaciones de los algoritmos clásicos.

Otro aspecto notable es el problema de la interfaz. El tiempo que toma extraer la información de estos registros se puede hacer extremadamente grande (en el orden de años) debido al principio de incertidumbre. Por ejemplo, bajo ciertas circunstancias, se deben aplicar energías bajas (cercanas al cero absoluto) por tiempos largos para no destruir el estado a recuperar. Esto no es un problema de la implementación, sino de los principios que rigen el mundo cuántico. Tal como las dos anteriores, es posible encontrar muchas otras divergencias de la computación cuántica con respecto a la clásica, las que son un problema por resolver.

Como se ha mencionado en el inicio de esta sección, el concepto de computadores cuánticos ha coexistido con los inicios de los paradigmas de computación clásica, pero fue hasta 1985 que se dio la primera implementación simulada [35], cuando se dio inicio a una gama de otras simulaciones y algunas realizaciones no simuladas. En 2012,

apareció en el mercado el primer computador cuántico comercial [4], aunque en 2013 que se probó que dicho computador era realmente cuántico [36] . Los avances recientes siguen en la misma línea: se pone énfasis en máquinas de propósito que puedan ejecutar exclusivamente el algoritmo de Shor [37], aunque existen detalles que no logran convencer a la comunidad, pero es una realidad que la computación cuántica es cada vez más cercana [37]. Todo esto hace imperante el desarrollo de sistemas de cifrados y los sistemas dinámicos han sido una esperanza.

Un fenómeno físico usado para crear cifrados resistentes al ataque cuántico es la propiedad de enredo cuántico, la cual describimos en la siguiente sección.

2.5. Enredo cuántico

El enredo cuántico es una propiedad de los sistemas cuánticos predicha en 1935 por Einstein, Podolsky y Rosen [38]. Fue verificada experimentalmente en la década de 1980, constatando que dicha propiedad no se puede conciliar con la constancia de la velocidad de la luz [39]. El fenómeno del enredo cuántico no tiene equivalente clásico, en el cual los estados cuánticos de dos o más objetos se describen mediante un estado único que involucra a todos los objetos del sistema, aun cuando los objetos estén separados espacialmente. Esto lleva a correlaciones entre las propiedades físicas observables. Por ejemplo, es posible preparar (enlazar) dos partículas en un solo estado cuántico de espín nulo, de forma que cuando se observe que una gira hacia arriba, la otra automáticamente recibe una señal y se muestra girando hacia abajo, pese a la imposibilidad de predecir, según los postulados de la mecánica cuántica, qué estado se observará [39].

El principio de exclusión de Pauli asegura que dos electrones juntos en el mismo espacio no pueden tener el mismo espín [40]. Como se ve en la figura 2.1, los electrones deben tener espines opuestos para cumplir el principio anteriormente descrito. Aunque los electrones sean separados una distancia infinita dicho principio se mantiene, por lo que, si cambiamos uno, el otro debe cambiar también. La entropía sirve como una medida del enredo cuántico [41].

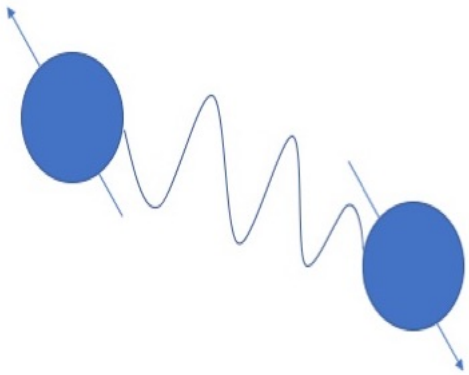


Figura 2.1: Un ejemplo del enredo cuántico usual es la separación de dos electrones que estuvieron previamente en un mismo punto

Un hecho relevante para este trabajo es que todas las versiones de criptografía cuántica resistentes a los ataques cuánticos se basan en el enredo cuántico. Aunque este no tiene contraparte clásica, las redes de mapas acoplados (RMA) representan una forma de correlación a distancia como la descrita. Esta característica compartida por el enredo cuántico y las RMA es la que se busca en el objetivo principal de esta tesis. Las RMA son una forma de describir un sistema dinámico, por lo que debemos aclarar este tipo de sistemas, tema que es tratado a continuación.

2.6. Sistemas dinámicos

Un sistema dinámico es aquel cuyo estado evoluciona en el tiempo. Hay dos tipos de sistemas dinámicos: lineales y no lineales. Son lineales aquellos cuyo comportamiento se puede expresar a partir de la suma de los comportamientos de sus partes. En los no lineales, en cambio, lo anterior no se cumple [42].

Dentro de los sistemas no lineales se encuentran categorizados los sistemas complejos, que son sistemas no lineales formados por muchas partes interconectadas entre sí. Los sistemas complejos exhiben tres propiedades claves [43]:

Sensibilidad a condiciones iniciales Se refiere a que muy pequeños cambio en las condiciones iniciales genera resultados muy distintos al caso donde no existen esos diminutos cambios.

Propiedades emergentes Son variables del sistema que no están presentes en las partes que conforman al sistema, el ejemplo clásico es el cerebro humano, ninguna de las neuronas tiene conciencia por si sola pero el cerebro si.

Ergodicidad Es la propiedad del sistema de ocupar todos los estados disponibles durante una larga evolución temporal.

Una propiedad que pueden presentar los sistemas complejos durante su evolución temporal es el caos, el cual puede estar o no presente. Se entiende como periodos caóticos a aquellos en los que se pierde la capacidad de predecir el siguiente estado del sistema. Gracias al trabajo de Lyapunov, es posible determinar cuándo el caos está presente en el sistema y cuál es su nivel de complejidad [43].

Un concepto de los sistemas dinámicos importante de aclarar es el de mapa, que es término matemático para resaltar alguna función que cumple propiedades muy especiales dentro un área de estudio, por ejemplo, en el caso del estudio de los sistemas dinámicos, se usa el mapa logístico, que es una función matemática sencilla que presenta caos [43].

2.7. Cifrado basado en sistemas complejos

Los cifrados basados en sistemas complejos son aquellos que aprovechan las propiedades claves de los sistemas caóticos para cifrar la información. Un cifrado basado en sistemas complejos, por su alta dependencia de las condiciones iniciales, hace imposible, a partir de los datos de la comunicación, conocer la descripción de su dinámica cuando se encuentra en estados caóticos, lo que puede representar una mejora sobre el esquema RSA. Los estados caóticos que pueden presentar los sistemas complejos tienen características relevantes como las siguientes: dependencia sensible a las condiciones iniciales, similitud con el comportamiento al azar y la continuidad del espectro del espacio de fase. El caos tiene aplicaciones potenciales en varios de los bloques funcionales de un sistema de comunicación digital como la compresión, el cifrado y la modulación, gracias a las similitudes que se pueden hacer con las características mencionadas.

En los primeros años del estudio de las similitudes entre el caos y la criptografía (entre 1992 y 1996), el principal objetivo fue desarrollar esquemas en los que un sistema caótico se utilizara tanto para modulación y codificación, de forma simultánea. Por lo complicado de este enfoque evolucionó en dos líneas de investigación: la modulación ba-

sada en caos [44] y la criptografía basada en sistemas complejos en estados caóticos [45].

La criptografía es reconocida como el mejor método de protección de datos contra ataques pasivos o activos y el algoritmo RSA como el esquema más usado. Una visión general de los acontecimientos recientes en el diseño de algoritmos criptográficos convencionales se da en el trabajo de Kocarev [46].

La estrecha relación entre el caos y la criptografía se discute en el libro de criptografía de Shannon [10]. En él, menciona que las buenas transformaciones de mezclas se dan a menudo por productos repetidos de dos operaciones sencillas no conmutativas. Tomando en cuenta que Heinz Hopf mostró, en su ejemplo de la masa de pastelería, que se puede mezclar una masa con una secuencia numerable de operaciones, la masa primero se extiende hacia fuera en una losa delgada, seguido, se doblada sobre si misma, se enrollada, se plega y luego se repite el proceso.

Una relación más detallada entre el caos y la criptografía es visible en el trabajo de Alvarez y Li [11]. En él se ve como la ergodicidad se compara con la confusión de la criptografía, la dinámica determinista con el hecho de que la aleatoriedad es realmente seudoaleatoriedad y la estructura compleja del caos con la complejidad de atacar la información cifrada. Sin embargo, hay otras que no se describe en ese artículo.

Una diferencia importante entre el caos y la criptografía es que las transformaciones de cifrado se definen en conjuntos finitos, mientras que el caos se definen en conjuntos infinitos (por ejemplo, los números reales), lo cual representa una dificultad en el desarrollo de sistemas de cifrados basados en el caos [46].

Entre los algoritmos caóticos de cifrado destacan los que usan mapas caóticos acoplados [20]. Ese es el tema de la siguiente sección.

2.8. Mapas caóticos acoplados

Las redes de mapas acoplados (RMA) son un tipo de sistema dinámico que se utiliza para modelar el comportamiento de sistemas no lineales. Una RMA consiste en un conjunto de elementos dinámicos, descritos por mapas, que interactúan o se acoplan con algunos otros elementos del conjunto. De esta manera, se tiene un sistema dinámico, discreto en el tiempo y en el espacio, con variables de estado continuas [47]. Estas redes están presentes en muchos sistemas de la naturaleza, siendo estos de gran complejidad matemática, aunque describan procesos simples con pocas variables en sus ecuaciones dinámicas [48]. Entre los sistemas que presentan estas redes se encuentran los siguientes:

Poblaciones Las migraciones o coexistencia de poblaciones de seres vivos se pueden ver como redes de mapas acoplados con cada mapa representando una población específica.

Reacciones químicas Cada ion se puede ver como un mapa que se acopla para lograr la reacción.

Convección Que está formada por la superposición de distintos vórtices.

Redes biológicas Las diferentes coexistencias entre especies como la simbiosis.

Oscilaciones Las presentes en distintos fluidos, desde estrellas en sus distintas fases

hasta microcristales en seres vivos.

Existe una similitud entre el comportamiento de estas redes y el enredo cuántico [20, 48]. Esto es relevante pues los cifrados realizados con RMA simulan algunos comportamientos del enredo cuántico.

2.9. Encriptación de dinámica distribuida.

Uno de los pocos algoritmos con algún grado de practicidad que se puede clasificar en la familia anteriormente descrita es el de encriptación de dinámica distribuida (EDD) [49]. Es un algoritmo teórico para el cifrado asimétrico que explota las propiedades de los sistemas dinámicos no lineales. Un sistema dinámico no lineal disipativo de alta dimensión se distribuye entre el transmisor y el receptor, por lo que se llama al método de cifrado distribuida dinámica. La dinámica del transmisor es pública y la del receptor está oculta, no compartida en el canal. Un mensaje se codifica mediante una modulación en los parámetros del transmisor, lo que se traduce en un desplazamiento del atractor global del sistema. Un receptor no autorizado no conoce las dinámicas ocultas en el receptor y no puede decodificar el mensaje [50].

La idea básica de la EDD es dividir un sistema dinámico de dimensión $D_T + D_R$ en dos partes con D_T variables de transmisor $t(n) = [t1(n); ...; t_{D_T}(n)]$, y las D_R variables del receptor de $r(n) = [r1(n); ...; r_{D_R}(n)]$. El receptor recibe la señal escalar $s_t(n)$ desde el transmisor, y el transmisor recibe la señal escalar $s_r(n)$ desde el receptor:

$$t(n+1) = F_T(t(n), s_r(n), m(n)), \tag{2.19}$$

y

$$r(n+1) = F_R(r(n), s_t(n), m(n)). \tag{2.20}$$

El mensaje $m(n)$ es el que se quiere cifrar. Permitimos que $m(n)$ tome valores de 0 o 1, esto crea un mensaje binario. Las llaves que se proponen son exactamente las dinámicas que se comparten.

Un receptor autorizado conoce todas las cantidades, públicas y privadas, pudiendo establecer antes de iniciar la comunicación los atractores admisibles, para todos los valores permitidos de $m(n)$, como se ve en la figura 2.2. Para el descifrado es necesario conocer previamente todos los puntos de los atractores. El proceso de descifrado corresponde al cálculo de la distancia de cada punto recibido por parte del transmisor hasta los puntos de los actractores, como se ve en la figura. Aunque un punto recibido no sea parte de un atractor, siempre con ese punto se puede iterar la dinámica, que es conocida únicamente por el receptor autorizado, para decidir a que atractor corresponde.

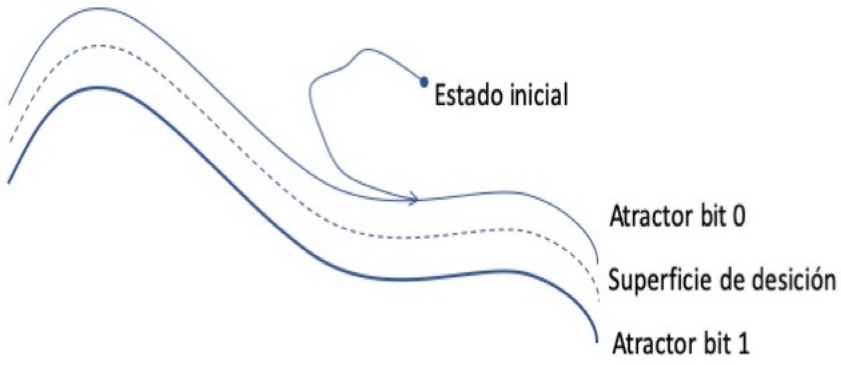


Figura 2.2: Idea detrás del algoritmo de descifrado de EDD

2.9.1. Desarrollos posteriores a EDD

Debido a que la practicidad de la EDD está cuestionada, se ha dado una nueva rama de trabajo [51]. Kocarev ha propuesto un cifrado de llave pública basado en mapas de Chebyshev que tiene un grado alto de originalidad y viabilidad, usando un solo mapa de Chebyshev en específico con su comportamiento particular [24]. En la más reciente revisión literaria sobre cifrados caóticos se menciona como el algoritmo presentado en los siguientes capítulos de este trabajo de tesis es el único desarrollo posterior a EDD que lo mejora y lo hace práctico para aplicaciones criptográficas [52]. Esto es posible debido a que los resultados de esta tesis ya fueron debidamente publicados [53].

Abarcado los conceptos teóricos relevantes para esta tesis, en el siguiente capítulo damos la estructura y métodos que atañen a la investigación realizada.

Capítulo 3

Metodología

En este capítulo se describe la metodología usada para conseguir los objetivos de este trabajo de tesis. El reporte del trabajo realizado se organiza por capítulos basados en su temática, que contienen tanto la metodología específica como los resultados, dada la naturaleza diferente de cada una de las actividades desarrolladas para cumplir dichos objetivos.

En una primera fase se seleccionó y se caracterizó un sistema de la naturaleza para el esquema de cifrado. Se usaron dos mapas: uno matemáticamente sencillo para minimizar la complejidad de la creación del nuevo algoritmo y el otro más complejo basado en la naturaleza, el cual representa una mejora con respecto al primero debido a que asegura, de forma independiente, los estados caóticos. En dicha fase se generó una ecuación diferencial que caracteriza a un sistema de la naturaleza. Un paso importante desarrollado para el mapa de la naturaleza es la comprobación de que los parámetros que describen al mapa sean realmente caóticos, por lo que dichos resultados están presentes en la sección 6.3. La caracterización comprueba que como resultado se obtiene

una red de mapa acoplado, según los detalles de la sección 2.8 del documento. Esta carecterización aparece en la sección 4.1.

La segunda fase fue plantear un nuevo esquema de cifrado que se pueda enmarcar dentro de los esquemas en la sección 2.9, con el afán de aprovechar las ventajas descritas. Aquí, fue necesario poder relacionar la dinámica de los mapas caracterizados con la dinámica propuesta en la EDD, dando como resultado una forma de hacer la distribución con el propósito de mejorar la creación de los algoritmos, que es la temática del capítulo 4 . Se hicieron, basados en los criterios criptográficos para los cifrados caóticos, los algoritmos de cifrado, descifrado y el ataque para este nuevo sistema criptográfico. Con el afán de optimizar el algoritmo de descifrado, se plantean tres modificaciones, donde se explora la computación en paralelo y una selección de los datos a descifrar valuada según la dinámica particular del descifrado de la sección 4.4.

La tercera parte consistió en elaborar el prototipo experimental, en cual se utilizó Python como lenguaje de desarrollo por las ventajas que representa en el manejo de datos de precisión y Kivy como motor gráfico debido a las capacidades multiplataforma de este. El prototipo fue capaz de cifrar, descifrar y atacar el texto cifrado con el principal de los ataques para esta familia de cifrados. Los sistemas caóticos utilizados para la creación de esquemas criptográficos tienen ataques conocidos según la familia del sistema utilizado. Basados en la familia a la que pertenece el cifrado desarrollado en esta investigación y con la ayuda de los diferentes formalismos planteados en la sección 2.7 se propuso un ataque específico para este cifrado [11]. Estos resultados son parte del capítulo 5.

La cuarta fase es experimental. En ella se planteó determinar la seguridad del cifrado. Esto se logró evaluando la resistencia del cifrado al ataque para la familia de cifrados a la que este pertenece. Además, para determinar la eficiencia del cifrado y descifrado, se midieron los tiempos de ejecución de los algoritmos para cadenas aleatorias de bits de distintas longitudes. Esto permitió comparar la eficiencia de las tres modificaciones planteadas al algoritmo de descifrado, tema de la sección 6.4.

El capítulo 4 describe la construcción del nuevo esquema de cifrado, con base en la metodología descrita en este capítulo.

Capítulo 4

Sistema criptográfico propuesto

En este capítulo se construye el nuevo sistema criptográfico, que consiste de tres partes: el armado, el cifrado y el descifrado. En la parte del armado, sección 4.1, se toma un mapa, que debe estar en estado caótico y se decide cómo distribuir la dinámica, con base en EDD. Para la facilidad matemática se usó el mapa logístico, que es el caso dinámicamente más sencillo que presenta caos.

Con la relación y la distribución de la dinámica armadas, se realiza el cifrado, en la sección 4.2. Esta es la primera vez que se usa este mapa para el caso de EDD. En la sección 4.3 se detalla el primer algoritmo computacional para el descifrado de EDD. Por último, en la sección 4.4 se dan tres variantes para buscar la optimización del descifrado.

4.1. Armado

Los sistemas caóticos tienen un gran potencial para cifrar información. Los sistemas criptográficos se clasifican en sistemas con llave privada y los sistemas con llave pública [54]. Se ha hecho un gran esfuerzo en crear sistemas con llave privada basados en caos, pero no con llave pública [55]. Uno de los sistemas de cifrado caóticos con llave pública más importante utilizan los mapas de Chebyshev para cifrar [56], pero su eficiencia es menor que la de RSA [57], un punto débil para los cifrados caóticos.

El mapa logístico es un excelente ejemplo de sistema caótico. Originalmente formulado para representar un modelo demográfico simple para explicar el aumento de una población, el mapa logístico es un mapa unimodal unidimensional y, como resultado, su dinámica es bastante limitada [58]. Se puede expresar mediante la ecuación

$$f(x) = \mu x(1 - x), \tag{4.1}$$

para $0 \leq \mu \leq 4$. El aspecto unimodal del mapa logístico lo hace inadecuado para aplicaciones criptográficas porque el parámetro se puede reconstruir a partir de las condiciones iniciales [59]. A pesar de dicha debilidad se ha creado un número razonable de propuestas [59]. Una nueva investigación mejora el mapa logístico para aplicaciones criptográficas, pero pierde la simplicidad matemática de la ecuación (4.1) [60].

Una red de mapa acoplado (RMA) es un sistema dinámico que modela el comportamiento de sistemas no lineales. Ella se utiliza predominantemente para estudiar cualitativamente la dinámica caótica de los sistemas espacialmente extendidos. Esto incluye la dinámica del caos espaciotemporal, en el que el número de grados efectivos

de libertad aumenta a medida que aumenta el tamaño del sistema. Una RMA incorpora un sistema de ecuaciones (acopladas o no acopladas), un número finito de variables, un esquema de acoplamiento global o local y los términos de acoplamiento correspondientes [61].

El mapa acoplado logístico es una de las RMA más simples y se basa en dos mapas logísticos acoplados mediante un acoplamiento lineal

$$x_{n+1} = f(x_n) + \alpha(y_n - x_n) \tag{4.2}$$

$$y_{n+1} = f(y_n) - \alpha(y_n - x_n), \tag{4.3}$$

con $f(x)$ como el mapa logístico de la ecuación (4.1), α es un parámetro de acoplamiento con dimensiones del sistema x y y. En el mapa logístico solo se observan dos rutas al caos (duplicación del período e intermitencia). La segunda dimensión del mapa logístico acoplado permite que ocurra la ruta cuasiperiódica [62]. El caso no simétrico del mapa acoplado logístico [63] ocurre cuando en las ecuaciones (4.2) y (4.3) se usa un parámetro μ_i diferente para cada $f_i(x)$, por lo que las ecuaciones toman la forma:

$$x_{n+1} = f_1(x_n) + \alpha(y_n - x_n) \tag{4.4}$$

$$y_{n+1} = f_2(y_n) - \alpha(y_n - x_n), \tag{4.5}$$

donde $f_1(x)$ significa usar el mapa logístico de la ecuación (4.1) con μ_1 y $f_2(x)$ con μ_2. En este sistema se observan múltiples atractores caóticos que mejoran la deficiencia unimodal del mapa logístico simple [59]. Las figuras 4.1 y 4.3 muestran ejemplos de atractores caóticos para el mapa acoplado logístico no simétrico (NLCM). El NLCM

tiene un rango caótico bien documentado para $3.63 \leq \mu \leq 4$ y $0 \leq \alpha \leq 1$ [64].

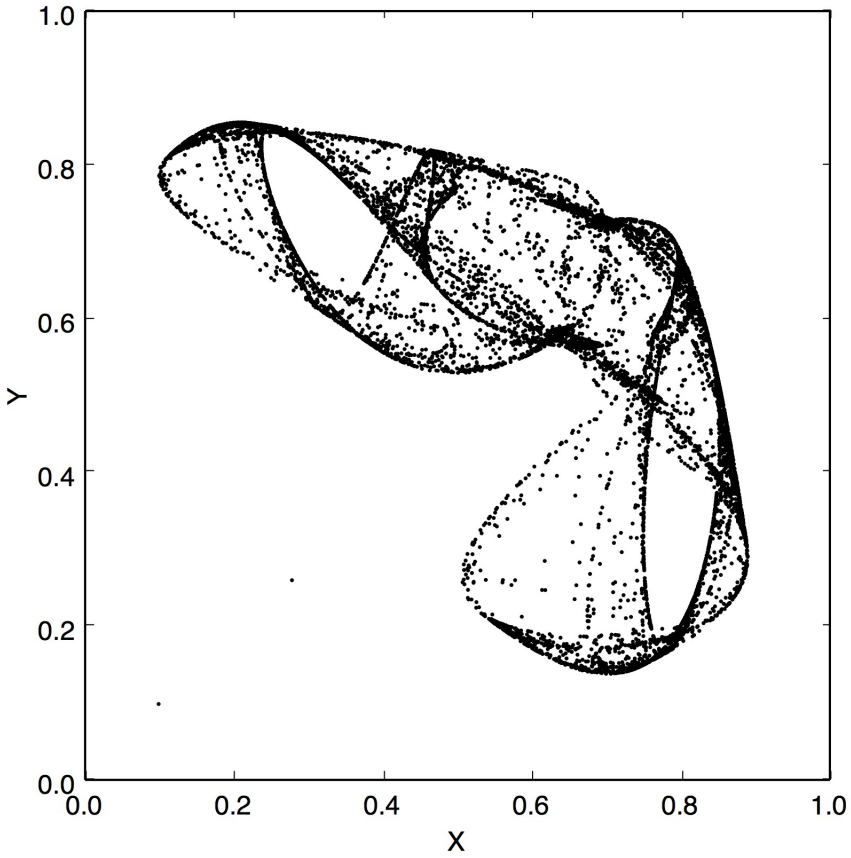


Figura 4.1: Atractor caótico para el mapa acoplado logístico no simétrico $\mu_1 = 3.1$, $\mu_2 = 2.9$ y $\alpha = 0.3314$.

Un esquema teórico para el cifrado asimétrico que explota las propiedades de los sistemas dinámicos no lineales de un sistema dinámico no lineal disipativo de alta dimensión que se distribuye entre un transmisor y un receptor, en el único cifrado caótico de llave pública conocido [65]. Por lo tanto, llaman al método encripción de dinámica

distribuida (EDD). La dinámica del transmisor es pública, y la dinámica del receptor es privada y no se comparte por el canal de comunicación. Un mensaje está codificado por la modulación en los parámetros del transmisor, y esto resulta en un cambio en el atractor general del sistema. Un receptor no autorizado no conoce la dinámica oculta en el receptor y no puede decodificar el mensaje [65]. Esta propuesta ha sido criticada debido a su difícil implementación, lo que la califica como no práctica [66].

Este trabajo toma la propuesta de EDD y la adapta para un sistema de baja dimensión utilizando el mapa logístico acoplado. Estudiamos el cifrado, el descifrado y el ataque común para este sistema criptográfico. Esto es importante para EDD porque proporciona una implementación sin pérdida de seguridad con un costo y velocidad aceptables, lo cual es un requisito criptográfico relevante para los sistemas de cifrado de basados en caos [11]. En nuestra comprensión profunda, durante el tiempo que escribimos este trabajo, esta es la primera implementación computacional completamente funcional para el cifrado de EDD, además de la prueba de concepto de la EDD original. El trabajo aquí presentado es el ejemplo faltante para EDD señalado en la literatura [57].

4.2. Cifrado

La idea básica del cifrado de dinámica distribuida (EDD) es dividir un sistema dinámico de dimensión $D_T + D_R$ en dos partes, donde las D_T variables del transmisor son $t(n) = [t1(n); ...; t_{D_T}(n)]$, y las D_R variables del receptor son $r(n) = [r_1(n); ...; r_{D_R}(n)]$. El receptor recibe la señal escalar $s_t(n)$ del transmisor, y el transmisor recibe la señal

escalar $s_r(n)$ del receptor para formar

$$t(n+1) = F_T(t(n), s_r(n), m(n)) \tag{4.6}$$

y

$$r(n+1) = F_R(r(n), s_t(n)), \tag{4.7}$$

donde $m(n)$, un valor binario (0 o 1), es el mensaje que se desea cifrar. La dinámica de receptor es F_R y la del transmisor es F_T. El receptor debe simular toda la dinámica antes de iniciar la comunicación. Esta simulación crea la lista de puntos necesarios para cifrar y descifrar el mensaje. Para realizar la simulación, el receptor selecciona los parámetros y las ecuaciones que servirán como llaves públicas y privadas, como se explica a continuación.

El cifrado para nuestra implementación de baja dimensión proviene de una relación entre las ecuaciones (4.4), (4.5) y (4.6), (4.7), donde x (ecuación 4.4) es la división dinámica del transmisor y y (ecuación 4.5) es la parte receptora

$$x_{(n+1)} = f_1(x_n) + \alpha(y_n - x_n) + A * m \tag{4.8}$$

$$y_{(n+1)} = f_2(y_n) - \alpha(y_n - x_n). \tag{4.9}$$

El parámetro A es una modulación del mensaje. En esta implementación A toma valores aleatorios entre 0.001 y 0.01 dando una seguridad adicional al sistema. La figura 4.2 muestra un mensaje de 8 bits cifrado (01010111, que utilizando el estándar ASCII es la letra W). Para una fácil identificación, diferenciamos los puntos que corresponden a bits 0 de los que corresponden a bits 1. La seguridad de esta implementación reside

en la superposición y la cercanía de esos puntos. Solo si se tiene la simulación anterior se puede descifrar el mensaje. La figura 4.4 muestra un atractor diferente con un mensaje más largo de 32 bits (01010111 01101111 01110010 01110100, que utilizando el estándar ASCII son cuatro letras: *Wort*). En este esquema, un atractor caótico diferente representa un par diferente de llaves criptográficas. La ecuación (4.8) con sus parámetros correspondientes es la llave pública y la ecuación (4.9) es la llave privada. Algo relevante es que el receptor no necesita conocer el parámetro A para descifrar el mensaje; en este sentido, el parámetro A representa una llave privada del transmisor que proporciona más seguridad al mensaje cifrado. El parámetro A no se usa en el proceso de descifrado. En un atractor caótico después de algunas iteraciones de la dinámica completa, solo conocida por el receptor, la señal enviada por el transmisor converge al atractor o no.

4.3. Descifrado

Un receptor autorizado conoce todas las cantidades, públicas y privadas, y puede establecer fuera de línea los atractores admisibles u otros aspectos dinámicos del sistema, para todos los valores permitidos de $m(n)$. Para el descifrado es necesario conocer previamente todos los puntos de la simulación. El proceso de descifrado corresponde al cálculo de la distancia de cada punto recibido desde el transmisor hasta los puntos de las simulaciones. Es necesario calcular esta distancia a cada punto de la simulación y seleccionar el valor mínimo. Si este valor es inferior a un parámetro de tolerancia es un bit 0 y sino es un bit 1. El parámetro de tolerancia es el mayor valor del rango que puede tomar el parámetro A. A pesar de que este proceso de descifrado parece ser

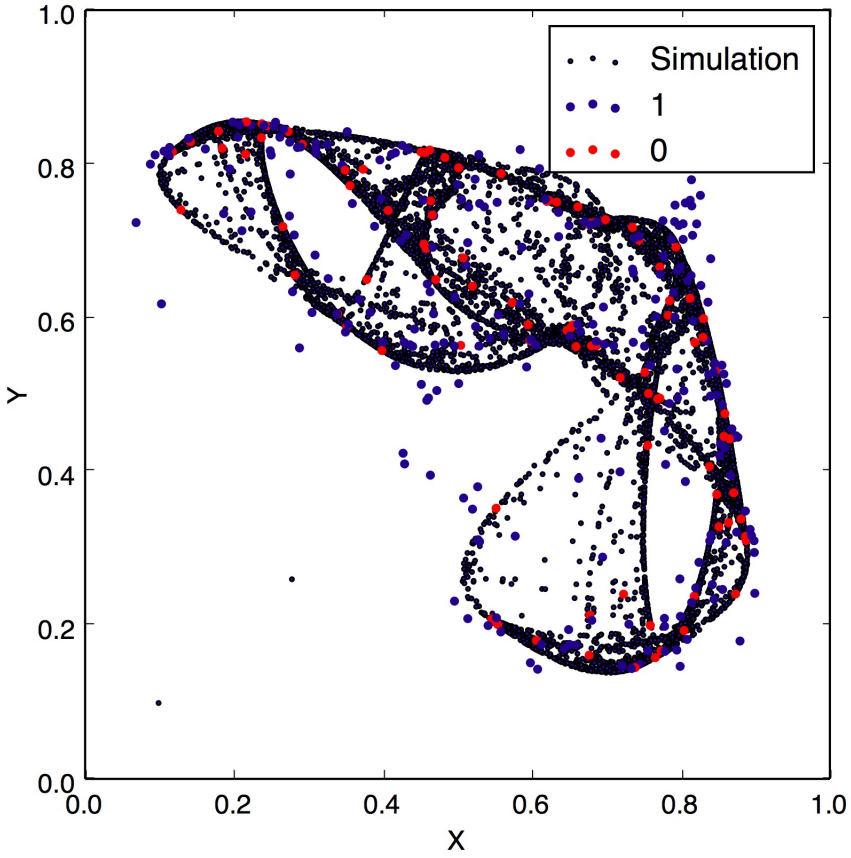


Figura 4.2: Un mensaje cifrado sobre la dinámica conocida de la Fig 4.1

fácil cuando se conoce la dinámica completa, es computacionalmente costoso, aspecto cubierto en el capítulo 6.

En este trabajo exploramos tres formas distintas de abordar y mejorar la problemática asociada al descifrado, dichas formas están descritas a continuación.

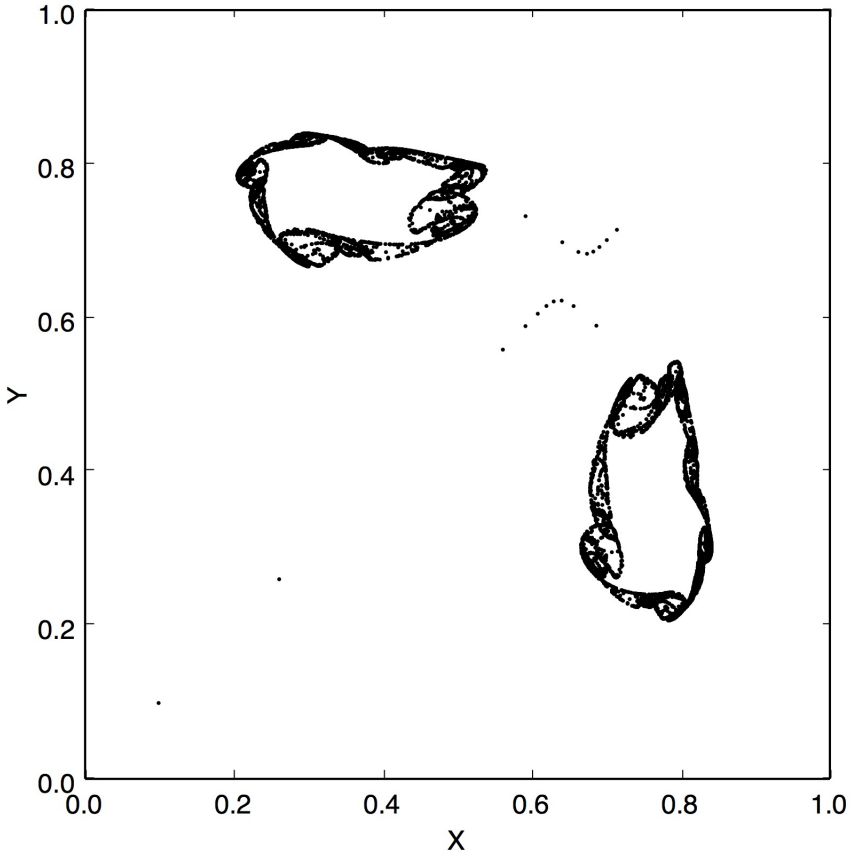


Figura 4.3: Atractor caótico para el mapa acoplado logístico no simétrico $\mu_1 = 2.91$, $\mu_2 = 2.9$ y $\alpha = 0.3314$.

4.4. Automatización del descifrado

Como se mencionó en el marco teórico, en la propuesta inicial de EDD, el algoritmo de descifrado es una mera inspección visual [65]. Entonces, lo descrito en la sección 4.3 es la primera forma de un algoritmo computacional para hacer el descifrado de EDD. En búsca de una mayor eficiencia en el tiempo de ejecución del algoritmo, planteamos tres modificaciones optimizan el proceso de la sección anterior: La que usa NumPy, la

que usa GPU y la que propone una modificación analítica.

4.4.1. Numpy

En esta primera modificación se usan las posibilidades de NumPy para hacer cálculos de álgebra lineal [67]. Se puede computar directamente la distancia euclídia sobre el arreglo de datos que contiene a los datos generados por la simulación previa hecha por el receptor. Por definición, para pares de datos (x_2, y_2) y (x_1, y_1), es decir, en sistemas de dos dimensiones, la distancia euclídia es

$$d = \sqrt{(y_2 - y_1)^2 + (x_2 - x_1)^2}. \tag{4.10}$$

Entonces para el descifrado descrito en la sección anterior, cuando llega un dato representado por un par, en este caso (x_2, y_2), se debe calcular la distancia a todos los puntos de la simulación que el receptor ha calculado previamente, es decir, debe calcular la ecuación 4.10 para cada dato obtenido en la simulación previa. Eso puede ocurrir millones de veces, pues un mínimo aceptable para producir una figura como 4.1 o 4.3 es un millón de puntos. Esto hace computacionalmente costoso el cálculo. Además, se debe calcular el mínimo valor de por lo menos de un millón de distancias euclídias.

Con la ayuda de NumPy se optimiza el cálculo, pues es posible poner todos los puntos generados en la simulación en un `numpy.array()` que nombramos `Sim` y, con la ayuda de `numpy.ones()`, se crea un arreglo de unos del tamaño de `Sim`, que llamamos `Unos`, el cual multiplicamos por el dato recibido para hacer un nuevo arreglo `Rec`. Con esto calcula:

```
d1=numpy.subtract(Rec,Sim)
```

```
d2=numpy.dot(d1,d1)
d=numpy.min(d2)
```

donde en `d1` se calcula la resta distribuida de los dos arreglos, en `d2` el producto interno de esa resta y en `d` el mínimo valor de todas las distancias euclídias, que es lo buscado. Si este valor mínimo se encuentra en el rango dado al parámetro A, se dice que es un cero el bit enviado y si no es un uno. Este manejador de datos propone la forma más eficiente de hacer el cálculo en un solo CPU. En la siguiente sección vemos el caso para varios procesadores.

4.4.2. GPU

La unidad de procesamiento gráfico (GPU, por sus siglas en ingles) está optimizada para hacer cálculos de álgebra lineal y ha sido de gran ayuda para aplicaciones que hacen muchos cálculos numéricos. Con ayuda del paquete Kivy para Python, es posible sacar ventaja de la GPU a través de OPENGL [68]. Con el uso de vectores se puede computar la distancia de la ecuación (4.10) de la sección 4.4.1, asistido por la GPU, el cual tiene un desempeño considerable para cálculos matemáticos.

El espíritu del código del cálculo en OPENGL es el mismo que el presentado en la sección 4.4.1, solo que en vez de arreglos se usan los `glVertex2f`, que permiten introducir los datos de la simulación al GPU. La función `gl_MaxVertexUniformVectors` obtiene la distancia mínima al dato recibido aprovechando al máximo el poder de la GPU, tal que, entre más núcleos se tenga, más se facilita el cálculo de esta función.

4.4.3. Optimización analítica

Para la optimización analítica es posible repensar el algoritmo de descifrado de la sección 4.3. Si al crear en la simulación los datos para x (ecuación 4.8) se van incluyendo de forma ordenada en el arreglo de datos con su respectivo y, cuando se debe calcular la distancia euclídea se puede hacer para una ventana de cercanía de unos cientos de datos sin necesidad de buscar en todo el arreglo. El tamaño de la ventana es determinado por el valor del coeficiente de Lyapunov para el punto donde se encuentra el dato recibido, por lo que la búsqueda es hace en el rango que permite usar solo datos cercanos al dato recibido. Por lo demás, el algoritmo es idéntico al descrito en la sección 4.4.1 .

De esta forma, se da por completado el objetivo específico 2 de esta tesis, donde se ha desarrollado un algoritmo de cifrado. En el capítulo 6 se da en forma usual el nuevo algoritmo criptográfico, con su respectivo ejemplo de su uso. Este cifrado ha sido publicado en [53]. Con un esquema de cifrado construido, procedimos a evaluar su seguridad, pues, aunque conocemos que el esquema es resistente a los ataques cuánticos, es fundamental estar seguros de que es resistente a los ataques clásicos. Esta es la tarea a la cual se aboca el siguiente capítulo.

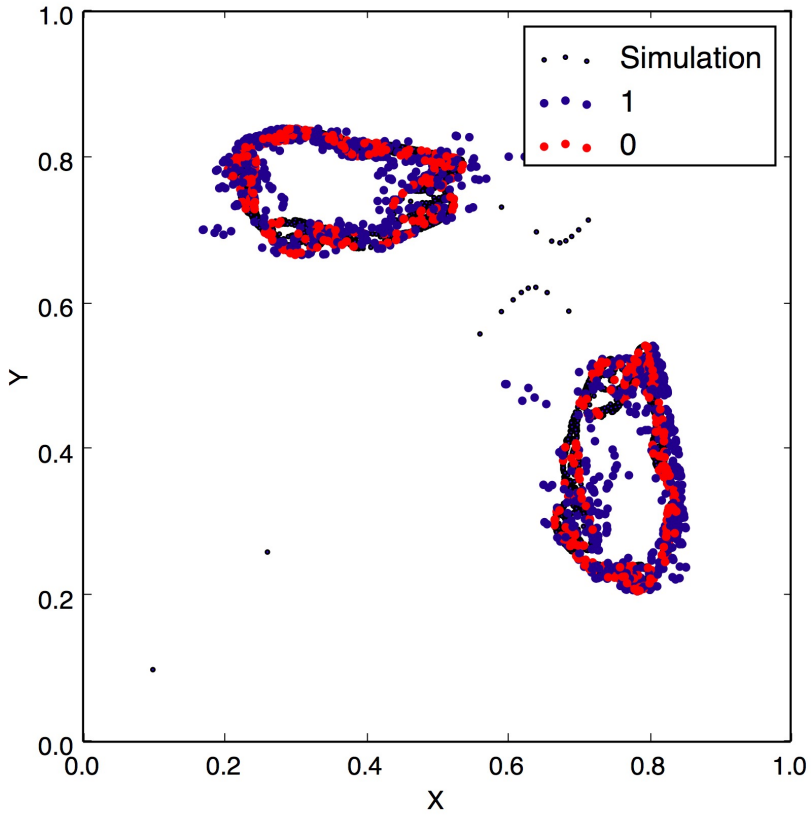


Figura 4.4: Un mensaje cifrado sobre la dinámica conocida de la Fig 4.3

Capítulo 5

Criptoanálisis

En este capítulo se analiza la seguridad del nuevo esquema de cifrado y se compara con el caso general de EDD de la sección 5.1. Para esto, se atacan los datos que viajan por un canal de comunicación entrenando una cadena de oculta de Markov, sección 5.2, para intentar reconstruir la superficie decisión, que sería la figura 4.1, en el caso a trabajar en este capítulo.

5.1. Mensaje transmitido por el canal

Un receptor no autorizado puede intentar varios métodos para atacar EDD y decodificar el mensaje secreto $m(n)$ pero, como se ha demostrado, el único ataque para el cual no existe una defensa obvia es el analizado en este capítulo [65]. Como la seguridad del cifrado reside en que la señal que viaja en un canal es caótica, nuestra implementación aún es defendible como el EDD original. La figura 5.1 muestra los datos que viajan en el canal de comunicación, donde un receptor no autorizado no puede resolver fácilmente el mensaje. Debido a la transitoriedad topológica, entre más datos

son transmitidos más espacio se ocupa en el canal transformando la figura 5.1 en un cuadro sólido relleno. Por otro lado, la figura 5.2 muestra el caso cuando el atractor no es caótico, que facilita identificar los dos estados (0 o 1). La figura no cambia por más estados que se trasmitan.

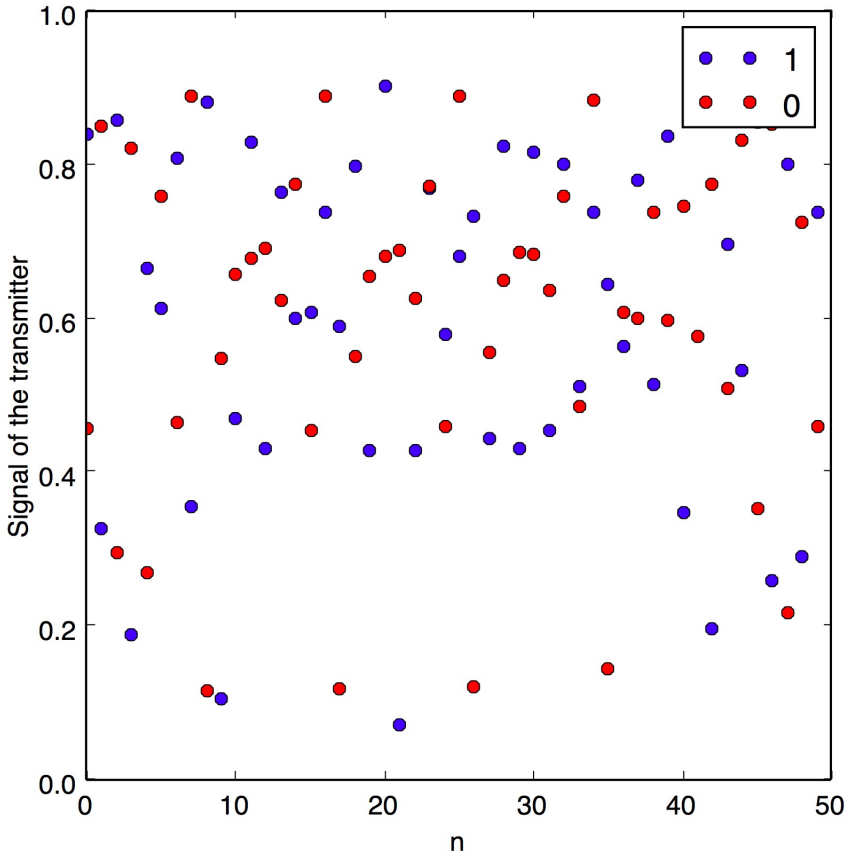


Figura 5.1: Transmisión en el canal de comunicación de la letra W con el nuevo algoritmo de cifrado cuando el atractor es caótico

Uno de estos métodos es reconstruir las posiciones de los atractores que corresponden a la transmisión de 0 o 1 almacenando y agrupando muestras de muchos bits

transmitidos. Conocer las posiciones de los atractores permite a un receptor no autorizado descodificar el mensaje utilizando el mismo método que el receptor autorizado [65].

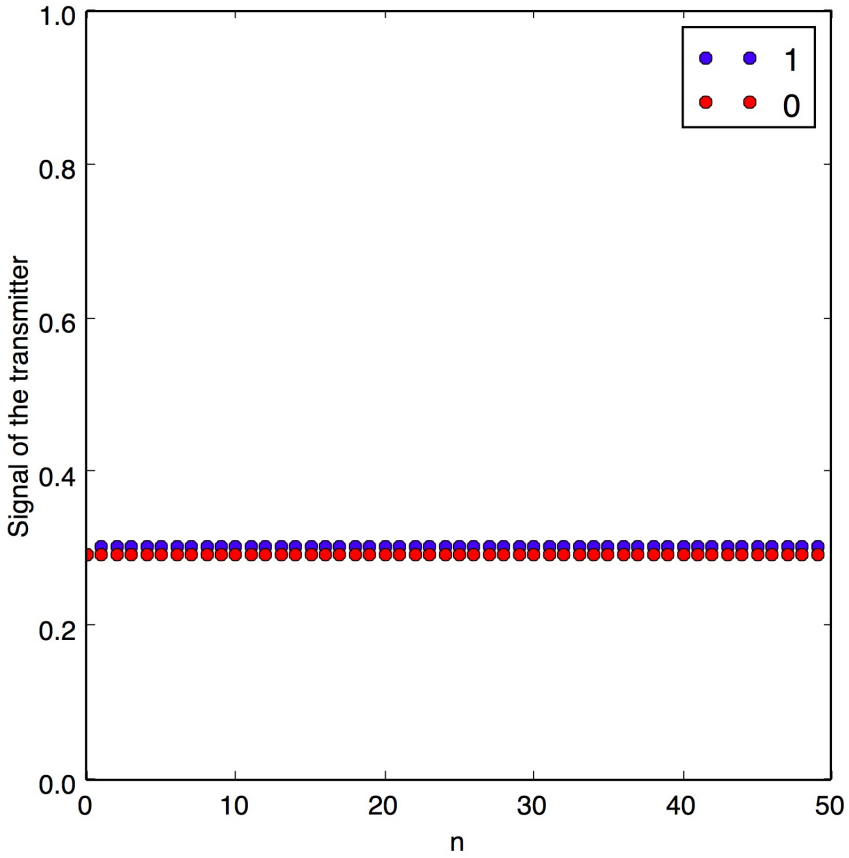


Figura 5.2: Transmisión en el canal de comunicación de la letra W con el nuevo algoritmo de cifrado cuando el atractor no es caótico.

La dinámica caótica transmitada por el canal, sumada al ruido natural y a otros, hacen que la dinámica sea estocástica.

5.2. Ataque por modelo de Markov

En este ataque un receptor no autorizado puede intentar generar un modelo de Markov oculto de la dinámica pública del transmisor para cada valor posible del mensaje m, y obtener una estimación m de la probabilidad máxima (ML, por sus siglas en inglés) del mensaje m. El mensaje decodificado está dado por

$$m' = \underset{(m\in(0,1))}{\text{máx}}\, p(s_t(1), ..., s_t(D_T + 1) \mid m). \tag{5.1}$$

Para generar el modelo oculto de Markov, el receptor no autorizado debe cuantificar el estado del transmisor en un espacio de incrustación reconstruido con retardo de tiempo y estimar las probabilidades de transición de estado, así como las probabilidades de observación del modelo [69].

El ataque propuesto fue implementado para nuestro modelo de baja dimensión. La figura 5.3 muestra cómo para un mensaje cifrado usando un mapa no caótico el ataque es efectivo, con una precisión de entrenamiento del 95 %. Cuando el número de bits es inferior al necesario para entrenar el modelo la precisión se reduce a menos del 40 % [11]. Esto hace que el ataque probabilístico no pueda ser llevado acabo [54]. No se puede evitar este ataque en particular. Eso se muestra en la figura 5.4, pero la línea roja, que es el límite de decisión, está lejos de ser correcto. La figura 5.5 tiene un número mayor de bits transmitidos, 60 000. El límite es más visible pero no es suficiente para recuperar el mensaje. La Figura 5.6 muestra un número de bits igual a lo necesario para realizar el ataque. Se ve cómo la curva de decisión puede resolverse para 0 o 1 bits. El número de estados que se pueden transmitir antes de que se pueda

resolver la curva de decisión está dado por la ecuación

$$N_s \approx \left(\frac{L_T}{L_q}\right)^{D_T} \tag{5.2}$$

donde L_T es el rango de los datos transmitidos, L_q es la cuantización en la señal y D_T la dimensión de la dinámica del transmisor [69]. Esta ecuación (5.2) sigue siendo útil en nuestro caso porque se ha derivado en general para cualquier RMA. Para nuestra implementación, $N_s \approx 4 \times 10^7$, lo que muestra por qué la Figura 5.6 puede resolver la curva de decisión y cómo la seguridad de esta implementación es del mismo nivel que la de la propuesta EDD original.

Una característica relevante a destacar es que un cambio muy pequeño en la dinámica del receptor crea atractores muy diferentes, como se observa en las figuras 4.1 y 4.3. El único cambio entre ellas es la variación en el valor de μ_1 en solo 0.2. Aun así, el diferencial y la dinámica producida es totalmente diferente. Esto es útil, porque muestra que no se necesita cambiar la llave pública para tener una nueva dinámica criptográfica que represente una buena opción para protegerse del ataque descrito en este capítulo.

El NLCM representa una mejora con respecto al mapa logístico, pero otro mapa con nuestro esquema criptográfico de baja dimensión, como el mapa lineal por partes [70] o el de un sistema de la naturaleza [71], representa un nuevo conjunto de llaves criptográficas.

Con esto se da como cumplido el objetivo específico 3, ya que se ha sometido exitosamente el algoritmo desarrollado a un ataque conocido para los EDD [53].

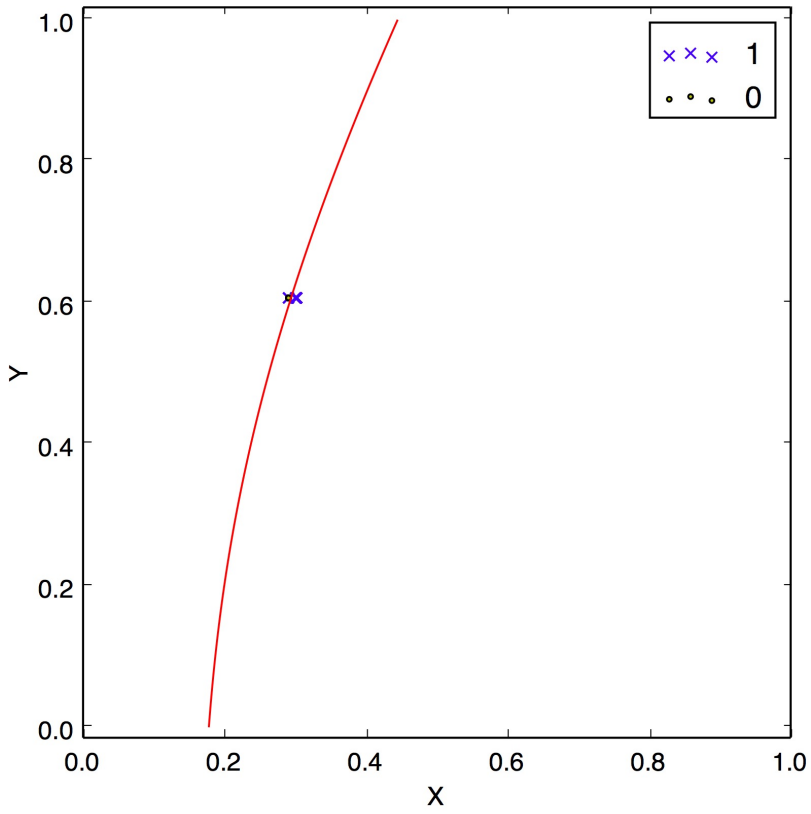


Figura 5.3: Superficie de decisión (línea continua) obtenida del ataque cuando el atractor no es caótico.

La degradación del caos es un problema bien conocido para los criptosistemas basados en caos [72]. En EDD, este problema se resuelve por dos medios. El primer medio es con un algoritmo de verificación de los datos. En EDD, esto no interfiere con el algoritmo de cifrado, ya que la simulación de los estados caóticos se hace previo al cifrado. El segundo medio es mediante el uso de mapas acoplados de la naturaleza, ya que hay información adicional por parte del fenómeno natural descrito por el mapa acoplado,

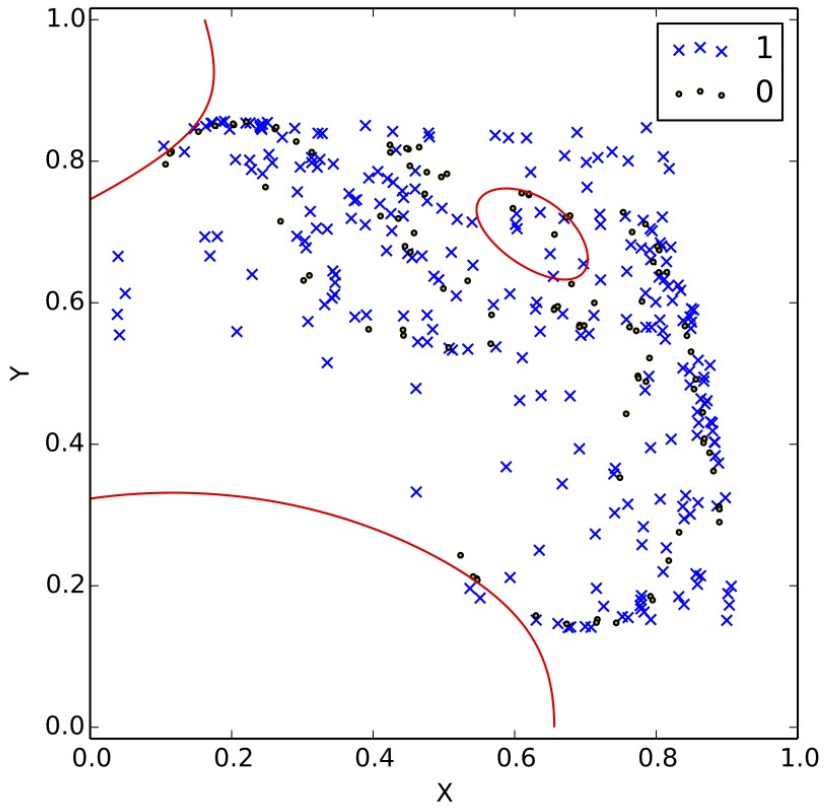


Figura 5.4: Superficie de decisión (línea continua) obtenida del ataque cuando para el atractor de la Fig. 4.1 con un número bajo de observaciones.

que permite indicar la degradación del caos. Este último medio es un resultado de esta tesis ya publicado [78, 53].

El trabajo realizado en este capítulo y el anterior se puede resumir en un algoritmo formal basado en los criterios criptográficos conocidos [11], temática del siguiente capítulo.

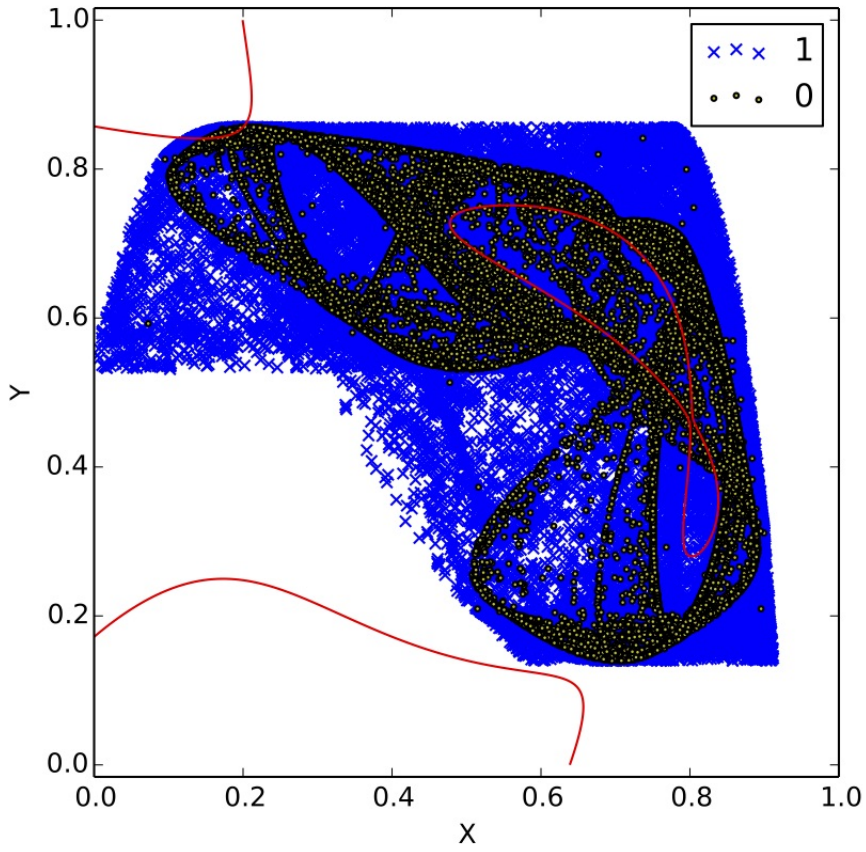


Figura 5.5: Superficie de decisión (línea continua) obtenida del ataque cuando para el atractor de la Fig. 4.1 con un número medio de observaciones.

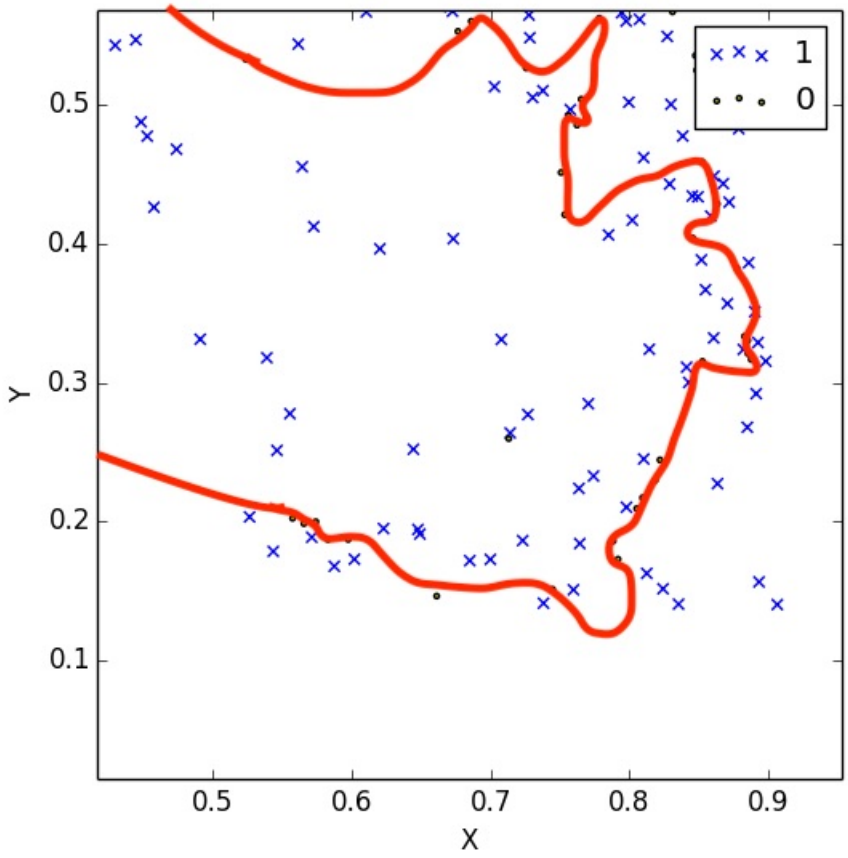


Figura 5.6: Superficie de decisión (línea continua) obtenida del ataque cuando el atractor es el mismo de la Fig. 4.1 y se conocen suficientes observaciones.

Capítulo 6

Aplicación del nuevo sistema criptográfico a un mapa de la naturaleza

En la sección 6.1 se detalla, en forma de algoritmo, el nuevo sistema criptográfico expuesto en el capítulo 4, con lo cual se abre la posibilidad a nuevas familias de sistemas de cifrado, aspecto detallado en la sección 6.2. Aprovechando lo anterior, se caracteriza el nuevo mapa obtenido de un fenómeno natural en la sección 6.3, aportando el procedimiento para obtener los parámetros que definen al mapa y finalizando con el estudio de los tiempos asociados al descifrado y cifrado para el algoritmo en la sección 6.4.

6.1. El Algoritmo

En esta sección se proporcionan formalmente los algoritmos de cifrado y descifrado de nuestra propuesta para después dar un ejemplo del escenario de comunicación entre

Alicia y Bob. Es necesario tener claro el proceso de selección de las llaves y los requerimientos de ejecución antes de describir los algoritmos.

Selección de llaves. Alicia debe seleccionar μ_1, μ_2 y α. Además, Bob tiene su propia llave privada, el parámetro A, que se selecciona en el intervalo aceptado por Alicia.

Requerimientos. Alicia con las llaves seleccionadas debe calcular la simulación con la ayuda de las ecuaciones (4.8) y (4.9). Esto genera una larga lista de puntos (X, Y). Además, Alicia define una tolerancia relacionada con el valor mínimo que aceptará para el parámetro A.

Algoritmo para el cifrado. Para cifrar un mensaje m, Bob debe hacer lo siguiente:

a) Obtener la llave pública auténtica de Alicia (ecuación (4.8) y μ_1)
b) Representar el mensaje como código binario.
c) Obtener los primeros cincuenta datos de Alicia X.
d) Calcular cincuenta datos Y para el primer bit, utilizando la ecuación (4.8), μ_1, datos X y el parámetro A.
e) Enviar datos Y a Alicia.
f) Repetir para los siguientes bits con un nuevo conjunto de cincuenta datos X de Alicia.

Algoritmo para descifrado. Para recuperar el mensaje, Alicia debe hacer lo siguiente:

a) Emparejar los primeros 50 datos para X enviados a Bob con los 50 primeros datos Y recibidos de Bob.
b) Tomar solo las últimas doce parejas.
c) Para cada par, calcule la distancia euclidiana a cada punto de la lista larga (X, Y) de la simulación y conserve el valor mínimo de la distancia.
d) Si el mínimo promedio de los doce calculados es mayor que la tolerancia es un bit 1, de lo contrario es un bit 0.
e) Repetir para los próximos cincuenta datos Y de Bob.

Ahora se presenta un ejemplo. Digamos que Alicia quiere comunicarse con Bob. Él tiene un mensaje muy importante para enviarle: la letra W. Quieren usar nuestra propuesta criptográfica para transmitir esta carta por un canal seguro. Primero, Alicia debe elegir las llaves de cifrado, μ_1, μ_2 y α, recordando que las llaves deben llevar a condiciones caóticas (ella podría verificar esto con la ayuda del exponente Lyapunov de NLCM [64]). Digamos que usa $\mu_1 = 3.1$, $\mu_2 = 2.9$, y $\alpha = 0.3314$, el mismo atractor de la figura 4.1. Alicia anuncia públicamente la ecuación (4.8) con μ_1 y α de la selección anterior, y mantiene μ_2 y la ecuación (4.9) secretos. Bob toma esta información pública para transmitir el mensaje.

Alicia hace la simulación fuera de línea usando las ecuaciones (4.8) y (4.9) y produce una larga lista de puntos (X, Y). La ecuación (4.9) con sus parámetros es la llave privada, que no viaja por canal. Para iniciar la comunicación de la larga lista que tiene, Alicia envía cincuenta datos X a Bob. Toma su mensaje y lo convierte a binario usando el estándar ASCII, por lo que W será 01010111, 8 bits. Toma el primer bit, el 0, usando la ecuación (4.8) vuelve a calcular un nuevo par Y para los cincuenta

recibidos de Alicia y se lo devuelve. En esta ecuación (4.8) el parámetro A es Bob quien realmente lo elige, es mejor si es aleatorio, también para calcular la ecuación necesita un valor de inicio Y_0 que selecciona también al azar.

Eva, un genio malvado que está escuchando los datos enviados por Bob en el canal, no puede reconstruir el primer bit a partir de los 50 números gracias a las llaves privadas de Bob y a que la ecuación (4.8) se encuentra en estado caótico. Eva tendrá que esperar hasta tener suficientes datos para usar el ataque descrito en la sección 5.2. Alicia recibe los 50 números Y y, utilizando los últimos 12, combinados con los últimos 12 de X que envió y calcula la distancia a cada punto de la larga lista a partir de la simulación que tiene y toma las distancias mínimas para los 12 pares. Si el promedio de las 12 distancias mínimas es menor que la tolerancia (el valor mínimo que A puede ser) es un bit 0 y si es mayor que la tolerancia es un bit 1, en este caso, Alicia verá un bit 0. Ahora, el proceso se repite para los siguientes bits. Alicia no necesita enviar datos X adyacentes a Bob para la transmisión del mensaje.

6.2. Familias de cifrados

El algoritmo anteriormente descrito no está estrictamente ligado al NLCM, por que podemos utilizar cualquier otra RMA de baja dimensionalidad. Esto abre un mundo de posibilidades para familias de sistemas de cifrados usando nuestro algoritmo. Un problema importante a atacar es la degradación del caos y de la eficiencia. Es conocido como al usar una RMA directa de la naturaleza podemos, gracias a las características del sistema, tener una corroboración externa del estado caótico del sistema, solventado la degradación del caos [73, 78].

Para usar otro RMA solo debemos remplazar en el algoritmo anterior la ecuación (4.8) por la correspondiente que describe el RMA en cuestión. Además, se deben identificar los parámetros correspondientes para μ_1 y μ_2 en el nuevo RMA.

Como es nuestra motivación, es interesante utilizar un mapa con alta entropía pues sabemos que los computadores cuánticos son incapaces de simularla. Esto proporcionaría una seguridad adicional al sistema criptográfico a utilizar.

6.3. Un mapa de la naturaleza los QPO

Un sistema por excelencia no lineal y de alta entropía es un agujero negro [74]. En esta sección proponemos un RMA para su caótica atmósfera. A partir del análisis cosmográfico, obtenemos los valores para los parámetros a utilizar en la ecuación del RMA.

Los agujeros negros mantienen a su alrededor una atmósfera conocida como *disco de acreción*. El disco puede tener oscilaciones. Aquellas que se pueden observar regularmente se les conocen como oscilaciones cuasiperiódicas (QPO, por sus siglas en inglés).

Al diseñar un modelo para la compleja dinámica del disco para un agujero, el objetivo es proponer la expresión matemática más simple con soluciones que incluyen subarmónicos cuadráticos y cúbicos de una manera compacta y observacional productiva. Esto se puede lograr con un solo oscilador no lineal impulsado de una dimensión, descrito por [75]

$$\ddot{x} + \omega_0^2 x - \epsilon x^2 - \delta x^3 = B\cos\omega t. \tag{6.1}$$

Este modelo es un dispositivo que incorpora sucintamente los modos de oscilación y por lo tanto, es más que una representación visual simplificada del sistema. Asumimos que la ecuación (6.1) representa la dinámica de los QPO después de que ha alcanzado un estado estacionario, siendo la variable x una medida del desplazamiento del fluido. Supongamos además que ϵ y δ son lo suficientemente pequeños como para que las consideraciones perturbativas tengan sentido. Suponga que el oscilador descrito por los dos primeros términos en el lado izquierdo de la ecuación, $\ddot{x} + \omega_0^2 x$, representan un modo fundamental del agujero negro (axisimétrico) g (una oscilación inercial-gravitacional), por lo que establecemos ω_0 como la frecuencia del modo g.

Procedemos utilizando un enfoque cosmográfico para determinar el valor de los parámetros cosmológicos. Buscamos hacer una comparación entre los valores medidos de la magnitud aparente (m) de las supernovas con los valores esperados dado la medición de su desplazamiento al rojo(z). La magnitud aparente está dada por

$$m = 5 \log \frac{d_L}{10} + M, \tag{6.2}$$

en términos de la distancia de luminosidad (d_L) y la magnitud absoluta (M), que se conoce para las supernovas constantes. Dado que la distancia de luminosidad se da en términos de la distancia física (r_0) entre la señal fuente y el observador y los z medidos, necesitamos expresar r_0 en términos de su medición del desplazamiento al rojo. Hacemos esto usando la geodésica nula en nuestra métrica del espacio-tiempo y el desplazamiento al rojo cosmográfico. Por la geodésica nula tenemos que

$$d_L = (1 + z) r_0 a_0 \tag{6.3}$$

y

$$-c\int_{t_*}^{t_0}\frac{dt}{R(t)} = f(r_0) = \int_{r_0}^{0}\frac{dr}{\sqrt{(1-kr^2)}}, \tag{6.4}$$

con

$$f(r_0) = \begin{cases} -\sin^{-1}(r_0) & (k=+1) \\ -r_0 & (k=0) \\ -\sinh^{-1}(r_0) & (k=-1) \end{cases} \tag{6.5}$$

Ampliamos el factor de escala $R(t)$ en la métrica del espacio tiempo en cuestión

$$R(t) = R(t_0)[1+H_0(t-t_0)-\frac{1}{2!}q_0H_0^2(t-t_0)^2+\frac{1}{3!}j_0H_0^3(t-t_0)^3+\frac{1}{4!}s_0H_0^4(t-t_0)^4+...] \tag{6.6}$$

y, utilizando la relación cosmológica de corrimiento al rojo con la expansión del factor de escala [76], obtenemos el tiempo de vuelo desde la fuente hasta el planeta tierra ($T \equiv t_0 - t_*$ donde t_* es el tiempo en el que se emitió la señal) en función del desplazamiento al rojo medido

$$z+1 = \frac{R(t_0)}{R(t_*)}, \tag{6.7}$$

con lo que el factor de escala es

$$\frac{R(t_0)}{R(t_*)} = 1 + H_0T + \frac{2+q_0}{2}H_0^2T^2 + \frac{6(1+q_0)+j_0}{6}H_0^3T^3 + \frac{24-s_0+8j_0+36q_0+6q_0^2}{24}H_0^4T^4 + ... \tag{6.8}$$

Invirtiendo numéricamente se obtiene

$$T\left(\frac{z}{H_0}\right)^{-1} = 1 - \left[1 + \frac{q_0}{2}\right] z + \left[1 + q_0 + \frac{q_0^2}{2} - \frac{j_0}{6}\right] z^2 - \left[1 + \frac{3}{2}q_0(1+q_0) + \frac{5}{8}q_0^3 - \frac{1}{2}j_0 - \frac{5}{12}q_0 j_0 - \frac{s_0}{24}\right] z^3 + ... \quad (6.9)$$

Resolvemos la integral del lado izquierdo en la ecuación (6.4) con la expansión del factor de escala $R(T)$ de la ecuación (6.6) y sustituimos T en términos de z como se hace en la ecuación (6.9). Por lo tanto, podemos usar la distancia de luminosidad d_L en términos del desplazamiento al rojo z. Empleamos un análisis de razón de probabilidad marginal para encontrar los mejores valores de ajuste de los parámetros cosmológicos. Entonces usamos los datos del catálogo de fuentes de rayos X [77], con lo que obtenemos los rangos de los parámetros ϵ y δ, para evaluar la ecuación (6.1) y representados en la figura 6.1.

Usando esto se puede reproducir para la RMA en la ecuación (6.1) las figuras usadas en el capítulo 4. Para el ejemplo de la figura 4.4 podemos observar en la figura 6.2 el cifrado y en la figura 6.3 es el resultado del ataque.

Eso muestra que sin un aumento en la dimensión comparado con NCLM, tenemos una mejor resolución en la retícula asociado al hecho de usar datos reales para el sistema. No es posible con el mismo número de datos de entrenamiento poder atacar el cifrado. Esto comprueba de forma experimental lo que la literatura afirma [78].

Con esto se cumple el objetivo específico 1 de esta tesis, ya que se ha caracterizado un sistema de la naturaleza que simula la relación de alta entropía para el enredo

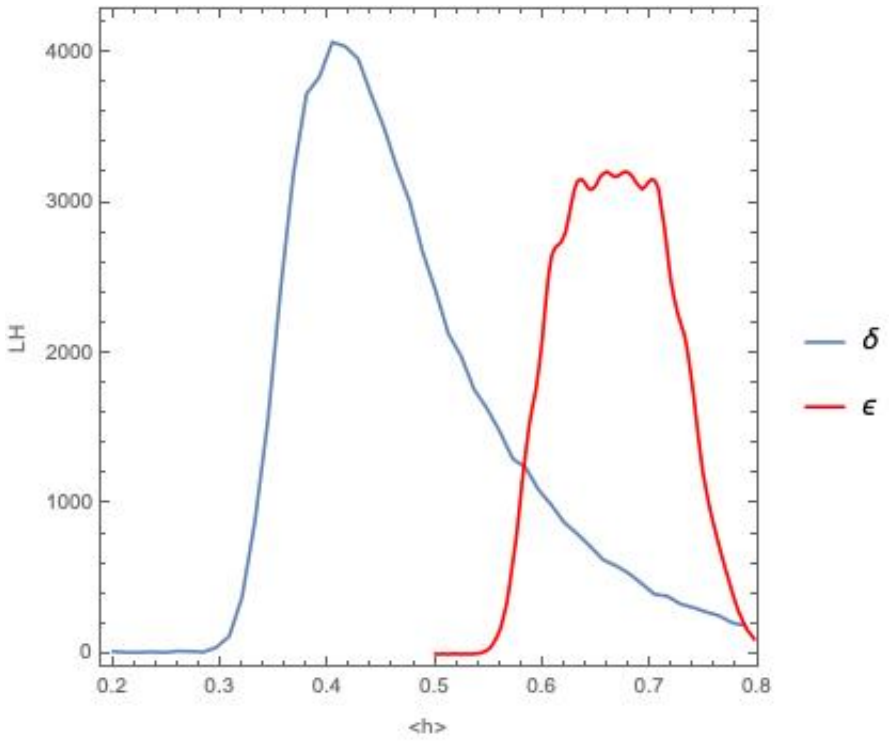


Figura 6.1: Determinación de los parámetros ϵ y δ

cuántico. Donde el detalle del procedimiento y los resultados han sido publicados [79].

6.4. Eficiencia computacional del algoritmo

Nuestra implementación se realizó en Python con la ayuda de los paquetes Numpy para la gestión de datos y HMMlearn para la cadena de Markov utilizada en el ataque. Los cálculos se realizaron en un sistema Linux con una PC Core i7 2.6GHz y una memoria RAM de 16 GB. Las figuras de la 4.1 al 4.4 se calcularon con un millón de puntos para el atractor caótico con un tiempo promedio de 33 s para el cálculo. La figura 5.6 tiene el tiempo de cálculo más alto. El entrenamiento de la cadena de Markov tomó

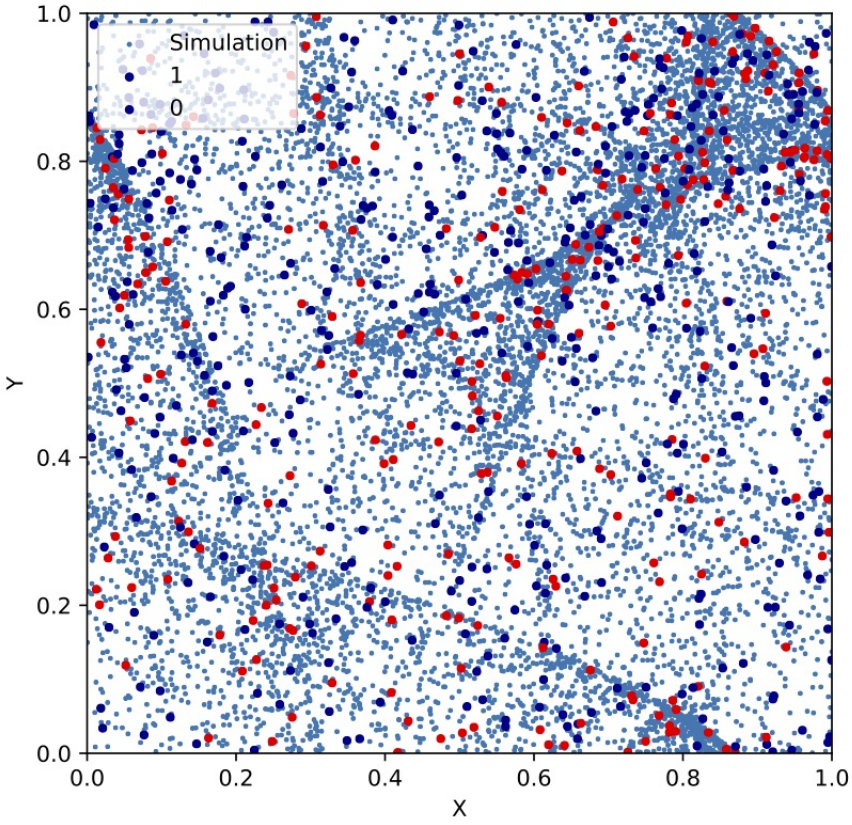


Figura 6.2: Un mensaje cifrado sobre la dinámica de la ecuación 6.1

un tiempo promedio de 72 horas debido a los 50 millones de bits necesarios para el entrenamiento. Se usaron 100 cadenas de bit para cada caso obtenidas aleatoriamente. El tiempo promedio es el reportado en las tablas 6.1 y 6.2, con una desviación estándar de 0.0001 en cada caso.

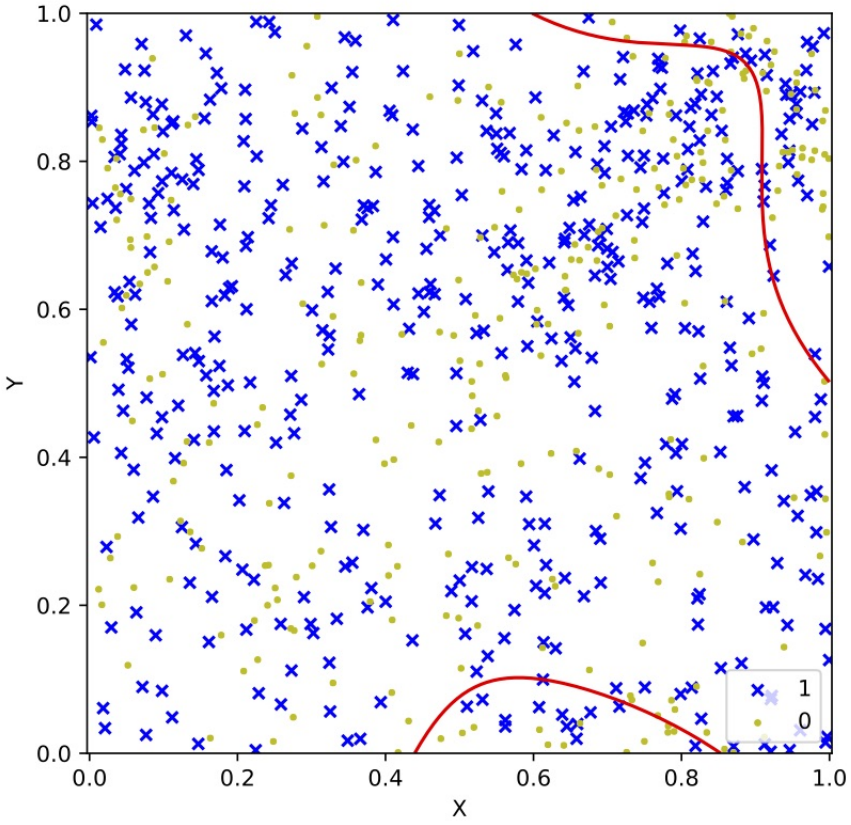


Figura 6.3: Ataque sobre el mensaje de la figura 6.2

Tabla 6.1: Detalles del tiempo de cifrado y descifrado.

Bits	Tiempo de cifrado (s)	Tiempo de descifrado (s)
8	0.0450	141.1081
16	0.0997	293.1638
32	0.2017	600.4974
64	0.3199	1318.4119
128	0.5973	2567.0675
256	1.7431	5297.5988

La tabla 6.1 muestra el rendimiento de los algoritmos de cifrado y descifrado donde, como se esperaba, es necesario más tiempo para procesar ambos debido a más bits. Es notable cómo el tiempo de cifrado es relativamente pequeño en comparación con el tiempo de descifrado, lo cual es claramente evidenciado en la figura 6.4. Este costoso tiempo de descifrado es la debilidad del EDD y abre una investigación adicional en este tipo de sistemas criptográficos, pero hemos mostrado cómo se puede implementar el EDD.

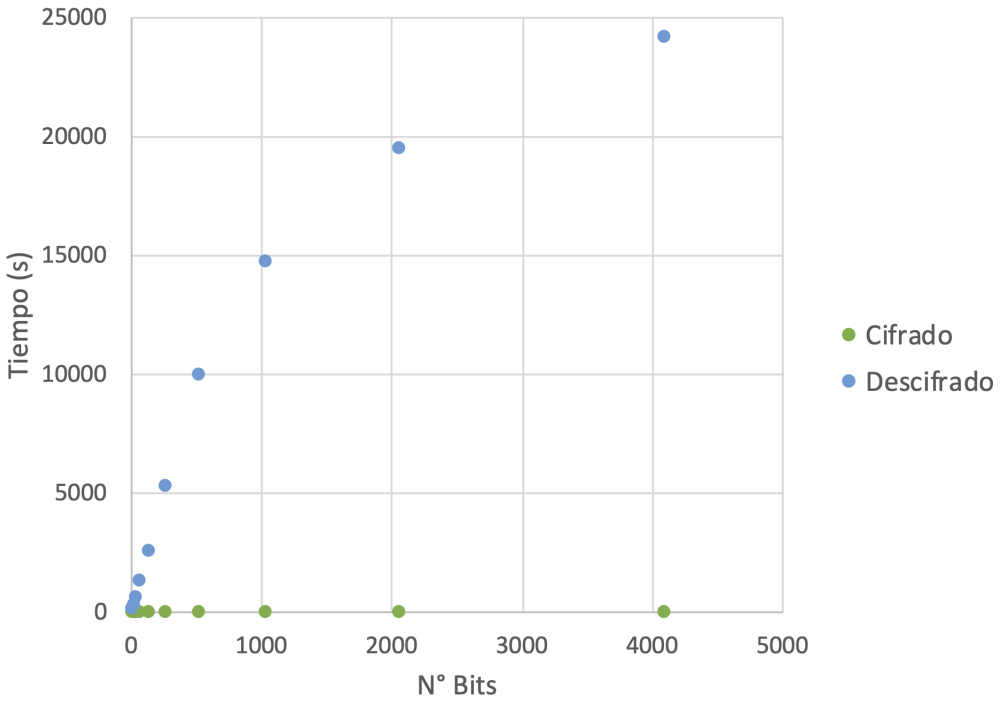


Figura 6.4: Tiempo de cifrado y descifrado con respecto al tamaño en bits, para los resultado de la tabla 6.1

Tabla 6.2: Detalles de los tiempos de cifrado y de los distintos descifrados.

Bits	Cifrado (s)	Descifrado (s)	Numpy (s)	GPU (s)	Analítico (s)
8	0.0450	141.1081	70.5541	2.0990	3.0212
16	0.0997	293.1638	146.5819	4.1330	6.0222
32	0.2017	600.4974	300.2487	8.1670	9.0232
64	0.3199	1318.4119	659.2060	16.2010	16.0242
128	0.5973	2567.0675	1283.5330	32.2350	32.0252
256	1.7431	5297.5988	2648.7990	60.2690	60.0262
512	2.8889	10028.1301	5014.0650	88.3030	89.0272
1024	4.0347	14758.6614	7379.3300	116.3370	117.3610
2048	5.1805	19489.1927	9744.5960	144.3710	145.8620
4096	6.3263	24219.7241	12109.8600	172.4050	174.3630

La tabla 6.2 muestra la realización de los algoritmos de cifrado y descifrado, donde, como se espera, el tiempo de ambos incrementa porque se necesitan procesar más bits. Es notable cómo el tiempo de cifrado es relativamente pequeño comparado con el tiempo de descifrado. El tiempo de descifrado obtenido con Numpy optimiza el tiempo de ejecución, aunque no lo hace considerablemente. En cambio, cuando los cálculos son hecho en la GPU como se muestran en la columna respectiva, se presenta una mejoría. Esto hace al algoritmo de descifrado práctico para su escalamiento al uso de aplicaciones criptográficas, aunque el uso de la GPU puede traer una debilidad a la seguridad informática [80], además de que no todas las plataformas cuentan con una. La modificación analítica planteada al algoritmo representa una clara ventaja, como se muestra en la última columna y en la figura 6.5, pues su desempeño es similar al

uso de la GPU, pero para implementarlo no se necesitan paquetes ni requerimientos adicionales.

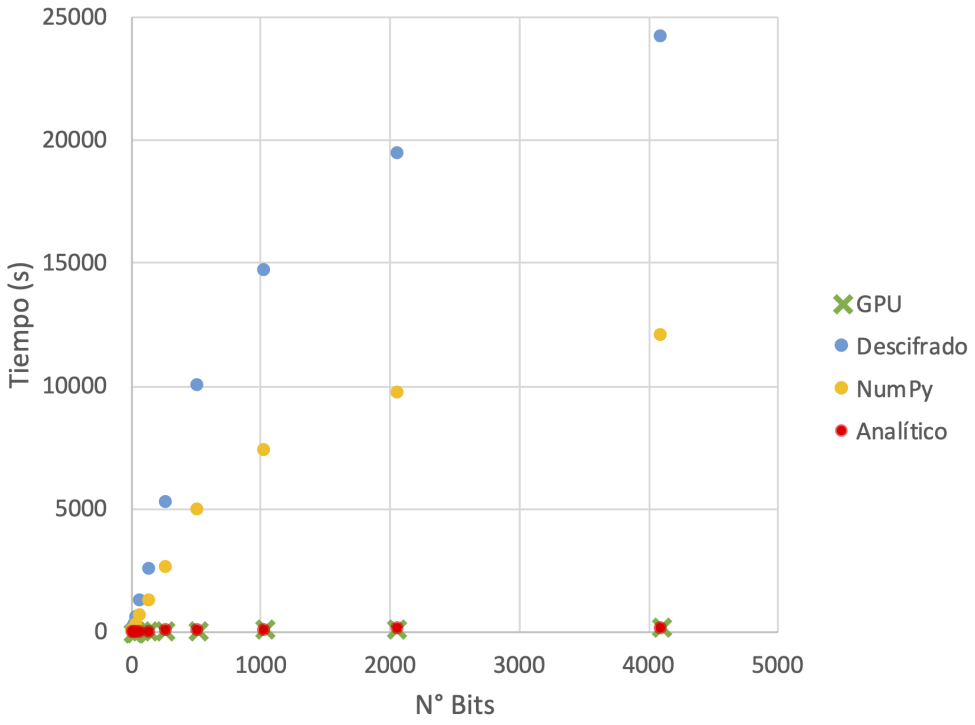


Figura 6.5: Tiempo de descifrado para las diferentes modificaciones al algoritmo descifrado, datos mostrados en la tabla 6.2

Con esto se da por cumplido el objetivo específico 4 donde se evaluó la eficiencia de los algoritmos que son parte del nuevo sistema criptográfico. Estos resultados ya fueron previamente publicados [81]. Llegando así al cumplimiento de todos los objetivos de esta tesis.

Capítulo 7

Conclusiones

En esta tesis se describe un nuevo esquema de cifrado caótico inspirado en EDD. Se construyeron los algoritmos de cifrado y descifrado con base en los requerimientos criptográficos para los cifrados caóticos. Además, se efectuó un análisis criptográfico del esquema creado que permitió evaluar la resistencia del cifrado al ataque conocido para EDD.

Se planteó por primera vez un escenario real de comunicación para EDD, lo que posibilitó hacer una implementación computacional para este nuevo esquema de cifrado y así evaluar la eficiencia de la ejecución de los algoritmos. Esto último permitió optimizar el algoritmo de descifrado mediante la computación paralela, con el uso de la GPU y mejores definiciones de los procesos asociados a través de las distancias caóticas usando el coeficiente de Lyapunov.

Se realizó una implementación completa de EDD, en la cual se han descrito los procesos de cifrado y descifrado, que muestran cómo la implementación de EDD es

posible con un sistema dinámico de baja dimensión. Esto es relevante para el campo de investigación, ya que brinda un ejemplo funcional para EDD con el mismo tipo de seguridad que brindan los sistemas de alta dimensión. Esto es un requisito criptográfico clave para los sistemas criptográficos basados en caos. Su seguridad no está asociada a la dificultad matemática sino a la propiedad caótica. Además, esta implementación abre la posibilidad de investigar mejores formas de incrementar la eficiencia de EDD, ya que es factible medir tiempos de ejecución y cantidad de espacio para los algoritmos de cifrado y descifrado.

Esta es la primera vez que se describe un escenario de comunicación para EDD. Se tiene una forma concreta y segura de asignar las llaves necesarias para el cifrado, lo que representa una mejoría a la EDD tradicional, con lo cual se tiene un sistema de cifrado funcional de llave pública resistente a ataques cuánticos. Con esto se da por satisfecho el objetivo general de este trabajo de tesis.

Un punto clave a resaltar es que este trabajo da por primera vez una automatización del procedimiento de descifrado para EDD, que en la propuesta original era una mera inspección visual. Aun con la modificación planteada al algoritmo de descifrado de EDD, se logró una eficiencia de ejecución del mismo nivel al caso del cifrado, requisito criptográfico fundamental para poder hacer disponible al público este nuevo esquema de cifrado.

La EDD de baja dimensión presentada en esta tesis representa una plataforma para evaluar diferentes mapas acoplados, ya que los algoritmos descritos de cifrado y descifrado no son dependientes formalmente del RMA seleccionado, por lo que el esquema

de cifrado es una familia de esquemas de cifrados para distintos mapas.

Con el uso de un mapa tomado de la naturaleza, se ha podido aumentar la seguridad sin perder la baja dimensionalidad. Además, la degradación del caos se puede verificar gracias al análisis cosmográfico para el mapa de la naturaleza presentado en esta tesis.

Como trabajo futuro, se pueden realizar de la seguridad de EDD de baja y alta dimensión para determinar, cuál es una dimensión óptima de trabajo, donde el compromiso seguridad-eficiencia no se vea maximizado, es decir, que la seguridad no se vea comprometida para hacer más eficiente la ejecución del esquema de cifrado.

También, al crear algoritmo con los requisitos criptográficos para esquemas caóticos, se puede pensar en la escalabilidad para el uso en aplicaciones de seguridad informática de este nuevo sistema de cifrado.

Por último, se pueden explorar las relaciones y diferencias entre este nuevo esquema de cifrado y RSA, para poder comparar en casos prácticos el comportamiento y rendimientos de ambos cifrados en condiciones similares de trabajo.

Bibliografía

[1] Singh, Simon: *The code book: the science of secrecy from ancient Egypt to quantum cryptography*. Random House LLC, 2011.

[2] Perlner, Ray A. y David A. Cooper: *Quantum resistant public key cryptography: a survey*. En *Proceedings of the 8th Symposium on Identity and Trust on the Internet*, páginas 85–93. ACM, 2009.

[3] Shor, Peter W.: *Algorithms for quantum computation: discrete logarithms and factoring*. En *Foundations of Computer Science, 1994 Proceedings., 35th Annual Symposium on*, páginas 124–134. IEEE, 1994.

[4] Smolin, John A. y Graeme Smith: *Classical signature of quantum annealing*. arXiv preprint arXiv:1305.4904, 2013.

[5] Bernstein, Daniel J., Johannes Buchmann y Erik Dahmen: *Post-quantum cryptography*. Springer, 2009.

[6] Deutsch, David: *Quantum theory, the Church-Turing principle and the universal quantum computer*. Proceedings of the Royal Society of London. A. Mathematical and Physical Sciences, 400(1818):97–117, 1985.

[7] Chander, Bhanu: *Quantum Cryptography Key Distribution: Quantum Computing*. En *Quantum Cryptography and the Future of Cyber Security*, páginas 84–108. IGI Global, 2020.

[8] Alagic, Gorjan, Gorjan Alagic, Jacob Alperin-Sheriff, Daniel Apon, David Cooper, Quynh Dang, Yi Kai Liu, Carl Miller, Dustin Moody, Rene Peralta *y cols.*: *Status report on the first round of the NIST post-quantum cryptography standardization process*. US Department of Commerce, National Institute of Standards and Technology, 2019.

[9] Morin, David: *Introduction to classical mechanics: with problems and solutions*. Cambridge University Press, 2008.

[10] Shannon, Claude E.: *Communication Theory of Secrecy Systems*. Bell system technical journal, 28(4):656–715, 1949.

[11] Alvarez, Gonzalo y Shujun Li: *Some basic cryptographic requirements for chaos-based cryptosystems*. International Journal of Bifurcation and Chaos, 16(08):2129–2151, 2006.

[12] Santos, L.F., G. Rigolin y C.O. Escobar: *Entanglement versus chaos in disordered spin chains*. Physical Review A, 69(4):042304, 2004.

[13] Lauter, Kristin: *How to Keep Your Secrets in a Post-Quantum World*. NOTICES OF THE AMERICAN MATHEMATICAL SOCIETY, 67(1), 2020.

[14] Rivest, Ronald L., Adi Shamir y Len Adleman: *A method for obtaining digital signatures and public-key cryptosystems*. Communications of the ACM, 21(2):120–126, 1978.

[15] Aggarwal, Divesh y Ueli Maurer: *Breaking RSA generically is equivalent to factoring*. En *Advances in Cryptology-EUROCRYPT 2009*, páginas 36–53. Springer, 2009.

[16] Ifrah, Georges, Edward Frank Harding, David Bellos, Sophie Wood *y cols.*: *The Universal History of Computing: From the Abacus to Quantum Computing*. John Wiley & Sons, Inc., 2000.

[17] Kocarev, Ljupco y Zarko Tasev: *Public-key encryption based on Chebyshev maps*. En *Circuits and Systems, 2003. ISCAS'03. Proceedings of the 2003 International Symposium on*, volumen 3, páginas III–28. IEEE, 2003.

[18] Stojanovski, Toni y Ljupco Kocarev: *Chaos-based random number generators-part I: analysis [cryptography]*. Circuits and Systems I: Fundamental Theory and Applications, IEEE Transactions on, 48(3):281–288, 2001.

[19] Pichler, Franz y Josef Scharinger: *Finite dimensional generalized baker dynamical systems for cryptographic applications*. En *Computer Aided Systems Theory EUROCAST'95*, páginas 465–476. Springer, 1996.

[20] Fridrich, Jiri: *Symmetric ciphers based on two-dimensional chaotic maps*. International journal of bifurcation and chaos, 8(06):1259–1284, 1998.

[21] Masuda, Naoki y Kazuyuki Aihara: *Cryptosystems with discretized chaotic maps*. Circuits and Systems I: Fundamental Theory and Applications, IEEE Transactions on, 49(1):28–40, 2002.

[22] Matthews, Robert: *On the derivation of a chaotic encryption algorithm*. Cryptologia, 13(1):29–42, 1989.

[23] Kocarev, Ljupco, Zarko Tasev y J. Makraduli: *Public-key encryption and digital-signature schemes using chaotic maps*. En *16th European Conference on Circuits Theory and Design, ECCTD*, volumen 3, 2003.

[24] Kocarev, Ljupco y Shiguo Lian: *Chaos-based cryptography*. Springer, 2011.

[25] Yan, Song Y.: *Quantum Attacks on Public-Key Cryptosystems*. Springer, 2013.

[26] Perlner, Ray A. y David A. Cooper: *Quantum resistant public key cryptography: a survey*. En *Proceedings of the 8th Symposium on Identity and Trust on the Internet*, páginas 85–93. ACM, 2009.

[27] Hoffstein, Jeffrey, Nick Howgrave-Graham, Jill Pipher, Joseph H. Silverman y William Whyte: *NTRUSIGN: Digital signatures using the NTRU lattice*. En *Topics in cryptology CT-RSA 2003*, páginas 122–140. Springer, 2003.

[28] Miller, Gary L.: *Riemann's hypothesis and tests for primality*. Journal of computer and system sciences, 13(3):300–317, 1976.

[29] Abramowitz, Milton y Irene A. Stegun: *Handbook of mathematical functions with formulas, graphs, and mathematical tables*, volumen 55. US Government printing office, 1948.

[30] Zhou, Jianqin, Jun Hu y Ping Chen: *Extended Euclid algorithm and its application in RSA*. En *The 2nd International Conference on Information Science and Engineering*, páginas 2079–2081. IEEE, 2010.

[31] Dingyi, Pei, Salomaa Arto y Ding Cunsheng: *Chinese remainder theorem: applications in computing, coding, cryptography*. World Scientific, 1996.

[32] Shor, Peter W.: *Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer*. SIAM journal on computing, 26(5):1484–1509, 1997.

[33] Turing, Allan M.: *On computable numbers, with an application to the entscheidungsproblem*. J. of Math, 58:345–363, 1938.

[34] Birkhoff, Garrett y John Von Neumann: *The logic of quantum mechanics*. Springer, 1975.

[35] De Raedt, Koen, Kristel Michielsen, Hans De Raedt, Binh Trieu, Guido Arnold, Marcus Richter, Th Lippert, H. Watanabe y N. Ito: *Massively parallel quantum computer simulator*. Computer Physics Communications, 176(2):121–136, 2007.

[36] Dickson, N.G., M.W. Johnson, M.H. Amin, R. Harris, F. Altomare, A.J. Berkley, P. Bunyk, J. Cai, E.M. Chapple, P. Chavez *y cols.*: *Thermally assisted quantum annealing of a 16-qubit problem*. Nature communications, 4:1903, 2013.

[37] Castelvecchi, D.: *How'spooky'is quantum physics? The answer could be incalculable.* Nature, 577(7791):461, 2020.

[38] Einstein, Albert, Boris Podolsky y Nathan Rosen: *Can quantum-mechanical description of physical reality be considered complete?* Physical review, 47(10):777, 1935.

[39] Aspect, Alain, Philippe Grangier y Gérard Roger: *Experimental realization of Einstein-Podolsky-Rosen-Bohm Gedankenexperiment: a new violation of Bell's inequalities*. Physical review letters, 49(2):91, 1982.

[40] Griffiths, David J. y Darrell F. Schroeter: *Introduction to quantum mechanics*. Cambridge University Press, 2018.

[41] Plenio, Martin B. y Shashank S. Virmani: *An introduction to entanglement theory*. En *Quantum Information and Coherence*, páginas 173–209. Springer, 2014.

[42] Goldstein, Herbert: *Classical mechanics*, volumen 4. Pearson Education India, 1962.

[43] Enns, Richard H.: *It's a nonlinear world*. Springer, 2010.

[44] Kolumban, G., M.P. Kennedy, G. Kis y Z. Jako: *FM-DCSK: A novel method for chaotic communications*. En *Circuits and Systems, 1998. ISCAS'98. Proceedings of the 1998 IEEE International Symposium on*, volumen 4, páginas 477–480. IEEE, 1998.

[45] Sushchik Jr, Mikhail, Nikolai Rulkov, Lawrence Larson, Lev Tsimring, Henry Abarbanel, Kung Yao y Alexander Volkovskii: *Chaotic pulse position modulation: A robust method of communicating with chaos*. Communications Letters, IEEE, 4(4):128–130, 2000.

[46] Kocarev, Ljupco: *Chaos-based cryptography: a brief overview*. Circuits and Systems Magazine, IEEE, 1(3):6–21, 2001.

[47] Kaneko, Kunihiko: *Theory and applications of coupled map lattices*, volumen 159. Wiley New York, 1993.

[48] Kaneko, Kunihiko: *Overview of coupled map lattices*. Chaos: An Interdisciplinary Journal of Nonlinear Science, 2(3):279–282, 1992.

[49] Tenny, Roy, Lev S. Tsimring, Larry Larson y Henry D.I. Abarbanel: *Using distributed nonlinear dynamics for public key encryption.* Physical review letters, 90(4):047903, 2003.

[50] Tenny, Roy, Lev S. Tsimring, Henry D. I. Abarbanel y Lawrence E. Larson: *Security of Chaos-Based Communication and Encryption.* En Larson, Lawrence E., Lev S. Tsimring y Jia Ming Liu (editores): *Digital Communications Using Chaos and Nonlinear Dynamics*, Institute for Nonlinear Science, páginas 191–229. Springer New York, 2006, ISBN 978-0-387-29787-3.

[51] Sun, Jinhui, Geng Zhao y Xufei Li: *An Improved Public Key Encryption Algorithm Based on Chebyshev Polynomials.* TELKOMNIKA Indonesian Journal of Electrical Engineering, 11(2):864–870, 2013.

[52] Teh, Je Sen, Moatsum Alawida y You Cheng Sii: *Implementation and practical problems of chaos-based cryptography revisited.* Journal of Information Security and Applications, 50:102421, 2020.

[53] Solís-Sánchez, Hugo y E. Gabriela Barrantes: *Using the Logistic Coupled Map for Public Key Cryptography under a Distributed Dynamics Encryption Scheme.* Information, 9(7):160, 2018.

[54] Menezes, Alfred J., Paul C. Van Oorschot y Scott A. Vanstone: *Handbook of applied cryptography.* CRC press, 2010.

[55] Wang, Xingang, Xiaofeng Gong, Meng Zhan y Choy Heng Lai: *Public-key encryption based on generalized synchronization of coupled map lattices.* Chaos: An Interdisciplinary Journal of Nonlinear Science, 15(2):023109, 2005.

[56] Kocarev, L., J. Makraduli y P. Amato: *Public-key encryption based on Chebyshev polynomials*. Circuits, Systems and Signal Processing, 24(5):497–517, 2005.

[57] Zhen, Ping, Geng Zhao, Lequan Min y Xiaodong Li: *A survey of chaos-based cryptography*. En *2014 Ninth International Conference on P2P, Parallel, Grid, Cloud and Internet Computing*, páginas 237–244. IEEE, 2014.

[58] Devaney, Robert: *An introduction to chaotic dynamical systems*. CRC Press, 2018.

[59] Arroyo, David, José María Amigó Garcia, Shujun Li y Gonzalo Alvarez: *On the inadequacy of unimodal maps for cryptographic applications*. 2010.

[60] Lawnik, M.: *Generalized logistic map and its application in chaos based cryptography*. 936(1):012017, 2017.

[61] Kaneko, Kunihiko: *Theory and applications of coupled map lattices*. Nonlinear science: theory and applications, 1993.

[62] Lloyd, Alun L.: *The coupled logistic map: a simple model for the effects of spatial heterogeneity on population dynamics*. Journal of Theoretical Biology, 173(3):217–230, 1995.

[63] Schult, R.L., Dennis B. Creamer, F.S. Henyey y J.A. Wright: *Symmetric and nonsymmetric coupled logistic maps*. Physical Review A, 35(7):3115, 1987.

[64] Zhang, Ying Qian y Xing Yuan Wang: *Spatiotemporal chaos in Arnold coupled logistic map lattice*. Nonlinear Anal. Model. Control, 18(4):526–541, 2013.

[65] Tenny, Roy, Lev S. Tsimring, Larry Larson y Henry D.I. Abarbanel: *Using distributed nonlinear dynamics for public key encryption*. Physical review letters, 90(4):047903, 2003.

[66] Xiao, Di, Xiaofeng Liao y Shaojiang Deng: *A novel key agreement protocol based on chaotic maps*. Information Sciences, 177(4):1136–1142, 2007.

[67] Idris, Ivan: *NumPy Cookbook*. Packt Publishing Ltd, 2015.

[68] Solis, Hugo: *Kivy cookbook*. Packt Publishing Ltd, 2015.

[69] Tenny, Roy, Lev S. Tsimring, Henry D.I. Abarbanel y Lawrence E. Larson: *Security of chaos-based communication and encryption*. En *Digital Communications Using Chaos and Nonlinear Dynamics*, páginas 191–229. Springer, 2006.

[70] Elhadj, Zeraoulia y J.C. Sprott: *Chaotifying 2-D piecewise-linear maps via a piecewise-linear controller function*. Nonlinear Oscillations, 13(3):352, 2011.

[71] Ortega-Rodríguez, Manuel, Hugo Solís-Sánchez, Vanessa López-Barquero, Bryan Matamoros-Alvarado y Ariadna Venegas-Li: *The 2: 3: 6 quasi-periodic oscillation structure in GRS 1915+ 105 and cubic subharmonics in the context of relativistic discoseismology*. Monthly Notices of the Royal Astronomical Society, 440(4):3011–3015, 2014.

[72] Li, Shujun, Xuanqin Mou, Yuanlong Cai, Zhen Ji y Jihong Zhang: *On the security of a chaotic encryption scheme: problems with computerized chaos in finite computing precision*. Computer physics communications, 153(1):52–58, 2003.

[73] Kürten, Karl E. y Grégoire Nicolis: *Bifurcation scenarios and quasiperiodicity in coupled maps*. Physica A: Statistical Mechanics and its Applications, 245(3-4):446–452, 1997.

[74] Wald, Robert M.: *Black hole entropy is the Noether charge*. Physical Review D, 48(8):R3427, 1993.

[75] Landau, L.D. y E.M. Lifshitz: *Course of theoretical physics. vol. 1: Mechanics.* Oxford, 1994.

[76] Visser, Matt: *Jerk, snap and the cosmological equation of state.* Classical and Quantum Gravity, 21(11):2603, 2004.

[77] Klis, Michiel Van der: *Compact stellar X-ray sources.* Cambridge University Press, 2006.

[78] Solís-Sánchez, Hugo y Elena Gabriela Barrantes: *Using Coupled Maps from the Nature for PKC.* En *WiP, 30st Annual Computer Security Applications Conference (ACSAC)*, 2014.

[79] Solís-Sánchez, Hugo, Manuel Ortega-Rodríguez, Luis A. Álvares, Esteban Dodero, E. Gabriela Barrantes y José M. Gamboa: *Cosmographic Analysis as framework to evaluate cosmological models.* En *Proceedings of the Fifteenth Marcel Grossman Meeting on General Relativity.* World Scientific, 2019.

[80] Neves, Samuel y Filipe Araujo: *On the performance of GPU public-key cryptography.* En *ASAP 2011-22nd IEEE International Conference on Application-specific Systems, Architectures and Processors*, páginas 133–140. IEEE, 2011.

[81] Solís-Sánchez, Hugo y Elena Gabriela Barrantes: *Implementación del Algoritmo de Descifrado para un Esquema Criptográfico de Dinámica Distribuida.* En *Memorias de congresos TEC*, 2017.

Anexo

Artículos científicos aceptados como requisito de graduación del doctorado.

Implementación del Algoritmo de Descifrado para un Esquema Criptográfico de Dinámica Distribuida

Hugo Solís-Sánchez
Centro de Investigaciones en Tecnologías de la Información y Comunicación y Escuela de Física
Universidad de Costa Rica
11501-2060 San José, Costa Rica
hugo.solis@ucr.ac.cr

Elena Gabriela Barrantes
Centro de Investigaciones en Tecnologías de la Información y Comunicación y Escuela de Computación e Informática
Universidad de Costa Rica
11501-2060 San José, Costa Rica
gabriela.barrantes@ecci.ucr.ac.cr

***Abstract*— Este trabajo ha adaptado el Cifrado de Dinámica Distribuida (CDD) a un sistema caótico de baja dimensión para evaluar la debilidad y seguridad de dicha adaptación en un ejemplo realista. Esto debido a que varios autores apuntan a una deficiencia del CDD en el proceso de descifrado. En concreto, se utilizó un mapa logístico acoplado, el cual presenta múltiples atractores caóticos hecho que elimina la deficiencia del mapa logístico simple manifiesta para las aplicaciones criptográficas. Además se exploran optimizaciones al algoritmo de descifrado llegando a rendimientos apreciables para su futura escalabilidad.**

Keywords—descifrado, criptografía, llave pública, caos, dinámica distribuida, mapa logístico, mapa logístico acoplado

I. Introducción

Los sistemas caóticos tienen grandes aplicaciones potenciales en el cifrado de la información. Los sistemas criptográficos se clasifican en dos ramas: llave privada y llave pública [1]. Se han hecho múltiples esfuerzo desde las aplicaciones del caos en la parte de llave privada en comparación con la otra rama [2].

El mapa logístico es por excelencia el ejemplo de un sistema caótico. Originalmente formulado para representar un modelo demográfico simple con el fin de explicar el aumento de una población, el mapa logístico es un mapa unimodal y a consecuencia de esto su dinámica es limitada [3]. Puede expresarse usando la ecuación:

$$f(x) = \mu x(1 - x) \tag{1}$$

Donde el parámetro μ está en el intervalo $0 \leqslant \mu \leqslant 4$. El aspecto unimodal del mapa logístico hace que sea inadecuado para las aplicaciones criptográficas, aunque se han creado un número razonable de aplicaciones basados en éste [4].

Un mapa de retícula acoplado (MRA) es un sistema dinámico que modela el comportamiento de sistemas no lineales. Se utilizan principalmente para estudiar cualitativamente la dinámica caótica de los sistemas espacialmente extendidos. Esto incluye la dinámica del caos espaciotemporal donde el número de grados de libertad efectivos diverge a medida que aumenta el tamaño del sistema. Los MRA incorporan un sistema de ecuaciones (acopladas o desacopladas), un número finito de variables, un esquema de acoplamiento global o local y los términos de acoplamiento correspondientes [5].

El mapa logístico acoplado es uno de los más simples MRA, considerado en sus inicios como un modelo sencillo biológicamente realista que incorpora efectos espaciales, se basa en dos mapas logísticos acoplados por un acoplamiento lineal:

$$x_{n+1} = f(x_n) + \alpha(y_n - x_n) \tag{2}$$

$$y_{n+1} = f(y_n) + \alpha(y_n - x_n), \tag{3}$$

Donde $f(x)$ es el mapa logístico de la ecuación (1) y α es un parámetro de acoplamiento. En el mapa logístico sólo se observan dos rutas al caos (duplicación del período e intermitencia), la segunda dimensión del mapa acoplado logístico permite que la ruta cuasiperiódica ocurra [6]. Los atractores caóticos múltiples presentes en este sistema eliminan el defecto criptográfico del mapa logístico simple. Las figuras 1 y 3 muestran ejemplos de atractores caóticos para el mapa logístico acoplado (MLA).

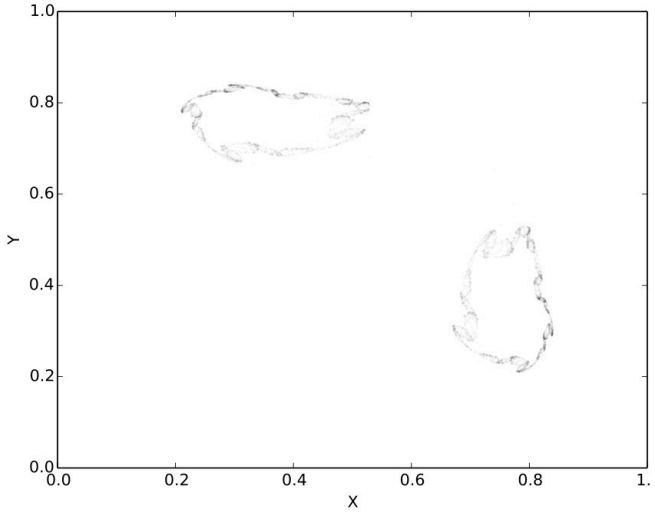


Fig 1. Atractor caótico para el mapa logístico acoplado para $\mu = 2.9$ $\alpha = 0.3314$

Este trabajo ha sido parcialmente financiado por MICITT y CONICIT FI-0204-2012 y por 834-B5-293 de la Universidad de Costa Rica por Vicerrectoría de Investigación.

Un grupo de la University of California San Diego introdujo un esquema teórico para el cifrado asimétrico que explota las propiedades de los sistemas dinámicos no lineales donde un sistema dinámico no lineal disipativo de alta dimensionalidad se distribuye entre un transmisor y un receptor, por esto se llama al método, cifrado de dinámica distribuida (CDD). La dinámica del transmisor es pública y la dinámica del receptor está oculta, no se comparte en el canal de comunicación. Un mensaje es codificado por modulación en los parámetros del transmisor, y esto resulta en un cambio del atractor general del sistema. Un receptor no autorizado no conoce la dinámica oculta en el receptor y no puede decodificar el mensaje [7].

Este trabajo tomará la propuesta teórica del CDD y la adaptada por primera vez a un sistema de baja dimensión (el MLA). Se estudiará el cifrado, el descifrado y el ataque común para este nuevo sistema criptográfico.

II. Cifrado

La idea básica del CDD es dividir un sistema dinámico de dimensión $D_T + D_R$ en dos partes con D_T variables de transmisor $t(n) = [t_1 (n); \ldots ; t_{D_T}(n)]$, y las D_R variables del receptor de $r(n) = [r_1 (n); \ldots ; r_{D_R}(n)]$. El receptor recibe la señal escalar $s_t (n)$ desde el transmisor, y el transmisor recibe la señal escalar $s_r (n)$ desde el receptor.

$$t(n+1) = F_T(t(n), s_r(n), m(n)) \quad (4)$$

$$r(n+1) = F_R(r(n), s_t(n), m(n)) \quad (5)$$

Donde $m(n)$ es el mensaje que queremos cifrar. Permitimos que $m(n)$ sólo tome valores 0 o 1, esto crea un mensaje binario.

El cifrado para nuestra implementación de baja dimensión viene de una relación entre las ecuaciones (2), (3) y (4), (5), donde ahora x (ecuación 2) será la dinámica dividida al transmisor y y (ecuación 3) será la parte del receptor, donde se sigue:

$$x_{n+1} = f(x_n) + \alpha(y_n - x_n) + A*m \quad (6)$$

$$y_{n+1} = f(y_n) + \alpha(y_n - x_n) \quad (7)$$

el parámetro A es una modulación del mensaje, en nuestra implementación A toma valores aleatorios entre 0,001 y 0,01 para proporcionar una seguridad adicional al sistema. La figura 2 muestra un mensaje de 8 bits encriptado (una letra, en este caso la W), para una fácil identificación diferenciamos los puntos que corresponden al bit 0 a los que corresponden al bit 1. La seguridad de esta implementación reside en la superposición y cercanía de esos puntos. Sólo si tiene la simulación anterior se puede descifrar el mensaje. La Figura 4 muestra un atractor diferente con un mensaje más largo de 32 bits (cuatro letras, en este caso Wort). En este esquema, un atractor caótico diferente representa un par diferente de claves criptográficas. La ecuación 2 con sus parámetros correspondientes es la llave pública y la ecuación 3 es la llave privada.

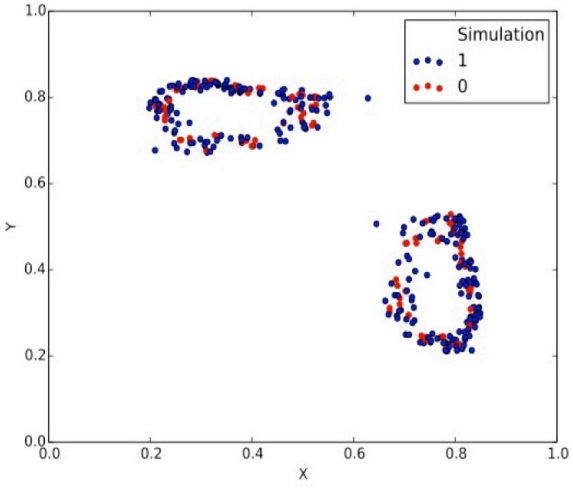


Fig 2. Mensaje cifrado sobre la dinámica totalmente conocida de la Fig 1.

III. Descifrado

Un receptor autorizado conoce todas las cantidades cifrado propuesto en la sección anterior (públicas y privadas) y puede establecer fuera de línea (antes de realizar la comunicación) los atractores permitidos, u otros aspectos dinámicos del sistema total, para todos los valores permitidos de $m(n)$. Para realizar el descifrado es necesario conocer previamente todos los puntos de la simulación. El proceso de descifrado corresponde al cálculo de la distancia euclídea de cada punto recibido a los puntos de la simulación. Si ésta distancia es menor que un parámetro de tolerancia, que está relacionado con el parámetro A de la ecuación (6) es un bit 0 y si es mayor es un bit 1.

Dicho procedimiento aunque fácil de describir es computacionalmente costoso como se muestra en la sección V de este trabajo y es donde reside la debilidad expuesta por varios autores como en [8]. En este estudio exploramos tres

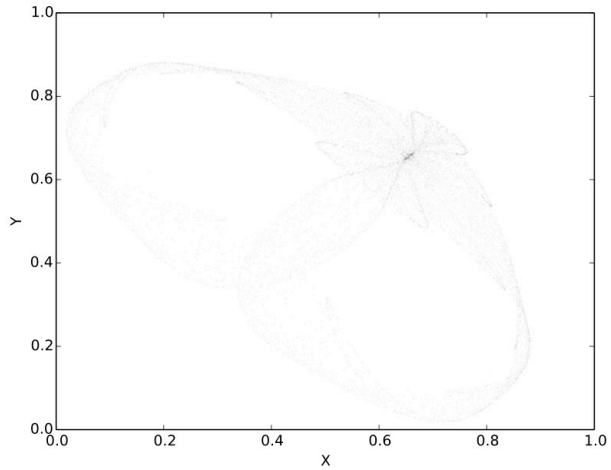


Fig 3. Otro atractor caótico para el mapa logístico acoplado para $\mu = 2.9$ $\alpha = 0.414$

formas distintas de abordar y mejorar la problemática, dicha variantes descritas a continuación.

A. Numpy

Haciendo uso de las posibilidades de Numpy de hacer cálculos de algebra lineal [9]. Se puede computar directamente la distancia euclídia sobre el arreglo de datos que contiene a los generados por la simulación.

B. GPU

La unidad de procesamiento gráfico (GPU, por sus siglas en ingles) se encuentra optimizada para hacer los cálculos de algebra lineal y ha sido de mucha ayuda para aplicaciones donde existen extensivos cálculos numéricos. Con ayuda del paquete Kivy para Python es posible sacar ventaja del GPU a través de OPENGL [10]. Con el uso de vectores se pueden computar la distancia del mismo del caso anterior con ayuda del GPU tiene un desempeño considerable para cálculos matemáticos.

C. Nuestra Propuesta

En este caso es posible repensar el algoritmo de descifrado, si los datos para x mientras se crean en la simulación se van incluyendo de forma ordenada en el arreglo de datos con su respectivo y, cuando se debe calcular la distancia euclídea se puede hacer para una ventana de cercanía de unos cientos datos sin necesidad de buscar en todo el arreglo. El tamaño de la ventana es determinada por el valor del parámetro A.

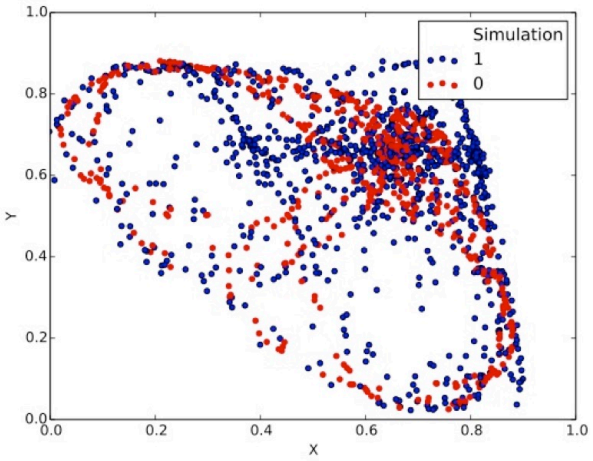


Fig 4. Mensaje cifrado sobre la dinámica totalmente conocida de la Fig 3.

IV. Criptoanálisis

Un receptor no autorizado puede intentar varios métodos para atacar CDD y descodificar el mensaje secreto $m(n)$. Uno de estos métodos consiste en reconstruir las posiciones de los atractores que corresponden a la transmisión de 0 y 1 almacenando y agrupando muestras de muchos bits transmitidos. Conocer las posiciones de los atractores permitiría al receptor no autorizado descodificar el mensaje utilizando el mismo método que el receptor autorizado [11].

La dinámica caótica puede contener ruido en el canal y o un circuito externo que harían que la dinámica sea estocástica. Un receptor no autorizado puede intentar generar un Modelo de Markov Oculto de la dinámica pública del transmisor para cada posible valor del mensaje m, y obtener una estimación de Máxima Verosimilitud (ML) del mensaje m. El mensaje decodificado será dado entonces por

$$m' = max_{m=(0,1)} p(s_t(1), \ldots, s_t(D_T+1)|m) \quad (8)$$

Para generar el modelo de Markov oculto, el receptor no autorizado necesitará cuantificar el estado del transmisor en un espacio de inserción reconstruido a partir de retardos en el tiempo y estimar las probabilidades de transición de estado así como las probabilidades de observación del modelo [8].

El ataque propuesto fue implementado para nuestro modelo de baja dimensionalidad. La figura 5 muestra cómo el caso de un mensaje cifrado usando un mapa no caótico donde la exactitud del entrenamiento es de 97% esto muestra la efectividad del ataque. Cuando el número de bits es menor que el necesario para entrenar el modelo según [7] la precisión se reduce a menos del 40% haciendo que ni aún el ataque probabilístico como se muestra en [8] pueda ayudar a este ataque en particular. La figura 6 muestra el caso en el que el número de bits es igual a necesario para realizar el ataque, es visible cómo la curva de decisión puede resolver para 0 o 1 bits.

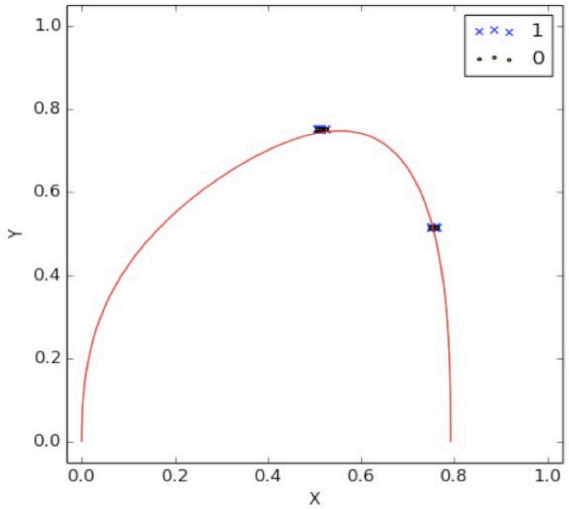


Fig 5. Superficie de decisión obtenida a partir de ataque cuando el atractor no es caótico.

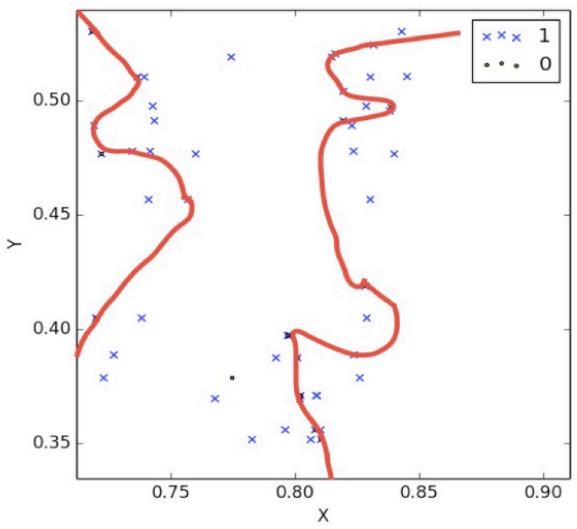


Fig 6. Superficie de decisión obtenida a partir de ataque cuando el atractor es el mismo de la Fig 1.

V. Experimento computacional

La implementación propuesta en este trabajo se ha hecho en Python con la ayuda del paquete Numpy para la gestión de datos y HMMLearn para la cadena de Markov utilizada en el ataque. Los cálculos se realizaron en un sistema Linux con un Core i7 2.6GHz PC con memoria RAM de 16 GB y gráficos por un Radeon Pro 450 con 2GB de memoria GDDR5. Las cifras de las Figuras 1 a 3 se calcularon con un millón de puntos para el atractor caótico con un tiempo medio de 33 s para el cálculo. La Figura 6 tiene el mayor tiempo de cálculo, en este caso, el entrenamiento de la cadena de Markov tomó un tiempo promedio de 72 horas debido a que se necesitan 5 millones de bits para el entrenamiento.

TABLA I. Detalles del tiempo de Cifrado/Decifrado

N bits	Cifrado tiempo (s)	Descifrado tiempo (s)	Numpy tiempo (s)	GPU tiempo (s)	Propuesta tiempo (s)
8	0.045	141.1081	70.5541	2.0990	3.0212
16	0.0997	293.1638	146.5819	4.1330	6.0222
32	0.2017	600.4974	300.2487	8.1670	9.0232
64	0.3199	1318.4119	659.2060	16.2010	16.0242
128	0.5973	2567.0675	1283.533	32.2350	32.0252
256	1.7431	5297.5988	2648.799	60.2690	60.0262
512	2.8889	10028.1301	5014.065	88.3030	89.0272
1024	4.0347	14758.6614	7379.330	116.337	117.361
2048	5.1805	19489.1927	9744.596	144.371	145.862
4096	6.3263	24219.7241	12109.86	172.405	174.363

La tabla 1 muestra la realización de los algoritmos de cifrado y descifrado donde como se espera el tiempo de ambos incrementos sobre más bits necesitan ser procesados. Es notable cómo el tiempo de cifrado es relativamente pequeño comparado con el tiempo de descifrado. La columna denominada Numpy presenta el tiempo de descifrado mejorando el manejo de los datos el computo de las operaciones, dicho aspecto aunque optimizan el tiempo de ejecución no lo hacen considerablemente, en cambio se los cálculos son hecho en el GPU como se muestran en la columna siguiente si se presenta una mejoría que hace al algoritmo de descifrado práctico para se escalamiento al uso de aplicaciones criptográficas, aunque el uso de GPU puede en si mismo presentar debilidades a la seguridad informática [12], además de que no todas las plataforma cuentan con uno.

Nuestra modificación al algoritmo mostrada en la última columna representa una clara ventaja pues su desempeño es similar al uso del GPU, si la necesidad de paquetes ni requerimientos adicionales para su implementación.

VI. Conclusiones

En este trabajo, se realizó una implementación completa del CDD, donde se describió los procesos de cifrado y descifrado, mostrando cómo la implementación de CDD es posible con un sistema dinámico de baja dimensión. Esto es relevante para este campo de investigación porque proporciona un ejemplo funcional para CDD con el mismo tipo de seguridad proporcionada por los sistemas de alta dimensionalidad. Además, este tipo de implementación abre la posibilidad de investigar mejores formas de fortalecer la eficiencia del CDD.

En la sección anterior se mostró como la mejora propuesta por este trabajo al algoritmo teórico de CDD debate las críticas de la eficiencia detrás del descifrado abriendo paso para uso en situaciones reales.

Referencias

[1] A. Menezes, P. van Oorschot, and S. Vanstone, Handbook of Applied Cryptography. CRC: Boca Raton, 1997.

[2] X. Wang, X. Gong, M. Zhan, and C. H. Lai, "Public-key encryption based on generalized synchronization of coupled map lattices" Chaos: An Interdisciplinary Journal of Nonlinear Science 15, 023109, 2005.

[3] R. L. Devaney, An introduction to chaotic dynamical systems, volume 13046. Addison-Wesley Reading,1989.

[4] D. Arroyo, J. M. Amigo, Garcia, S. Li, and G. Alvarez, "On the inadequacy of unimodal maps for cryptographic applications", RECSI, Spain, 2010.

[5] K. Kaneko, Theory and applications of coupled map lattices, volume 159. Wiley: New York, 1993.

[6] A. L. Lloyd, "The coupled logistic map" Journal of Theoretical Biology 173, 217, 1995.

[7] R. Tenny, L. S. Tsimring, L. Larson, and H. D. Abarbanel, "Using Distributed Nonlinear Dynamics for Public Key Encryption" Physical Review Letters 90, 047903, 2003.

[8] D. Xiao, X. Liao, and S. Deng, "A novel key agreement protocol based on chaotic maps" Information Sciences 177, 1136, 2007.

[9] I. Idris, NumPy Cookbook. Packt Publishing Ltd: Birmingham, 2012.

[10] H. Solis, Kivy Cookbook. Packt Publishing Ltd: Birmingham, 2015.

[11] R. Tenny, L. Tsimring, H. Abarbanel, and L. Larson, "Security of chaos-based communication and encryption", in Digital Communications Using Chaos and Nonlinear Dynamics, edited by L. Larson, L. Tsimring, and J.-M. Liu, pages 191–229, (Springer, New York, 2006).

[12] S. Neves and F. Araujo, "On the performance of GPU public-key cryptography". InApplication-Specific Systems, Architectures and Processors (ASAP), 2011 IEEE International Conference on 2011 Sep 11 (pp. 133-140). IEEE.

Cosmographic Analysis as framework to evaluate cosmological models

Hugo Solís-Sánchez*[12], Manuel Ortega-Rodríguez[1], Luis A. Álvares[1], Esteban Dodero[1], E. Gabriela Barrantes[2], and José M. Gamboa[1]

[1]*Escuela de Física and* [2]*Centro de Investigaciones en Tecnologías de la Información y Comunicación, Universidad de Costa Rica, San José, 11501-2060, Costa Rica*

**E-mail: hugo.solis@ucr.ac.cr*

www.ucr.ac.cr

By using a cosmographic analysis of the redshift data of type Ia supernovae, we are able to get the expansion of the scale factor, obtaining the current values of the Hubble, deceleration, jerk and snap parameters. Our data is then used to compare the fitness of various proposed alternative cosmological models. Since our method assumes only the validity of general relativity at the cosmic scale, along with the isotropy and homogeneity of the universe, they are very useful for comparison between different cosmological models, including the fitness of ΛCDM model. Our method is based on the order expansion of the scale factor present in the FRW metric and using a Monte Carlo integration to find the best fit order parameters of the scale factor to reproduce the observed data, we make use of parallel paradigm to improve the computational time behind the model. We find the known result an accelerated expansion of the universe. With access to better measurements of type Ia supernovae redshifts and more data, the cosmographic results will be significantly improved.

Keywords: Cosmography; Type Ia Supernovae; ΛCDM; deceleration parameter; Monte Carlo Integration

1. Introduction

Although there is almost complete agreement on the accelerated expansion of the universe[1], ΛCDM being the most favored cosmological model by the current data, there is no consensus to the current specific value of this decelerated expansion and to whether or not this model is still prevalent in the future[2]. The decelerated expansion q_0 is the parameter indicating the current rate of acceleration of the expansion of the universe, therefore, it is the first indicator of the fitness of any cosmological model, so a precise determination of its value is necessary for the diagnostic of any cosmological model. There recently have been plenty of research but there is no consensus for its current value[3].

We obtain the expansion terms of the scale factor in the FRW line element to test different cosmological models. We compare the expected value of the apparent magnitude in terms of the measured redshift to each supernova with the measured value of its apparent magnitude. This comparison is made by means of a likelihood ratio test to find the current values of the coefficients in the expansion of the factor scale that best fit the measured data.

Our interest is in comparing the results from our analysic with proposed cosmological models (as in ref.4 and ref.5). Even though cosmographic analysis has been previously studied (as in ref.6 and ref.7), we are performing the analysis using parallel computing what permits to make the expansion to higher orders than the

usuals. We found that some results have significant changes when calculations are improved and how the cosmography works as a framework to assess cosmological models.

2. Friedman-Robertson-Walker and Cosmography

2.1. *Friedman-Robertson-Walker*

The line element for all the homogeneous and isotropic models of the universe is the Friedman-Robertson-Walker metric, shown in 1. Where $a(t)$ is the expansion factor of the universe, which gives us the rate at which the universe is expanding. This value depends on the content and matter-energy densities for the universe and it is theoretically found using the Friedman equation, which is obtained from the Einstein's field equations. The constant k is only determined by its sign, if $k < 0$ the universe is said to be open and the spatial hypersurfaces have negative constant curvature, if $k > 0$ the universe is said to be closed and spatial hypersurfaces have positive constant curvature, for $k = 0$ the universe is said to be flat with the spatial hypersurfaces being Euclidean with curved spacetimes.

$$ds^2 = c^2dt^2 - a(t)^2[\frac{dr^2}{(1-kr^2)} + r^2(d\theta^2 + \sin^2\theta d\phi^2)] \tag{1}$$

2.2. *Cosmographic Analysis*

We proceed by using a cosmographic approach to determine the value of cosmological parameters. We seek to make a comparison of the measured values of the apparent magnitud (m) of the supernovae with the expected values given its measured redshift (z). The apparent magnitud is given by (2) in terms of the luminosity distance (d_L) and the absolute magnitude (M) which is known for supernovae to constant. Since the luminosity distance (3) is given in terms of the physical distance (r_0) between the source signal and the observer and the measured z we need to express r_0 in terms of the measured redshift. We do this by using the null geodesic in FRW and the cosmographic redshift. By the null geodesic we have:

$$m = 5\log\frac{d_L}{10} + M \tag{2}$$

$$d_L = (1+z)r_0a_0 \tag{3}$$

$$-c\int_{t_*}^{t_0}\frac{dt}{R(t)} = f(r_0) = \int_{r_0}^{0}\frac{dr}{\sqrt{(1-kr^2)}} \tag{4}$$

$$\text{Where: } f(r_0) = \begin{cases} -\sin^{-1}(r_0) & (k=+1) \\ -r_0 & (k=0) \\ -\sinh^{-1}(r_0) & (k=-1) \end{cases}$$

We expand the scale factor in FRW:

$$R(t) = R(t_0)[1+H_0(t-t_0)-\frac{1}{2!}q_0H_0^2(t-t_0)^2+\frac{1}{3!}j_0H_0^3(t-t_0)^3+\frac{1}{4!}s_0H_0^4(t-t_0)^4+...]$$

and using the cosmological redshift relation with the scale factor expansion[8], we obtain the flight time from the source to us ($T \equiv t_0 - t_*$ where t_* is the time at which the signal was emitted) as a function of the measured redshift.

$$z+1 = \frac{R(t_0)}{R(t_*)}$$

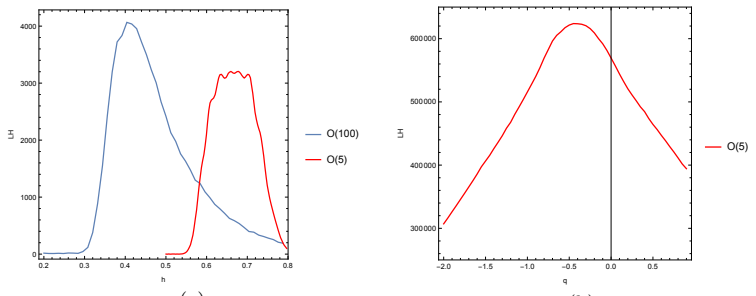
$$\frac{R(t_0)}{R(t_*)} = 1+H_0T+\frac{2+q_0}{2}H_0^2T^2+\frac{6(1+q_0)+j_0}{6}H_0^3T^3+\frac{24-s_0+8j_0+36q_0+6q_0^2}{24}H_0^4T^4+...$$

Numerically inverting:

$$T\left(\frac{z}{H_0}\right)^{-1} = 1-\left[1+\frac{q_0}{2}\right]z+\left[1+q_0+\frac{q_0^2}{2}-\frac{j_0}{6}\right]z^2$$

$$-\left[1+\frac{3}{2}q_0(1+q_0)+\frac{5}{8}q_0^3-\frac{1}{2}j_0-\frac{5}{12}q_0j_0-\frac{s_0}{24}\right]z^3+...$$

We solve the left side integral in eq. 4 with the expansion of the scale factor and substitute T in terms of z as found above. Therefore we are able to use the luminosity distance in terms of the redshift, to which we then employ a marginal likelihood ratio analysis in order to find the best fit values for the cosmological parameters. We are then able to use this data (from ref.6) to compare with any cosmological model.



(a) (b)

Fig. 1. Here are the results of our cosmographic analysis. (a)The Hubble parameter (b) The deceleration parameter.

In figure 1 (a), we are showing the results for the Hubble parameter for two orders of expansion. It is visible how when the order is higher the average is lower than the usual value, for 5 order we found $\langle H \rangle = 100\,\langle h_0 \rangle = 67$ and for 100 order is $\langle H \rangle = 47$. The figure 1 (b) shows the deceleration parameter which in agreement with recent results is negative[3] $\langle q_0 \rangle = -0.48$.

3. Comparison of different models

We compare our obtained factor scale with two different theoretically proposed cosmological models. The first one[4] proposes the gravitational constant (G) and the cosmological constant (Λ) are not constants but instead functions of time. With the standard Friedman equations derivation and proposing they relate to each other by $G\rho = \frac{\eta\Lambda}{8\pi}$ with ρ the density of the perfect fluid and η a constant, there are two possible scale factors and deceleration parameters:

- Case A: $n \neq 0$

$$a(t) = (nlt + C_1)^{\frac{1}{n}} \tag{5}$$

$$q = n - 1 \tag{6}$$

- Case B: $n = 0$

$$a(t) = C_2 e^{lt} \tag{7}$$

$$q = -1 \tag{8}$$

The second one[5] proposes measuring the average expansion rate in a universe in which a set of spherically symmetric sub-regions expand in an accelerated way, *Average Expansion Rate Approximation (AvERA)*. It has the appeal that it conclusively resolves the tension between the measured values of the *Hubble constant* but the great drawback is that it is difficult to match with the homogeneity observed in the CMB.

We plot them in figure (2) with our obtained results to see how they compare to each other. We see that although one has several parameters to adjust to get greater similarity with our results can be more closely approximated by Case A of the given model, Case B is an exponential, so it starts at the value one and never approximates the rest of the curves.

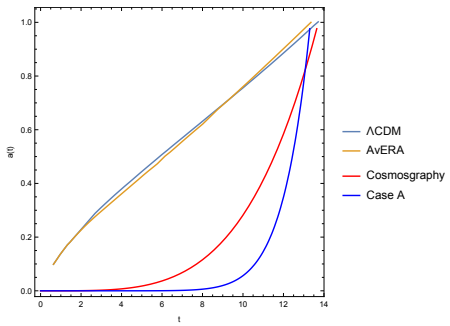


Fig. 2. Comparison of different models with the cosmographic result.

Acknowledgments

This work is partially supported by MICITT and CONICIT grant FI-0204-2012 and by grant 112-B8-107 of the Universidad de Costa Rica's Vicerrectoría de Investigación.

References

1. A. G. Riess, S. A. Rodney, D. M. Scolnic et Al. *ApJ* **853**, 126 (2018).
2. R. G. Cai and Z. L. Tuo. *Phys. Lett. B* **706**, 116 (2011).
3. I. Tutusaus, B. Lamine, A. Dupays and A. Blanchard . *A&A* **602**, A73 (2017).
4. S. K. Tripathi and R. K. Dubey. *Indian J. Sci. Res.* **2**, 95 (2011).
5. G. Rcz, L. Dobos, R. Beck, I. Szapudi and I. Csabai. *MNRAS: Letters* **469**, L1-L5 (2017).
6. M. V. John. *ApJ* **614**, 1 (2004).
7. S. Capozziello, V. F. Cardone, and V. Salzano. *Phys. Rev. D* **78**, 063504 (2008).
8. M. Visser. *Class. Quantum Grav* **21**, 2603 (2004).

information

MDPI

Article

Using the Logistic Coupled Map for Public Key Cryptography under a Distributed Dynamics Encryption Scheme

Hugo Solís-Sánchez [1,*] and E. Gabriela Barrantes [2]

[1] Centro de Investigaciones en Tecnologías de la Información y Comunicación and Escuela de Física, Universidad de Costa Rica, 11501-2060 San Jose, Costa Rica

[2] Centro de Investigaciones en Tecnologías de la Información y Comunicación and Escuela de Computación e Informaática, Universidad de Costa Rica, 11501-2060 San Jose, Costa Rica; gabriela.barrantes@ecci.ucr.ac.cr

* Correspondence: hugo.solis@ucr.ac.cr; Tel.: +506-2511-3037

Received: 14 May 2018; Accepted: 29 June 2018; Published: 2 July 2018

Abstract: Nowadays, there is a high necessity to create new and robust cryptosystems. Dynamical systems have promised to develop crypto-systems due to the close relationship between them and the cryptographic requirements. Distributed dynamic encryption (DDE) represents the first mathematical method to generate a public-key cryptosystem based on chaotic dynamics. However, it has been described that the DDE proposal has a weak point in the decryption process related to efficiency and practicality. In this work, we adapted the DDE to a low-dimensional chaotic system to evaluate the weakness and security of the adaption in a realistic example. Specifically, we used a non-symmetric logistic coupled map, which is known to have multiple chaotic attractors improving the shortcomings related to the simple logistic map that manifests its inadequacy for cryptographic applications. We found a full implementation with acceptable computational cost and speed for DDE, which it is essential because it provides a key cryptographic requirement for chaos-based cryptosystems.

Keywords: public key encryption; cryptography; chaos; chaotic cryptography; logistic map; logistic coupled map

1. Introduction

Chaotic systems have great potential to be applied on the encryption of information. Crypto-systems are classified into branches: private-key and public-key [1]. In chaos-based systems, a great deal of effort has been done in the private-key part in comparison with the public [2]. One of the most important public-key chaos-based cryptosystem is presented in [3] where Chebyshev maps are used to encrypt, but its efficiency is still lower than RSA [4], a common weak point for this kind of crypto-proposal.

The logistic map is an excellent example of a chaotic system. Originally formulated to represent a simple demographic model to explain the increase of a population, the logistic map is a one-dimensional unimodal map and, as a result, its dynamics are quite limited [5]. It can be expressed by using the equation:

$$f(x) = \mu x(1 - x) \tag{1}$$

where parameter μ is in the interval $0 \leq \mu \leq 4$. The unimodal aspect of the logistic map makes it inadequate for cryptographic applications because the parameter μ can be reconstructed from initial conditions, as in [6], even though a reasonable number of applications have been created [6]. A new relevant study [7] has been conducted to improve the logistic map for cryptographic applications, but losing the mathematical simplicity of Equation (1).

A coupled map lattice (CML) is a dynamical system that models the behavior of non-linear systems. They are predominantly used to qualitatively study the chaotic dynamics of spatially-extended systems. This includes the dynamics of spatiotemporal chaos where the number of effective degrees of freedom diverges as the size of the system increases. CML incorporates a system of equations (coupled or uncoupled), a finite number of variables, a global or local coupling scheme, and the corresponding coupling terms [8].

The logistic coupled map is one of the simplest CMLs, first considered as the simplest biologically realistic model that incorporates spatial effects, it is based on two coupled logistic maps by a linear coupling:

$$x_{n+1} = f(x_n) + \alpha(y_n - x_n) \tag{2}$$

$$y_{n+1} = f(y_n) - \alpha(y_n - x_n) \tag{3}$$

where $f(x)$ is the logistic map of Equation (1) and α is a coupling parameter. In the logistic map only two routes to chaos are observed (period doubling and intermittency), and the second dimension of the logistic coupled map allows the quasiperiodic route to occur [9]. The non-symmetric case of the logistic coupled map [10] occurs when, in Equations (2) and (3), we use a different parameter μ for $f(x)$, so the equations take the form:

$$x_{n+1} = f_1(x_n) + \alpha(y_n - x_n) \tag{4}$$

$$y_{n+1} = f_2(y_n) - \alpha(y_n - x_n) \tag{5}$$

where $f_1(x)$ means to use the logistic map of Equation (1) with μ_1 and $f_2(x)$ with μ_2. Multiple chaotic attractors are observed in this system improving the unimodal shortcoming of the simple logistic map [8]. Figures 1 and 2 show examples of chaotic attractors for the non-symmetric logistic coupled map (NLCM). The NLCM has a well documented chaotic range for $3.63 \leq \mu \leq 4$ and $0 \leq \alpha \leq 1$ [11].

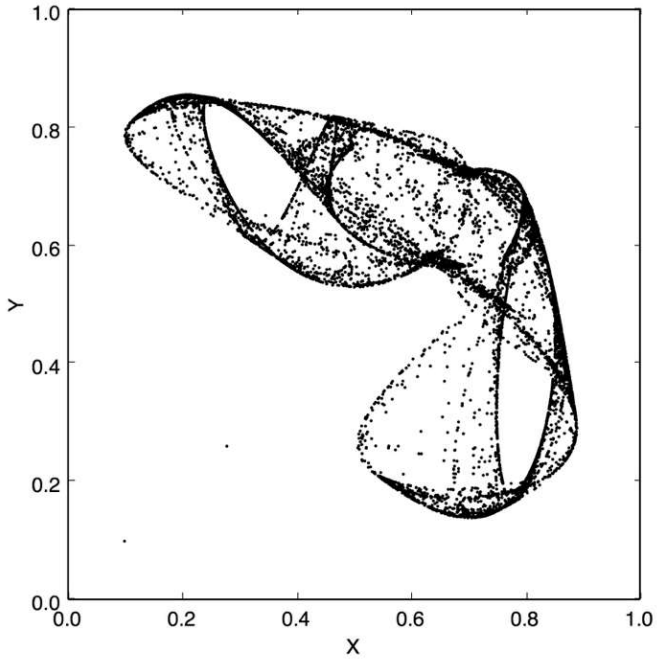


Figure 1. Chaotic attractor for the non-symmetric logistic coupled map $\mu = 3.1$, $\mu = 2.9$ and $\alpha = 0.3314$.

A group from University of California San Diego (UCSD) introduced a theoretical scheme in [12] for asymmetric encryption exploiting properties of nonlinear dynamical systems where a high-dimensional dissipative non-linear dynamical system is distributed between a transmitter and

a receiver. Therefore, they call the method distributed dynamics encryption (DDE). The transmitter dynamics are public, and the receiver dynamics are private, and they are not shared in the channel. A message is encoded by modulation of the parameters of the transmitter, and this results in a shift of the overall system attractor. An unauthorized receiver does not know the hidden dynamics of the receiver and cannot decode the message [12]. This proposal has been criticized due to its difficulties in the implementation and is categorized as non-practical [13].

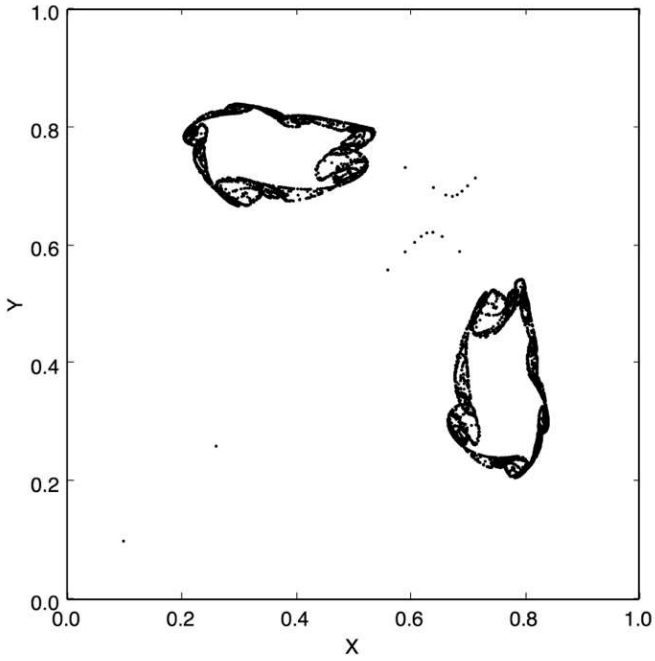


Figure 2. Another attractor for the logistic coupled map $\mu = 2.91$, $\mu = 2.9$, and $\alpha = 0.3314$.

This work will take the proposal of DDE and adapt it for a low-dimensional system using the coupled logistic map. We will study the encryption, the decryption, and the common attack for this cryptographic system. This is important for DDE because it provides an implementation without loss of security with acceptable cost and speed, which is a relevant cryptographic requirement for chaos-based cryptosystems suggested by [14]. It is our understanding that, at the time we wrote this work, this is the first fully functional computational implementation for encryption of DDE, besides the concept proof presented by the UCSD group. Even more, the work presented here is the missing example for DDE pointed out in the literature [4].

2. Encryption

The basic idea of distributed dynamics encryption (DDE) is to split a dynamical system of dimension $D_T + D_R$ into two parts with D_T transmitter variables $t(n) = [t_1(n); \ldots ; t_{DT}(n)]$, and D_R receiver variables $r(n) = [r_1(n) ; \ldots ; r_{DR}(n)]$. The receiver receives the scalar signal $s_t(n)$ from the transmitter, and the transmitter receives the scalar signal $s_r(n)$ from the receiver:

$$t(n+1) = F_T(t(n), s_r(n), m(n)) \tag{6}$$

$$r(n+1) = F_R(r(n), s_t(n)) \tag{7}$$

where $m(n)$ is the message which we want to encrypt. We allow that $m(n)$ only takes values 0 or 1; this requirement creates a binary message. The receiver must simulate the entire dynamics before she

starts the communication; this will create a list of points necessary for encrypting and decrypting the message. To perform the simulation, she will select the parameters and equations that will serve as the public and private keys, which is explained below.

The encryption for our low-dimensional implementation comes from a relation between Equations (4)–(7), where x (Equation (4)) will be the dynamic split to the transmitter and y (Equation (5)) will be the receiver part:

$$x_{n+1} = f_1(x_n) + \alpha(y_n - x_n) + \ \mathrm{A} * \mathrm{m} \tag{8}$$

$$y_{n+1} = f_2(y_n) - \alpha(y_n - x_n) \tag{9}$$

where the parameter A is a modulation of the message. In our implementation A takes random values between 0.001 and 0.01 to provide additional security to the system. Figure 3 shows an eight-bit encrypted message (01010111, which, using the ASCII standard, is the letter, "W"). For an easy identification we differentiate the points which correspond to a 0 bit to those which correspond a 1 bit. The security of this implementation resides in the overlap and closeness of those points. Only if you have the previous simulation can you decrypt the message. Figure 4 shows a different attractor with a longer message of 32-bits (01010111 01101111 01110010 01110100, which, using the ASCII standard, is a four-letter message, "Wort"). In this scheme, a different chaotic attractor represents a different pair of cryptographic keys. Equation (8), with its corresponding parameters, is the public key and Equation (9) is the private key. Something relevant is that the receiver does not need to know the parameter A to decrypt the message: in this sense parameter A represents a private key of the transmitter, providing more security to the encrypted message. Parameter A is not used in the decryption process because we are using a chaotic attractor, which, after some iterations of the full dynamic, is only known by the receiver, and the signal sent by the transmitter will converge to the attractor or not.

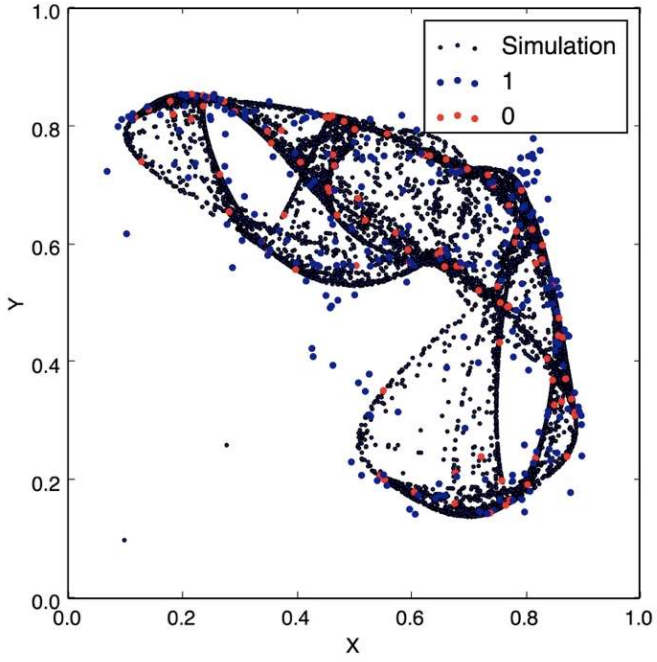


Figure 3. A message encrypted over the fully-known dynamic of Figure 1.

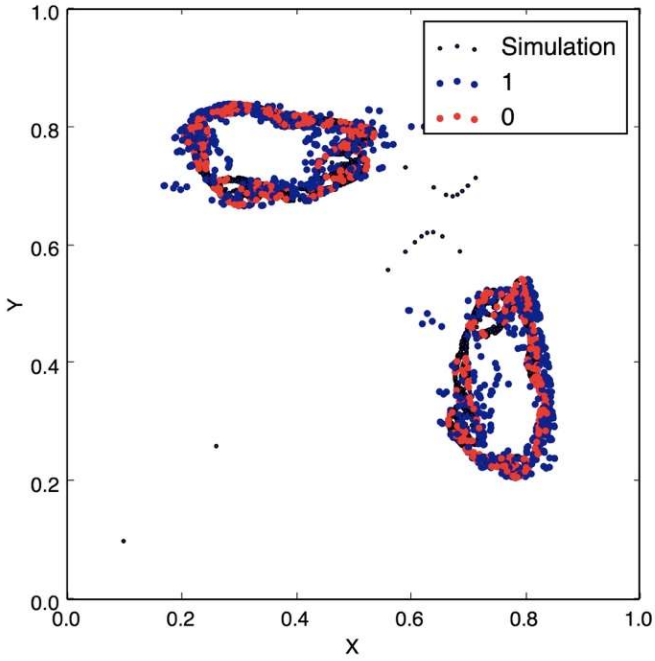


Figure 4. A longer message over the full dynamic of Figure 2.

3. Decryption

An authorized receiver knows all quantities, public and private, and can establish off-line the admissible attractors, or other dynamical aspects of the total system, for all allowed values of $m(n)$. For decryption, it is necessary to previously know all the points of the simulation. The decrypting process corresponds to the calculation of the distance of each received point from the transmitter to the points of the simulations. It is necessary to compute this distance to every point of the simulation and select the minimum value. If this value is lower than a tolerance parameter, which is related to parameter A of Equation (8), it is a 0-bit, and if it is greater, it is a 1-bit.

Even though this decryption process seems to be easy when the full dynamic is known, it is computational expensive, an aspect covered in Section 6 of this paper.

4. Cryptanalysis

An unauthorized receiver may attempt several methods to attack DDE and decode the secret message $m(n)$, but it has been demonstrated in [15] that the only one where non-defense can be used it is the one analyzed in this section. As the security resides in the fact the signal traveling in the channel is chaotic, our implementation is still as defensible as the original DDE. Figure 5 shows the data traveling through the communication channel, where an unauthorized receiver cannot easily resolve the message, and also due to topologically transitivity, as more data is transmitted in the channel, all the space will be occupied. On the other hand, Figure 6 shows the case when the attractor is not chaotic: here, it is easy to identify the two states (0 or 1).

One such method is to reconstruct the positions of the attractors that correspond to the transmission of 0 and 1 by storing and clustering samples of many transmitted bits. Knowing the positions of the attractors would enable the unauthorized receiver to decode the message using the same method as the authorized receiver [12].

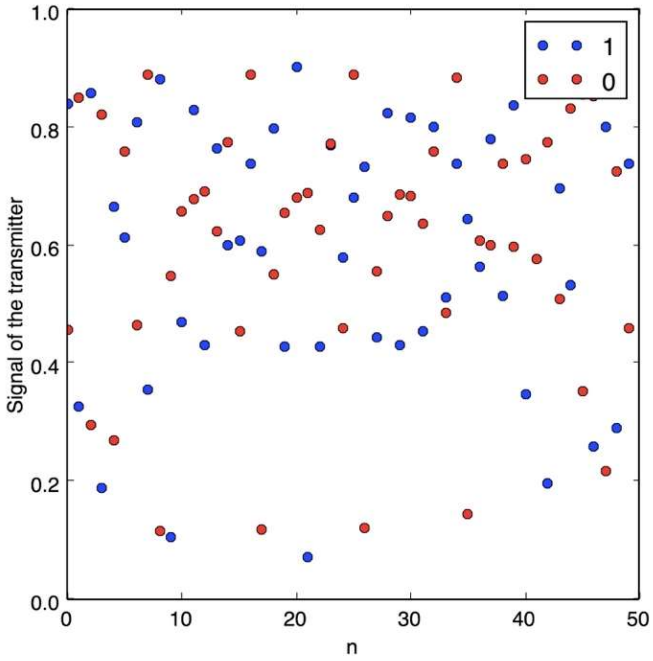


Figure 5. An example of the data, which can be captured from the communication channel by an unauthorized receiver.

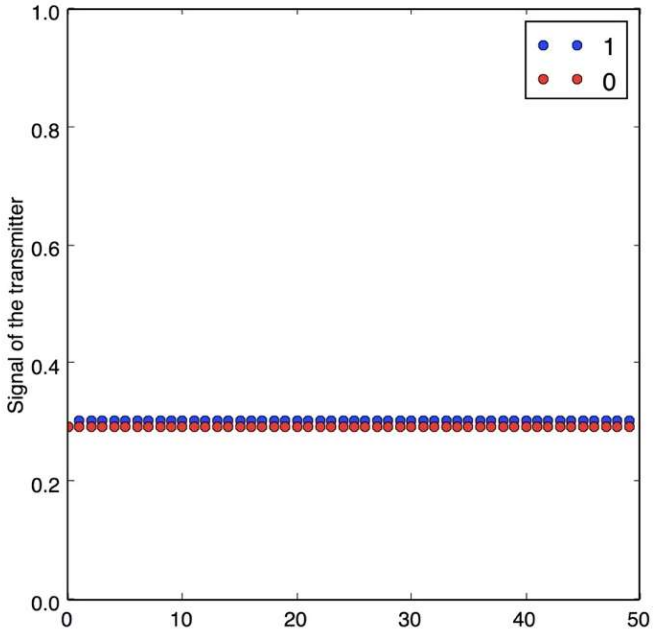


Figure 6. An example of the data, which can be captured from the communication channel when the attractor is not chaotic.

The chaotic dynamics may contain channel and noise that make the dynamics stochastic. An unauthorized receiver may attempt to generate a hidden Markov model of the transmitter public

dynamics for each possible value of the message m, and obtain a maximum likelihood (ML) estimation m′ of the message m. The decoded message will then be given by:

$$m' = max_{m\in(0,1)} p(s_t(1), \ldots, s_t(D_T + 1)|m) \tag{10}$$

In order to generate the hidden Markov model, the unauthorized receiver will need to quantize the transmitter state in a time delay reconstructed embedding space, and to estimate the state transition probabilities, as well as the observation probabilities of the model [15].

The attack proposed was implemented for our low-dimensional model. Figure 7 shows how the case of a message encrypted using a non-chaotic map where the training accuracy is 95%, which shows the effectiveness of the attack. When the number of bits is lower than that necessary to train the model, according to [14], the accuracy reduces to lower than 40%, in which a probabilistic attack, as in [1], cannot help this particular attack. This case is shown in Figure 8 where the red line is the decision boundary, which is far from being correct. Figure 9 has a larger number of bits, 60,000, and the boundary is more visible, but it is not enough to recover the message. Figure 10 shows the case where the number of bits is equal to that necessary to perform the attack: it is visible how the decision curve can resolve for 0 or 1-bits. Equation (9) from [15] shows the number of states that can be transmitted before the decision curve can be resolved:

$$Ns \approx \left(\frac{L_T}{L_q}\right)^{D_T} \tag{11}$$

where L_T is the range of the data transmitted and L_q is the quantization in the signal. Equation (11) is still useful in our case because it has been derived in general for any CML. For our implementation this number is around 4×10^7 which shows why Figure 10 can resolve the decision curve and how the security of this implementation is of the same level as the original DDE proposal.

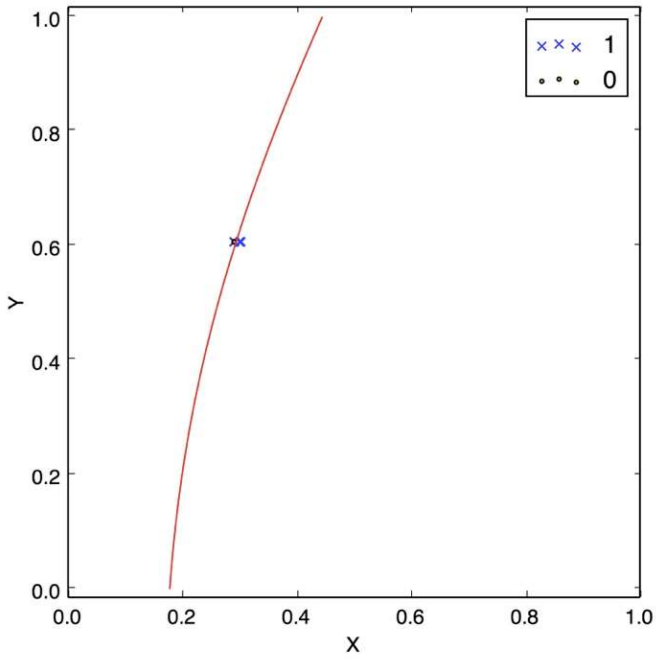


Figure 7. The decision surface (solid line) obtained from the attack when the attractor is not chaotic.

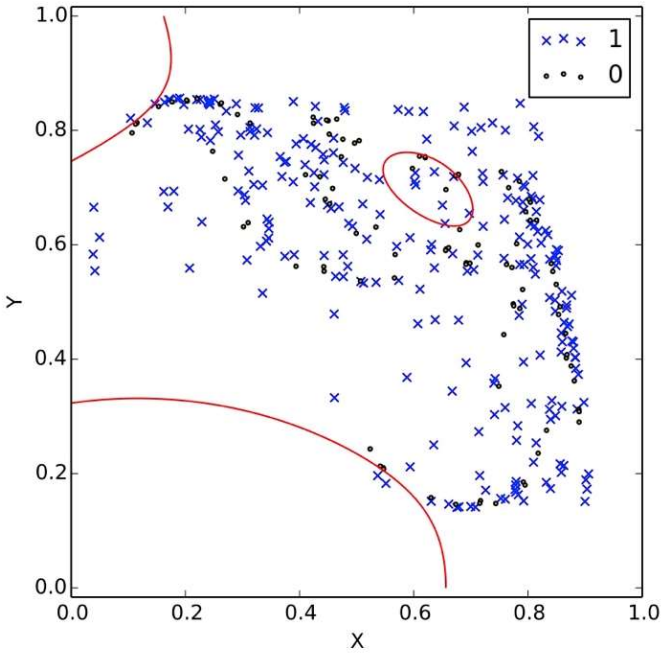


Figure 8. The decision surface (solid line) obtained from the attack when the attractor of Figure 1 has a low number of observations.

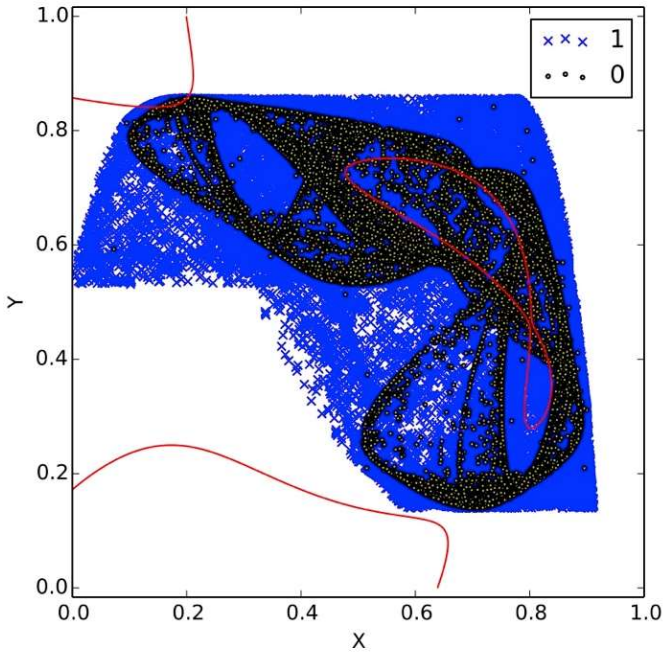


Figure 9. The decision surface (solid line) obtained from the attack for when the attractor of Figure 1 has a medium number of observations.

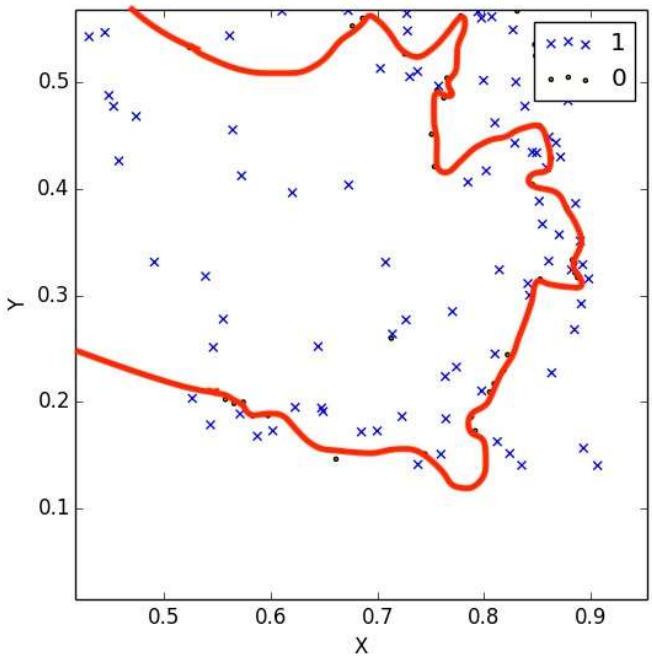


Figure 10. The decision surface (solid line) obtained from the attack when the attractor is the same as from Figure 1 and enough observations are known.

A relevant feature to point out is to see how a very small change in the dynamics of the receiver creates very different attractors. This can be observed in Figures 1 and 2, where the only change between them is the value of μ with just a 0.2 differential, and the produced dynamic is totally different. This is very useful because we do not need to change the public key to have a new crypto dynamic, representing a good option to protect from the attack described in this section.

The NLCM represents an improvement over the logistic map, but another map with our low-dimensional crypto scheme, like the piecewise linear map [16], represents a new pair of crypto keys.

The chaos degradation is a well-known problem for chaos-based cryptosystems [17]. In DDE, this problem is addressed by two means: (i) as the chaotic process is involved only in the simulation part, which happens off communication, algorithms of quality verification can be performed before using the data for the encryption; and (ii) also, as we are proposing when coupled maps from nature are used, there is information from the characterization of the phenomena that can indicate the degradation of the chaos, i.e., when the data is taken from an analog circuit.

5. Communication Scenario

In this section, we are going to provide the encryption and decryption algorithms for our proposal and give an example of the communication scenario between Alice and Bob.

Key selection. Alice should select μ_1, μ_2, and α. Additionally, Bob has his own private key: the parameter A, which is selected in the interval accepted by Alice.

Requirements. Alice, with the key selected, must compute the simulation with the help of Equations (8) and (9). This will generate a long list of (X,Y) points. Furthermore, she defines a tolerance which is related with the minimum value that she will accept for the parameter A.

Algorithm for encryption. To encrypt a message m, Bob should do the following:

(a) Obtain Alice's authentic public key (Equation (8) and μ_1).
(b) Represent the message as binary code.
(c) Obtain Alice's first 50 X data.

(d) Compute 50 Y data for the first bit, using Equation (8), μ_1, X data, and parameter A.
(e) Send Y data to Alice.
(f) Repeat for the next bits with a new 50 X data from Alice.

Algorithm for decryption. To recover the message Alice should do the following:

(a) Pair the first X data sent to Bob with the Y data received from Bob.
(b) Take just the last 12 pairs.
(c) For each pair calculate the Euclidian distance to each point of the long (X,Y) list of the simulation and preserve the minimum value of the distance.
(d) If the average of the 12 minimum pairs is greater than the tolerance, it is a 1-bit, otherwise it is a 0-bit.
(e) Repeat for the next 50 Y data from Bob.

Now for the example: Let us say that Alice wants to communicate with Bob, and he has a very important message to send her, the letter W. They want to use our crypto-proposal to transmit this letter by a secure channel. First Alice needs to choose the crypto-keys, μ_1, μ_2, and α. Recall that they must lead to chaotic conditions (she could verify this with help of the Lyapunov exponent of NLCM from [11]). Let us say she uses $\mu_1 = 3.1$, $\mu_2 = 2.9$, and $\alpha = 0.3314$, the same attractor from Figure 1. She will announce publicly Equation (8) with μ_1 and α from the previous selection and keep secret Equation (9) and μ_2. Bob will take this public information to transmit the message. Alice makes the simulation offline using Equations (8) and (9), and it will produce a long list of (X,Y) points. Equation (9), with its parameters, is the private key; it does not travel by channel. To start the communication from the long list that Alice has, she sends 50 X data to Bob. He will take his message and convert it to binary, he could use ASCII, so W will be 01010111, in eight bits. He takes the first bit, 0, and using Equation (8) recalculates a new pair Y for the fifty received from Alice and send back to her, in this Equation (8) it is the parameter A which Bob actually chooses, it is better if it is random, also to compute Equation he needs a Y_0 starting value which he selects also randomly. Eve the evil genius, who is listening in the channel from the data sent by Bob, cannot reconstruct the first bit from the 50 numbers thanks to the private keys of Bob and the fact that Equation (8) is in a chaotic state. Eve will need to wait until having enough data to use the attack described in the previous section. Alice receives the 50 Y numbers and using the last 12, pairs them with the last 12 of X that she sent and calculates the distance to every point in the long list from the simulation that she has and takes the minimum distances for the 12 pairs. If the average of the 12 minimum distances are lower than the tolerance (the minimum value that A can be) is a 0-bit, and if it is greater than the tolerance, it is a 1-bit. In this case, Alice will see a 0-bit. Now, the process is repeated for the next bits. Alice does not need to send adjacent X data to Bob for the transmission of the message.

6. Computer Experiment

Our implementation has been made in Python with the help of packages Numpy for the data management and HMMlearn for the Markov chain used in the attack. The calculations were performed on a Linux system with a Core i7 2.6GHz PC with 16 GB of RAM. Figures 1–4 were computed with one million points for the chaotic attractor with an average time of 33 s for the calculation. Figure 10 has the higher computation time; in this case, the Markov chain training took an average time of 72 h due to the 50 million bits needed for the training.

Table 1 shows the performance of the encryption and decryption algorithms where, as expected, the time of both increases as more bits are needed to be processed. It is remarkable how the encryption time is relatively small compared with the decryption time. This expensive decryption time is the weakness of DDE and opens additional research into this kind of crypto-system. However, we have shown how the DDE may be implemented.

Table 1. Details of the encryption/decryption time.

No. Bits	Encryption Time (s)	Decryption Time (s)
8	0.045	141.1081
16	0.0997	293.1638
32	0.2017	600.4974
64	0.3199	1318.4119
128	0.5973	2567.0675
256	1.7431	5297.5988

7. Conclusions

In this paper, we have made a full implementation of DDE, where we have described both encryption and decryption processes, showing how the implementation of DDE is possible with a low-dimensional dynamical system. This is relevant to this research field because it provides a functional example for DDE with the same kind of security provided by the high-dimensional systems which it is a key cryptographic requirement for chaos-based cryptography. Further, this implementation opens the possibility to investigate better ways to enhance the efficiency of DDE.

Our low-dimensional DDE represents a platform to evaluate different coupled maps. Future research can be done in the comparison of the security of low-dimensional and high-dimensional DDE.

Author Contributions: Conceptualization, H.S.-S.; Data curation, H.S.-S.; Formal analysis, H.S.-S.; Funding acquisition, E.G.B.; Investigation, H.S.-S. and E.G.B.; Methodology, H.S.-S.; Project administration, H.S.-S.; Supervision, E.G.B.; Validation, H.S.-S. and E.G.B.; Writing–original draft, H.S.-S.; Writing–review & editing, H.S.-S. and E.G.B.

Funding: This research was funded by MICITT and CONICIT grant number FI-0204-2012 and Universidad de Costa Rica's Vicerrectoría de Investigación, grant number 834-B5-293.

Acknowledgments: We want to thank Manuel Ortega-Rodríguez for helpful discussions about non-linear physics and complexity.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Katz, J.; Menezes, A.J.; Van Oorschot, P.C.; Vanstone, S.A. *Handbook of Applied Cryptography*; CRC Press: Boca Raton, FL, USA, 1996; ISBN 9780849385230.
2. Wang, X.; Gong, X.; Zhan, M.; Lai, C.H. Public-key encryption based on generalized synchronization of coupled map lattices. *Chaos* **2005**, *15*, 023109. [CrossRef] [PubMed]
3. Kocarev, L.; Makraduli, J.; Amato, P. Public-key encryption based on Chebyshev polynomials. *Circ. Syst. Signal Process.* **2005**, *24*, 497–517. [CrossRef]
4. Zhen, P.; Zhao, G.; Min, L.; Li, X. A survey of chaos-based cryptography. In Proceedings of the 2014 Ninth International Conference on P2P, Parallel, Grid, Cloud and Internet Computing, Guangdong, China, 8–10 November 2014; pp. 237–244.
5. Devaney, R.L. *An Introduction to Chaotic Dynamical Systems*; CRC Press: Boca Raton, FL, USA, 1996; ISBN 9780201130461.
6. Arroyo, D.; Amigo-Garcia, J.M.; Li, S.; Alvarez, G. *On the Inadequacy of Unimodal Maps for Cryptographic Applications*; RECSI: Tarragona, Spain, 2010; ISBN 9788469333044.
7. Lawnik, M. Generalized logistic map and its application in chaos based cryptography. *J. Phys. Conf. Ser.* **2017**, *936*, 012017. [CrossRef]
8. Kaneko, K. *Theory and Applications of Coupled Map Lattices*; Wiley: New York, NY, USA, 1993; Volume 159, ISBN 978-0471937418.
9. Lloyd, A.L. The coupled logistic map: A simple model for the effects of spatial heterogeneity on population dynamics. *J. Theor. Biol.* **1995**, *173*, 217–230. [CrossRef]
10. Schult, R.L.; Creamer, D.B.; Henyey, F.S.; Wright, J.A. Symmetric and nonsymmetric coupled logistic maps. *Phys. Rev. A* **1987**, *35*, 3115–3118. [CrossRef]

11. Zhang, Y.Q.; Wang, X.Y. Spatiotemporal chaos in Arnold coupled logistic map lattice. *Nonlinear Anal. Model. Control* **2013**, *18*, 526–541.
12. Tenny, R.; Tsimring, L.S.; Larson, L.; Abarbanel, H.D. Using distributed nonlinear dynamics for public key encryption. *Phys. Rev. Lett.* **2003**, *90*, 047903. [CrossRef] [PubMed]
13. Xiao, D.; Liao, X.; Deng, S. A novel key agreement protocol based on chaotic maps. *Inf. Sci.* **2007**, *177*, 1136–1142. [CrossRef]
14. Alvarez, G.; Li, S. Some Basic Cryptographic Requirements for Chaos-Based Cryptosystems. *Int. J. Bifurc. Chaos* **2006**, *16*, 2129–2151. [CrossRef]
15. Tenny, R.; Tsimring, L.; Abarbanel, H.; Larson, L. Security of chaos-based communication and encryption. In *Digital Communications Using Chaos and Nonlinear Dynamics*; Larson, L., Tsimring, L., Liu, J.-M., Eds.; Institute for Nonlinear Science, Springer: New York, NY, USA, 2006; pp. 191–229. ISBN 978-0387297873.
16. Elhadj, Z.; Sprott, J.C. Chaotifying 2-D piecewise-linear maps via a piecewise-linear controller function. *Nonlinear Oscill.* **2011**, *13*, 352–360. [CrossRef]
17. Li, S.; Mou, X.; Cai, Y.; Ji, Z.; Zhang, J. On the security of a chaotic encryption scheme: Problems with computerized chaos in finite computing precision. *Comput. Phys. Commun.* **2003**, *153*, 52–58. [CrossRef]