Agradecimientos

Resumen

Con el creciente uso de las redes de computadoras la transmisión y almacenamiento de la información digital, la protección de la confidencialidad, integridad y autenticidad de la información se ha convertido en una preocupación importante. El proyecto desarrollado en esta tesis propone una alternativa de cifrado de clave simétrica utilizando las propiedades de los sistemas dinámicos, es decir, un **esquema de transmisión segura basada en caos con tolerancia a perdida de información**.

Se presenta un nuevo esquema criptográfico basado en el caos que consiste en una transformación de texto plano no iterativo y alta dimensionalidad (k mapas) poblando dinamicamente una tabla de búsqueda (LUT) de acceso aleatorio que supera a la seguridad y la velocidad de previos esquemas de cifrado. Análisis experimental del esquema propuesto revela excelentes propiedades estadísticas, robustez a los ataques: diferencia y de texto plano elegido,así como de alto rendimiento para las comunicaciones multimedia en tiempo real.

Índice general

Índice de figuras

índice de tablas

Capítulo 1

Introducción

Durante las primeras décadas de su existencia, las redes de computadoras fueron usadas principalmente por investigadores universitarios para el envío de correo electrónico, y por empleados corporativos para compartir impresoras.

Hoy en día la seguridad en los datos es un aspecto importante en cualquier ámbito, debido a que cierta información principalmente datos bancarios, cuentas de usuarios y datos personales necesitan ser protegidos, es decir, que la información no pueda ser entendible para otras personas. En el ámbito computacional existen varias formas de lograr transformar la información legible, en información totalmente diferente a la original, esto se logra sometiendo los datos originales en un proceso que involucra operaciones matemáticas y/o permutaciones con el fin de que la información original sea ilegible para cualquier usuario, a éstos procesos se les conoce como algoritmos de cifrado.

Debido a que los algoritmos de cifrado cada vez se van volviendo mas sofisticados es muy poco probable que algún usuario sin conocer la clave pueda descifrar la información, como consecuencia aquellos usuarios que tienden a violar la seguridad no atacan directamente al algoritmo, lo que hacen es tratar de adivinar o generar claves aleatorias hasta lograr obtener la clave original.

La seguridad es un tema amplio, en su forma más sencilla se ocupa de garantizar que los curiosos no puedan leer, o peor aún, modificar mensajes dirigidos a otros destinatarios, además de usar mecanismos para verificar que el mensaje supuestamente enviado por una persona realmente venga de ella y no de otra. También se ocupa del problema de la captura y reproducción de mensajes legítimos, y de la gente que intenta negar el envío de mensajes.

En la actualidad toda la seguridad se basa en principios de criptografía para garantizar la confidencialidad de la información, a excepción de la seguridad en la capa física, debido a esto en este trabajo se propone un nuevo esquema de cifrado información basado en mapas caóticos, además debido al incremento de las aplicaciones multimedia y la demanda del acceso a la red se contempla que el cifrado sea tolerante a la perdida de información durante la transmisión para garantizar que la información tenga la mayor fludez posbile durante su reproducción.

1.1. Definición del problema

En los últimos años ha habido un crecimiento explosivo en el uso de equipo redes, por lo que la información digital, tales como texto, imágenes y otros archivos multimedia archivos se transmiten frecuentemente a través de las redes. La información digital que viaja a través de las redes puede potencialmente ser interceptada por alguien que no sea el previsto receptor.

La seguridad en sistemas de comunicación de multimedia (texto, audio, imagen y video) representan un reto difícil de alcanzar para los actuales estándares de cifrado (RSA-Rivest-Shamir-Adelman, AES- Advanced Encryption Standard e IDEA-International Data Encryption algorithm) ya que actualmente se requiere el procesamiento de grandes cantidades de información a velocidades que fluctúan entre los kilobits/seg (Kbps) hasta los Megabits/seg (Mbps). Esto debido a la aparición de las aplicaciones multimedia que permiten establecer una comunicación "en tiempo real" entre dos usuarios sin importar su ubicación geográfica.

La fluidez de los datos de una aplicación multimedia (audio o video) recibidos está determinada por los tiempos OTT (One way Transmision Time) y RTT (Round Trip Time), así como de una cuantificación inicial de las pérdidas de los paquetes y su distribución. Si un paquete se pierde, la calidad de la información se degrada a menos que sea recobrada con mecanismos de corrección de errores (FEC- Forward Error Correction) o retransmisión (ARQ -Automatic Repeat Request). Si el retardo de una vía es demasiado grande y se pierde la sincronía en la reproducción, conducirá a pérdidas de información por retraso. Por otro lado, un RTT grande degradará la interactividad de la aplicación [1].

1.2. Justificación

La teoría del caos es una disciplina científica que se centra en el estudio de sistemas no lineales que son muy sensibles a las condiciones iniciales que es similar al comportamiento aleatorio, y continua del sistema. Las propiedades de los sistemas caóticos son:

- Determinista, esto quiere decir que ellos tienen algunas ecuaciones matemáticas que determinan y gobiernan su comportamiento.
- Impredecible y no lineal, esto significa que son sensibles a las condiciones iniciales. Incluso un cambio muy ligero en el punto de partida puede llevar a resultados muy diferentes
- parecen ser al azar y desordenado, pero en realidad no lo son. Bajo el comportamiento al azar, hay un sentido de orden y el patrón.

La naturaleza altamente impredecible y aleatoria es la característica más atractiva de sistema caótico determinista, debido a que son problemas probabilísticamente

difíciles de resolver, eliminando una de las desventajas fundamentales de la criptografía convencional.

El esquema de cifrado que se propone está basado en un arreglo de n mapas caóticos independientemente iterados junto con un sistema de retroalimentación espacio-temporal usado como proceso de difusión de la información, por lo que además de garantizar la confidencialidad de la información debe ser tolerante a pérdida de información para evitar retrasos en la reproducción ya que la pérdida de algunos paquetes de información no afecta en su totalidad la reproducción, por ejemplo, durante una video conferencia la perdida de información debe mostrar manchas negras o algun tipo de distorción sin interrumpir la transmisión.

1.3. Objetivos

1.3.1. Objetivo general

Proponer un nuevo esquema de cifrado basado en caos para transmisión de información y tolerante a perdida de paquetes de información para evitar retrasos en su reproducción.

1.3.2. Objetivos específicos

- Analizar el funcionamiento de los sistemas caóticos y su uso para cifrado de información.
- Utilización de n mapas caóticos para incrementar la seguridad y robustez.
- Realizar el esquema encriptado para información multimedia (audio, video).
- Implementación del esquema para poder comparar su rendimiento con otros esquemas basados en caos.
- Probar el esquema con información de imagenes.
- Probar el esquema con información de audio.
- Probar el esquema con información de video.
- Analizar su robustez a ataques conocidos (ataque de criptotexto, ataque de texto plano conocido, ataque de texto plano selectivo y ataque de criptotexto selectivo)
- Analizar su rendimiento bajo distintas plataformas (Netbooks, notebook)

1.4. Alcances y limitaciones

Para este proyecto una ves encontrado las funciones más adecuadas para llevar a cabo el proceso de cifrado, se debe realizar la implementación del algoritmo tanto de cifrado como la de descifrado para llevar a cabo las pruebas en las que se van simulando la perdida de los paquetes, y asi realizar el análisis de los resultados que se vayan obteniendo.

1.5. Organización del documento

El documento se encuentra divido en x capítulos:

- Capítulo 1: Introducción, da una descripción del problema mencionado, lo que se espera alcanzar, los objetivos específicos los cuales ayudarán a alcanzar el objetivo general, así como los alcances y limitaciones del proyecto.
- Capítulo 2: Seguridad Informática, se da la definición de lo que es la seguridad informática y los problemas a los que se enfrenta uno hoy en día y muestra las acciones que se pueden realizar para evitar que la seguridad sea violada y la información llegue a manos de personas no autorizadas.
- Capítulo 3: seguridad y criptografía, en éste capítulo se describen algunas de las tecnicas de cifrado que se han utilizado a lo largo de la historia para garantizar la confidencialidad de la información.
- Capítulo 4: Introducción al caos, aquí se explica lo que es el caos así como las propiedades y el comportamiento que los sistemas dinámicosue poseen.

Capítulo 2

Seguridad informática

Para comenzar a hablar de criptografía y sistemas criptográficos primero debemos tener bien claro que es la seguiridad. [2]

2.1. Definición

Se entiende por seguridad de los sistemas de información al conjunto de recursos (metodologías, planes, programas y dispositivos físicos) encaminados a lograr que los recursos de computo disponibles en un ambiente dado, sean accedidos única y exclusivamente por quieres tienen la autorización para hacerlo.

La seguridad informática debe vigilar principalmente por las siguientes propiedades:

- **Confidencialidad**:

  Se define como la "condición que asegura que la información no pueda estar disponible o ser descubierta por o para personas, entidades o procesos no autorizados". La información debe ser vista y manipulada únicamente por quienes tienen el derecho o la autoridad de hacerlo. A menudo se la relaciona con la intimidad o privacidad, cuando esa información se refiere a personas físicas.

- **Integridad**:

  Se define como la "condición de seguridad que garantiza que la información es modificada, incluyendo su creación y borrado, solo por el personal autorizado". La información debe ser consistente, fiable y no propensa a alteraciones no deseadas. Un ejemplo de ataque a la integridad es la modificación no autorizada de saldos en un sistema bancario o de calificaciones en un sistema escolar.

- **Control**:

  Permite asegurar que solo los usuarios autorizados puedan decidir cuando y como permitir el acceso a la misma.

- **Disponibilidad**:

Se define como el "grado en el que un dato esta en el lugar, momento y forma en que es requerido por el usuario autorizado. Situación que se produce cuando se puede acceder a un sistema de información en un periodo de tiempo considerado aceptable". Se asocia a menudo a la fiabilidad técnica (tasa de fallos) de los componentes del sistema de información. La información debe estar en el momento que el usuario requiere de ella. Un ataque a la disponibilidad es la negación de servicio o "tirar" el servidor.

- **Autenticación**:

  Se define como "el mecanismo que permite conocer si la persona que esta accediendo a un sistema, es realmente quien debe acceder y no un extraño". El no repudio se refiere a los que se hacen sobre temas de correo electrónico para garantizar la autenticidad del remitente (un mecanismo son las firmas digitales).

  Adicionalmente pueden considerarse aspectos adicionales relacionados pero que incorporan aspectos particulares:

  - **Protección a la replica**: mediante la cual se asegura que una transacción sólo puede realizarse una vez, a menos que se especifique lo contrario. No se deberá poder grabar una transacción para luego reproducirla, con el propósito de copiar la transacción para que parezca que se recibieron múltiples del mismo remitente original.
  - **No repudio**: mediante la cual se evita que cualquier entidad que envió o recibió información niegue, ante terceros, que la envió o recibió.
  - **Consitencia**: se debe poder asegurar que el sistema se comporte como se supone que debe hacerlo ante los usuarios que corresponda.
  - **Aislamiento**: este aspecto, íntimamente relacionado con la confidencialidad permite regular el acceso al sistema, impidiendo que personas no autorizadas hagan uso del mismo.
  - **Auditoria**: es la capacidad de determinar que acciones o procesos se están llevando a cabo en el sistema, así como quién y cuando las realiza.

2.1.1. Análisis del objetivo de la seguridad informática

Para comenzar el análisis de la seguridad informática se deberá conocer las características de lo que se pretende proteger: la información.

Se define **Dato** como la unidad mínima con la que se compone cierta información (Datum = a lo que se da).

La *información* es una agregación de datos que tiene un significado especifico mas allá de cada uno de estos, y tendrá un sentido particular según como y quien la procese.

Establecer el valor de la información es algo totalmente relativo, pues constituye un recurso que, en muchos casos, no se valora adecuadamente debido a su intangibilidad, cosa que no ocurre con los equipos, las aplicaciones y la documentación.

Existe información que debe o puede ser pública y aquella que debe ser privada. En esta última debemos maximizar los esfuerzos para preservarla de ese modo reconociendo las siguientes características en la información:

- Es crítica: es indispensable para garantizar la continuidad operativa.
- Es valiosa: es un archivo con valor en si mismo
- Es sensitiva: deber ser conocida por las personas que la procesan y sólo por ellas.

2.1.2. Principios básicos para la seguridad

- Suponer que el diseño del sistema es público
- El defecto deber ser: sin acceso.
- Revisar permanentemente.
- Los mecanismos de protección deben ser simples, uniformes y construidos en las capas más básicas del sistema.
- Los mecanismos deber ser aceptados sicológicamente por los usuarios.

Requisitos

Los requisitos en seguridad de la información dentro de una organización sufren continuamente muchos cambios. Antes que se extendiera la utilización de los equipos de procesamiento de datos, la seguridad de la información, que era de valor para una institución se conseguía fundamentalmente por medios físicos y administrativos. Como por ejemplo el uso de cajas de seguridad con combinaciones de apertura para almacenar documentos confidenciales.

Con la introducción de las computadoras, fue evidente la necesidad de herramientas automáticas para proteger los ficheros y otras informaciones almacenadas en su memoria.

Uno de los problemas que afecta a la seguridad, es la introducción de los sistemas distribuidos y la utilización de redes y facilidades de comunicación para transportar datos entre terminales de usuarios y computadoras, y de computadora a computadora. Así como también el enorme crecimiento que ha tenido internet en la última década.

Como ya mencionamos se la debe considerar a la seguridad como un aspecto de gran importancia en cualquier organización que trabaje con sistemas informáticos.

2.1.3. La seguridad como problema cultural

Una de las paradojas es que ha pesar de que cada vez se destinan mayores recursos para el área informática y que ésta se ha vuelto esencial para la gestión de negocios de las empresas, el presupuesto asignado específicamente al tema de seguridad, no ha crecido en la misma proporción. Por esto es fundamental crear conciencia al interior

de las organizaciones para que puedan dimensionar en su justa medida la relevancia del problema, porque si se miran los presupuestos informáticos dentro de las empresas, vemos que han crecido notablemente, pero no ha ocurrido lo mismo con los presupuestos asignados a las áreas de seguridad.

Mientras más tecnología se incorpora, mas se agranda la brecha en lo que son debilidades de seguridad. Actualmente hay empresas que basan sus procesos de negocios en TI y eso provoca que la empresa este dependiendo cada vez más de estas herramientas tecnológicas y paralelamente van creciendo los temas relacionados con la seguridad. Por eso es fundamental la creación de conciencia en las empresas.

Una de las razones por las cuales no ha despegado fuertemente el comercio electrónico en el país es que ante la decisión de las empresas de abrirse a este tema, que van a requerir el desarrollo de mecanismos de seguridad, prefieren postergarla y si ha este sumamos la escasa condición de la legislación con respecto al tema, la opción queda desechada.

La tecnología disponible hoy en día hace posible una transferencia electrónica en forma segura, el problema es que la gente no sabe como hacerlo y tiene como consecuencia que el país se esta quedando atrás no por un problema tecnológico, sino por un problema de mentalidad. Sin duda como vemos la seguridad es fundamental no solo para evitar desastres o perdidas irrecuperables que afecten el funcionamiento de las organizaciones sino que también para potencial nuevas áreas de negocios que permitan el crecimiento de diferentes actores del mercado.

2.1.4. La seguridad como proceso

Uno de los puntos de consenso en el tema es que la seguridad es un proceso y no actividad particular que desarrolla la empresa, un proceso que barre todas las unidades funcionales de esta. Al hablar de seguridad hay que involucrar muchos aspectos que no solo están relacionados con herramientas tecnológicas. Abordar el tema de seguridad no solo implica una solución de hardware y software, también involucra un conocimiento sobre el riesgo que significa no dar confiabilidad a la información, lo que en ocasiones tiene que ver con un desconocimiento de parte de los administradores de sistemas sobre el tema.

El problema hay que enfrentarlo con tecnología, pero también debe involucrar a los tomadores de decisiones, que son finalmente quienes deciden las inversiones, ellos deben comprender claramente la problemática para destinar los recursos necesarios para garantizar la confiabilidad, disponibilidad e integridad de los datos.

2.1.5. Factores que afectan a los sistemas informáticos

Los principales factores que se observan sobre los sistemas informáticos tienen orígenes diversos. Así, si consideramos las amenazas externas, el hardware puede ser físicamente dañado por agua, fuego, terremotos, sabotajes,... Las mismas causas pueden daños los medios magnéticos de almacenamiento externo. La información contenida en

éstos, también puede verse afectada por campos magnéticos intensos y frecuentemente, por errores de operación. Las líneas de comunicación pueden ser interferidas, etc.

Otros tipos de amenazas provienen de usuarios o empleados infieles. Así, los primeros pueden usurpar la personalidad de usuarios autorizados y acceder indebidamente a datos para su consulta o borrado, o aunque algo más complicado, modificar en su provecho programas de aplicación.

Otras amenazas más sutiles provienen de inadecuados controles de programación. Así, el problema de residuos, es decir, de la permanencia de información en memoria principal cuando ésta es liberada por un usuario o, en el caso de dispositivos externos cuando ésta es incorrectamente borrada.

Una técnica fraudulenta muy usada consiste en transferir información de un programa a otro mediante canales ilícitos y no convencionales (canales ocultos). En la Fig. 2.1 de la Pág11 vemos las amenazas más frecuentes a la seguridad de un sistema de información.

Las amenazas pueden ser analizadas en tres momentos: antes del ataque, durante y después del mismo. Estos mecanismos conformarán políticas que garantizarán la seguridad de nuestro sistema.

- La prevención (antes): mecanismos que aumentan la seguridad o fiabilidad de un sistema durante su funcionamiento normal. Por ejemplo el cifrado de información para su posterior transmisión.

- La detección (durante): mecanismos orientados a revelar violaciones a la seguridad. Generalmente son programas de auditoria.

- La recuperación (después): mecanismos que se aplican, cuando la violación de un sistema ya se ha detectado, para retornar este a su funcionamiento normal. Por ejemplo recuperación desde las normas de seguridad (backup) realizadas.

Es muy importante ser consciente que por más que unas empresa sea la más segura desde el punto de vista de ataques externos, los hackers, virus, etc. La seguridad de la misma será nula si no se ha previsto como combatir un incendio.

2.2. Seguridad física

La seguridad física consiste en la "aplicación de barreras físicas y procedimientos de control, como medidas de prevención y contramedidas ante amenazas a los recursos e información confidencial". Se refiere a los controles y mecanismos de seguridad dentro y alrededor del centro de cómputo así como los medios de acceso remoto al y desde él mismo; implementados para proteger el hardware y medios de almacenamiento de datos.

Este tipo de seguridad está enfocado a cubrir las amenazas ocasionadas tanto por el hombre como por la naturaleza del medio físico en que se encuentra ubicado el centro.

Las principales amenazas que se prevén en la seguridad física son:

- Desastres naturales, incendios accidentales, tormentas e inundaciones.
- Amenazas ocasionadas por el hombre.
- Disturbios, sabotajes internos y externos deliberados.

No hace falta recurrir a películas de espionaje para sacar ideas de cómo obtener la máxima seguridad en un sistema informático, además de que la solución sería extremadamente cara. A veces basta recurrir al sentido común para darse cuenta que cerrar una puerta con llave o cortar la electricidad en ciertas áreas siguen siendo técnicas válidas en cualquier entorno.

Los peligros más importantes que se corren en un centro de procesamiento son incendios, inundaciones, condiciones climatológicas. El objetivo es mantener una serie de acciones para seguirlas en forma eficaz y oportuna para la prevención, reducción, recuperación y corrección frente a estos tipos de riesgos.

2.3. Seguridad lógica

Es importante recalcar que la mayoría de los daños que puede sufrir un sistema de computo no será sobre los medios físicos sino contra información por él almacenada y procesada.

La seguridad física sólo es una parte del amplio espectro que se debe cubrir para no vivir con una sensación ficticia de seguridad. Como ya se ha mencionado, el activo más importante que se posee es la información, y por lo tanto deben existir técnicas, más allá de la seguridad física, que la aseguren. Estas técnicas las brinda la seguridad lógica.

La seguridad lógica consiste en la "aplicación de barreras y procedimientos que resguarden el acceso a los datos y sólo se permita acceder a ellos a personas autorizadas para hacerlo".

Existe un viejo dicho en la seguridad informática que dicta que "todo lo que no está permitido debe estar prohibido" y esto es lo que debe asegurar la seguridad lógica.

Los objetivos que se plantean serán:

- Restringir el acceso a los programas y archivos.
- Asegurar que los operadores puedan trabajar sin una supervisión minuciosa y no puedan modificar los programas ni los archivos que no le correspondan.
- Asegurar que estén utilizados los datos, archivos y programas correctos en y por el procedimiento correcto.
- Que la información transmitida sea recibida sólo por el destinatario al cual ha sido enviada y no a otro.
- Que la información recibida sea la misma que ha sido transmitida.

- Que existan sistemas alternativos secundarios de transmisión entre diferentes puntos.

- Que se disponga de pasos alternativos de emergencia para la transmisión de información.

Una vez conocidas las vulnerabilidades y ataques a las que está expuesto un sistema es necesario conocer los recursos disponibles para protegerlo Mientas algunas técnicas son evidentes (seguridad física por ejemplo) otras pautas no lo son tanto e incluso algunas pueden ocasionar una sensación de falsa seguridad.

Muchas vulnerabilidades estudiadas son el resultado de implementación incorrecta de tecnologías, otras son consecuencias de la falta de planeamiento de las mismas pero, como ya se ha mencionado la mayoría de los agujeros de seguridad son ocasionados por los usuarios de dichos sistemas y es responsabilidad del administrador detectarlas y encontrar la mejor manera de cerrarlos.

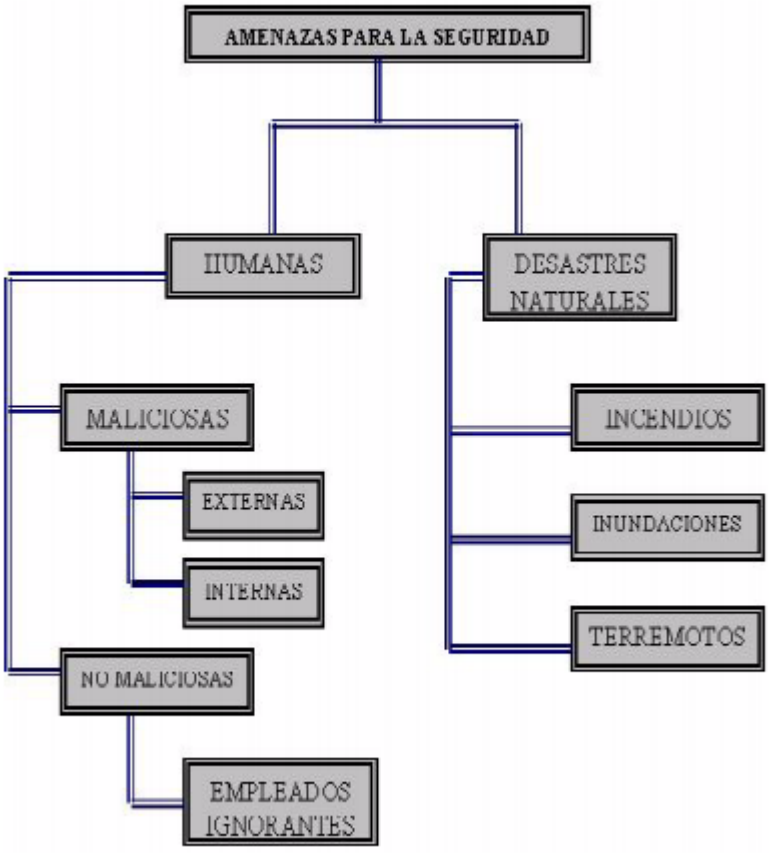


Figura 2.1: Amenazas frecuentes

2.4. Controles de acceso

Estos controles pueden implementarse en el sistema operativo, sobre los sistemas de aplicación, en bases de datos, en un paquete específico de seguridad o en cualquier otro utilitario. Constituyen una importante ayuda para proteger al sistema operativo de la red, al sistema de aplicación y demás software de la utilización o modificaciones no autorizadas; para mantener la integridad de la información y para resguardar la información confidencial de accesos no autorizados.

Asimismo, es conveniente tener en cuenta otras consideraciones referidas a la seguridad lógica, como por ejemplo las relacionadas al procedimiento que se lleva a cabo para determinar si corresponde un permiso de acceso a un determinado recurso.

2.4.1. Acceso - uso - autorización

La identificación de estas palabras es muy importante ya que el uso de algunas implica un uso desapropiado de las otras. Específicamente "Acceso" y "Hacer uso" no son el mismo concepto cuando se estudian desde el punto de vista de un usuario y de un intruso. Por ejemplo:

- Cuando un usuario tiene acceso autorizado, implica que tiene autorizada el uso de un recurso.
- Cuando un atacante tiene acceso desautorizado está haciendo uso desautorizado del sistema.

Pero, cuando un atacante hace uso desautorizado de un sistema, esto implica que el acceso fue autorizado. Luego un ataque será un intento de acceso, o uso desautorizado de un recurso, sea satisfactorio o no. Un incidente envuelve un conjunto de ataques que pueden ser distinguidos de otro grupo por las características del mismo.

2.4.2. Identificació de las amenazas

La identificación de amenazas requiere conocer los tipos de ataques, el tipo de acceso, la forma operacional y los objetivos del ataque.

Las consecuencias de los ataque se podrían clasificar en:

- Data Corruption: la información que no contenía defectos para a tenerlos.
- Denial of Service (DoS): servicios que deberían estar disponibles no lo están.
- Leakage: los datos llegan a destinos a los que no deberían llegar.

Desde 1990 hasta nuestros días, el CERT viene desarrollando una serie de estadísticas que demuestran que cada día se registran más ataque informáticos, y estos son cada vez más sofisticados, automáticos y difíciles de rastrear.

Cualquier persona, sin tener grandes conocimientos, pero con una potente y estable herramienta de ataque desarrollada por los gurús, es capaz de dejar fuera de servicio cualquier servidor de información de cualquier organismo en internet, simplemente siguiendo las instrucciones que acompañan la herramienta.

La información puede ser aprovechada para fines menos lícitos que para los cuales fue pensada, pero esto es algo ciertamente difícil de evitar.

2.4.3. Delito informático

Ya hemos dejado en claro la importancia de la información en el mundo altamente tecnificado de hoy. También se ha dejado en claro cada uno de los riesgos "naturales" con los que se enfrenta nuestro conocimiento y la forma de enfrentarlos.

El desarrollo de la tecnología informática ha abierto las puertas a nuevas posibilidades de delincuencia antes impensables. La cantidad de los delitos así ocasionados es a menudo muy superior a la usual en la delincuencia tradicional y también son mucho más elevadas las posibilidades de que no lleguen a descubrirse o castigarse.

2.4.4. Riesgos "no naturales"

Son aquellos que se encuadran en el marco del delito. El delito informático implica actividades criminales que los países han tratado de encuadrar en figuras típicas de carácter tradicional, tales como robos, hurtos, fraudes, falsificaciones, estafas, sabotajes. Sin embargo, debe destacarse que el uso de las técnicas informáticas han creado nuevas posibilidades del uso indebido de las computadoras lo que ha creado la necesidad de regulación por parte del derecho.

Se considera que no existe una definición formal y universal de delito informático pero se han formulado conceptos respondiendo a realidades nacionales contretas: " no es tan facíl dar un concepto sobre delitos informáticos, en razón de que su misma denominación alude a una situación muy especial, ya que para hablar de "delitos" en el sentido de acciones típicas, es decir, tipificadas o contempladas en textos jurídicos penales, se requiere la expresion "delitos informáticos" esté consignada en los códigos penales".

En 1983, la Organización de Cooperación y Desarrollo económico (OCDE) inició un estudio de las posibilidades de aplicar y armonizar en el plano internacional las leyes penales a fin de luchar contra el problema del uso indebido de los programas computacionales.

En 1992, la Asociación Internacional de Derecho Penal, durante el coloquio celebrado en Wurzburgo Alemania, adoptó diversas recomendaciones respecto a los delitos informáticos, entre ellas que, en medida que el derecho penal no sea suficiente, deberá promoverse la modificación de la definición de delitos existentes o la creación de otros nuevos, sino basta con la adopción de otras medidas como por ejemplo el "principio de subsidiariedad".

Se entiende **Delito** como: "la acción penada por las leyes por realizarse en perjuicio de algo o alguien, o por ser contraria lo establecido por aquéllas". Finalmente la OCDE publicó un estudio sobre delitos informáticos y el análisis de la normativa jurídica en donde se reseñan las normas legislativas vigentes y se define **delito informático** como "cualquier comportamiento antijurídico, no ético o no autorizado, relacionado con el procesado automático de datos y/o transmisiones de datos".

"Los delitos informáticos se realizan necesariamente con la ayuda de los sistemas informáticos, pero tienen como objeto del injusto la información en sí misma".

2.4.5. Estrategias de seguridad

El diseñar una estrategia de seguridad depende en general de la actividad que se desarrolla, sin embargo, se pueden considerar tres pasos generales:

1. Crear una política global de seguridad: se debe establecer el status de la información para la empresa u organización, debe de contener un objetivo general, la importancia de la tecnología para la empresa, el periodo de tiempo de validez de la política, los recursos con que se cuentan y los objetivos específicos de la empresa. Además debe establecerse la calidad de la información a manejar según el objetivo, es decir, que se establezca cuando o para quien la información debe ser confidencial, cuando debe verificarse su integridad, su autenticidad, tanto de la información como de los usuarios.[3]

2. Realizar un análisis de riesgo: enumerar todo tipo de riesgos a los cuales esta expuesta la información y cuales son las consecuencias, los posibles atacantes, amenazas, etc.

3. Aplicar las medidas correspondientes de seguridad: esto se puede plantear como la terminación de toda la estructura de seguridad de la información. Una vez planteada la política, es decir, cuanto vale la información, decir que tanto se pierde si le pasa algo a la información o que tanto se gana si esta protegida, debemos de establecer las medidas para que cumpliendo con las políticas las perdidas sean las menores posibles.

Las posibles medidas a establecer se pueden dividir en:

Tipo	Preventivas	Detectivas	Correctivas
Protección física	PF	DF	CF
Medidas técnicas	PT	DF	CF
Medidas de organización	PO	DO	CO

Tabla 2.1: Medidas para minimizar perdidas.

Donde:

PF podrá ser el control en el acceso de entrada, protección del hardware, respaldo de datos;

DF podría se detectores de movimiento, de metales, monitores de vigilancia, ...

CF podría ser respaldos de fuente de poder;

PT firewalls, criptografía, bitácora;

DT control de acceso lógico, sesión de autentificación,

CT programas antivirus;

PO podría ser organizaciones en las claves de acceso;

DO monitoreos de auditoría y finalmente plan de acciones, respaldos autmáticos correspondiente a CO.

En cuanto a las políticas de seguridad de hoy es importante hablar de un sistema cien por ciento seguro, sencillamente porque el costo de la seguridad total es muy alto. Por eso las empresas, en general, asumen riesgos: deben optar entre perder un negocio o arriesgarse a ser hackeadas.

La RFC 1244 define **Política de Seguridad** como: "Una declaración de intenciones de alto nivel que cubre la seguridad de los sistemas informáticos y que proporciona las bases para definir y delimitar responsabilidades para las diversas actuaciones técnicas y organizativas que se requerirán".

La política se refleja en una serie de normas, reglamentos y protocolos a seguir, donde se definen las medidas a tomar para proteger la seguridad del sistema, pero ante todo, "una política de seguridad es una forma de comunicarse con los usuarios... siempre hay que tener en cuenta que la seguridad comienza y termina con personas".

La cuestión es que, en algunas organizaciones puntuales, tener un sistema de seguridad muy acotado les impediría hacer más negocios. "Si un hacker quiere gastar cien mil dólares en equipos para descifrar una encriptación, lo puede hacer porque es imposible de controlarlo. Y en tratar de evitarlo se podrían gastar millones de dólares". La solución a medias, entonces sería acotar todo el espectro de seguridad, en lo que hace a plataformas, procedimientos y estrategias. De esta manera se puede controlar todo un conjunto de vulnerabilidades, aunque no se logre la seguridad total. Esto significa ni más ni menos que un gran avance con respecto a unos años atrás.

Algunas organizaciones gubernamentales y no gubernamentales internacionales han desarrollado documentos, directrices y recomendaciones que orientan en el uso adecuado de las nuevas tecnologías para obtener el mayor provecho y evitar el uso indebido de las mismas, lo cual puede ocasionar serios problemas en los bienes y servicios de las empresas en el mundo.

En este sentido, las Políticas de Seguridad Informática (PSI), surgen como una herramienta organizacional para concientizar a cada uno de los miembros de una organización sobre la importancia y sensibilidad de la información y servicios críticos. Estos permiten a las compañias desarrollarse y mantenerse en su sector de negocios.

Cada sistema es único y por lo tanto la política de seguridad a implementar no será única. El proponer o identificar una política de seguridad requiere un alto compromiso con la organización, agudeza técnica para establecer fallas y debilidades, y constancia para renovar y actualizar dicha política en función del dinámico ambiente

que rodea las organizaciones modernas.

En definitiva la seguridad informática no tiene una solución definitiva aquí y ahora, sino que es y será el resultado de la innovación tecnológica, a la par del avance tecnológico, por parte de aquellos que son los responsables de nuestros sistemas.

2.4.6. Otras medidas de seguridad

Mecanismos de autorización

Un sistema de computación puede verse como una colección de objetos (procesos, procesadores, segmentos de memoria, discos, impresoras, archivos, semáforos). Cada objeto debe tener un nombre único para poder identificarlo y un número finito de operaciones que los procesos pueden efectuar sobre él. Podemos ver a estos objetos como tipos abstractos de datos.

Obviamente, un proceso no debe poder accesar objetos sobre los que no tengan autorización. También debe ser posible restringir el uso de un objeto por parte de un proceso sólo a ciertas operaciones. Por ejemplo, un proceso podría tener autorización para leer, pero no para escribir un determinado archivo.

Dominios de protección

Un dominio de protección es un conjunto de pares (objeto, operaciones), cada par identifica un objeto y las operaciones permitidas sobre él.

En cada instante, cada proceso ejecuta dentro de un dominio de protección. Los procesos pueden cambiar de un dominio a otro en el tiempo; el cómo depende mucho del sistema. En UNIX, se asocia un dominio a cada usuario; dado un usuario y el grupo al cual pertenece, se puede construir una lista de todos los objetos que puede accesar y con qué operaciones. Cuando un usuario ejecuta un programa almacenada en un archivo con propiedad de otro usuario B, el proceso puede ejecutar dentro del dominio de pretección de A o B, dependiendo del bit de dominio o *setuserid* bit del archivo.

Este mecanismo se usa con algunos utilitarios. Por ejemplo, el programa *passwd* debe tener privilegios que un usuario común no tiene, para poder modificar el archivo donde se guardan las claves. Lo que se hace es que el archivo /bin/passwd que contiene el programa es propiedad del superusuario, y tiene el setuserid encendido. Este esquema es peligroso: un proceso puede pasar de un estado en que tiene poco poder a otro en que tiene poder absoluto. Cualquier error en un programa como passwd puede significar un gran hoyo en la seguridad del sistema. Cuando se hace una llamada al sistema también se produce un cambio de dominio, puesto que la llamada se ejecuta en modo protegido.

Listas de acceso

A cada objeto se asocia una lista de pares (dominio, derechos). Es lo que se conoce como lista de acceso o ACL. Si pensamos en archivos Unix, podemos almacenar esta lista en el nodo-i de cada archivo, y sería algo así como:

((Juan,*,RW),(Pedro,Profes,RW),(*,Profes,R))

En la práctica se usa un esquema más simple y menos poderoso, pero que puede considerarse aún una lista de accesos, reducida a 9 bits. 3 para el dueño (RWX), 3 para el grupo y 3 para el resto del mundo.

Windows NT usa listas de accesos con todo el nivel de detalle que uno quiera: para cualquier usuario o grupo, se puede especificar cualquier subconjunto de derechos para un archivo, de entre {RWXDPO}.

CAPACIDADES

La otra posibilidad es almacenar la matriz por filas. En este caso, a cada proceso se le asocia una lista de capacidades. Cada capacidad corresponde a un objeto más las operaciones permitidas.

Cuando se usan capacidades, lo usual es que, para efectuar una operación M sobre un objeto O, el proceso ejecute la operación especificando un puntero a la capacidad correspondiente al objeto, en vez de un puntero al objeto. La sola posesión de la capacidad por parte del proceso quiere decir que tiene los derechos que en ella se indican Por lo tanto, se debe evitar que los procesos puedan "falsificar" capacidades.

Una posibilidad es mantener las listas de capacidades dentro del sistema operativo, y que los procesos sólo manejen punteros a las capacidades, no las capacidades propiamente. Otra posibilidad es cifrar las capacidades con una clave conocida por el sistema, pero no por el usuario. Este enfoque es particularmente adecuado para sistemas distribuidos, y es usado en Amoeba.

Un problema de las capacidades es que puede ser difícil revocar derechos ya entregados. En Amoeba, cada objeto tiene asociado un número al azar, grande, que también está presente en la capacidad. Cuando se presenta una capacidad, ambos números deben coincidir. De esta manera, para revocar los derechos ya otorgados, se cambia el número asociado al objeto.

Mecanismos de autentificación

La autentiicación, consiste en identificar a los usuarios que entrar al sistema, se puede basar en posesión (llave o tarjeta), conocimiento (clave) o en un atributo del usuario (huella digital).

Claves

El mecanismo de autentiicación más ampliamente usado se basa en el uso de claves o passwords; es fácil entender y fácil de implementar. En UNIX existe un archivo /etc/passwd donde se guarda los nombres de usuarios y sus claves, cifradas mediante una función one-way F. El programa login pide nombre y clave, computa F(clave), y busca el par (nombre, F(clave)) en el archivo.

Con claves de 7 caracteres tomados al azar de entre los 95 caracteres ASCII que se

pueden digitar con cualquier teclado, entonces las 957 posibles claves deberían desincentivar cualquier intento por adivinarla. Sin embargo, una proporción demasiado grande de las claves escogidas por los usuarios son fáciles de adivinar, pues la idea es que sean también fáciles de recordar. La clave también se puede descubrir mirando cuando el usuario la digita o si el usuario hace login remoto, interviniendo la red y observando todos los paquetes que pasan por ella. Por último, además de que las claves se pueden descubrir, éstas también se pueden "compartir", violando las reglas de seguridad. En definitiva, el sistema no tiene ninguna garantía de que quien hizo login es realmente el usuario que se supone que es.

Identificación física

Un enfoque diferente es usar un elemento físico difícil de copiar, típicamente una tarjeta con una banda magnética. Para mayor seguridad este enfoque se suele combinar con una clave (como es el caso de los cajeros automáticos). Otra posibilidad es medir características físicar particulares del sujeto: huella digital, patrón de vasos sanguíneos de la retina, longitud de los dedos. Incluso la firma sirve.

Otros métodos de protección

1. Sistemas de detección de intrusos: son sistemas que permiten analizar las bitácoras de los sistemas en busca de patrones de comportamiento o eventos que puedan considerarse sospechosos, sobre la base de la información con la que han sido previamente alimentados. Pueden considerarse como monitores.

2. Sistemas orientados a conexión de red: monitorean las conexiones que se intentan establecer en una red o equipo en particular, siendo capaces de efectuar una acción sobre la base de métricas como: origen y destino de la conexión, servicio solicitado, permisos, etc. Las acciones que pueden emprender suelen ir desde el rechazo de la conexión hasta alerta al administrador. En esta categoría estan los cortafuegos (Firewalls) y los Wrappers.

3. Sistemas de análisis de vulnerabilidades: analizan sistemas en busca de vulnerabilidades conocidas anticipadamente. La "desventaja" de estos sistemas es que pueden ser utilizados tanto por personas autorizadas como por personas que buscan acceso no autorizado al sistema.

4. Sistemas de protección a la integridad de la información: sistemas que mediante criptografía o sumas de verificación tratan de asegurar que no ha habido iteraciones indeseadas en la información que se intenta proteger. Algunos ejemplos son los programas que implementan algoritmos como Message Digest (MD5) o Secure Hash Algorithm (SHA), o bien sistemas que utilizan varios de ellos como PGP, Tripwire y DozeCrypt.

5. Sistemas de protección a la privacidad de la información: herramientas que utilizan criptografía para asegurar que la información sólo sea visible para quien tiene autorización. Su aplicación se realiza principalmente en las comunicaciones entre dos entidades. Dentro de este tipo de herramientas se encuentran: Pret Good Privacy (PGP), Secure Socket Layer (SSL) y los certificados digitales.

   Resumiendo, un modelo de seguridad debe estar formado por múltiples componentes o capas que pueden ser incorporadas de manera progresiva al modelo global de seguridad en la organización, logrando así el método más efectivo para disuadir el uso no autorizado de sistemas y servicios de red.

2.4.7. Firewall

Quizás uno de los elementos más publicitados a la hora de establecer seguridad, sean estos elementos. Aunque deber se uno de los sistemas a los que más se debe prestar atención, distan mucho de ser la solución final a los problemas de seguridad.

De hecho, los firewalls no tienen nada que hacer contra técnicas como la ingeniería social y el ataque de insiders.

Un firewall es un sistema (o conjunto de ellos) como lo vemos en la Fig.2.2 de la Pág. 20, que está ubicado entre dos redes y que ejerce la política de seguridad establecida. Es el mecanismo encargado de proteger una red confiable de una que no lo es (por ejemplo Internet).

Puede consistir en distintos dispositivos, que tienen los siguientes objetivos:

1. Todo el tráfico desde dentro hacia fuera y viceversa, debe pasar a través de él.

2. Sólo el tráfico autorizado, definido por la política local de seguridad es permitido.

Como puede observarse, el firewall, sólo sirve de defensa perimetral de las redes, no defienden de ataques o errores provenientes del interior, como tampoco puede ofrecer protección una vez que el intruso lo traspasa.

Algunos firewall aprovechan esta capacidad de que toda la información entrante y saliente debe pasar a través de ellos para proveer servicios de seguridad adicionales como la encriptación del tráfico de la red. Se entiende que si dos firewall están conectados, ambos deben "hablar" el mismo método de encriptación-desencriptación para entablar la comunicación.

2.4.8. Routers y Bridges

Cuando los paquetes de información viajan entre su destino y origen, vía TCP/IP, estos pasan por diferentes Routers (enrutadores a nivel de Red).

Los routers son dispositivos electrónicos encargados de establecer comunicaciones externas y de convertir los protocolos utilizados en las LAN en protocolos de WAN y viceversa.

En cambio, si se conectan dos redes del tipo LAN se utilizan Bridges, los cuales son puentes que operan a nivel de Enlace.

La evolución tecnológica les ha permitido transformarse en computadoras muy especializadas capaz de determinar, si el paquete tiene un destino externo y el camino más corto y menos congestionado hacia el router de la red destino. En caso de que el paquete provenga de afuera, determina el destino en la red interna y lo deriva a la máquina correspondiente o devuelve el paquete a su origen en caso de que él no sea el destinatario del mismo.

Los routers "toman decisiones" en base a un conjunto de datos, regla, filtros y excepciones que le indican que rutas son las más apropiadas para enviar los paquetes.

Algunas medidas básicas:

- Demorar la respuesta ante claves erróneas; aumentar la demora cada vez. Alertar si hay demasiados intentos.
- Registrar todas las entradas. Cada vez que un usuario entra, checar cuándo y desde dónde entró la vez anterior.
- Hacer revisiones periódicas de claves fáciles de adivinar, procesos que llevan demasiado tiempo corriendo, permisos erróneos, actividades extrañas (por ejemplo cuando un usuario está de vacaciones).
- Para los más paranoicos: poner trampas para descubrir intentos de uso no autorizado.

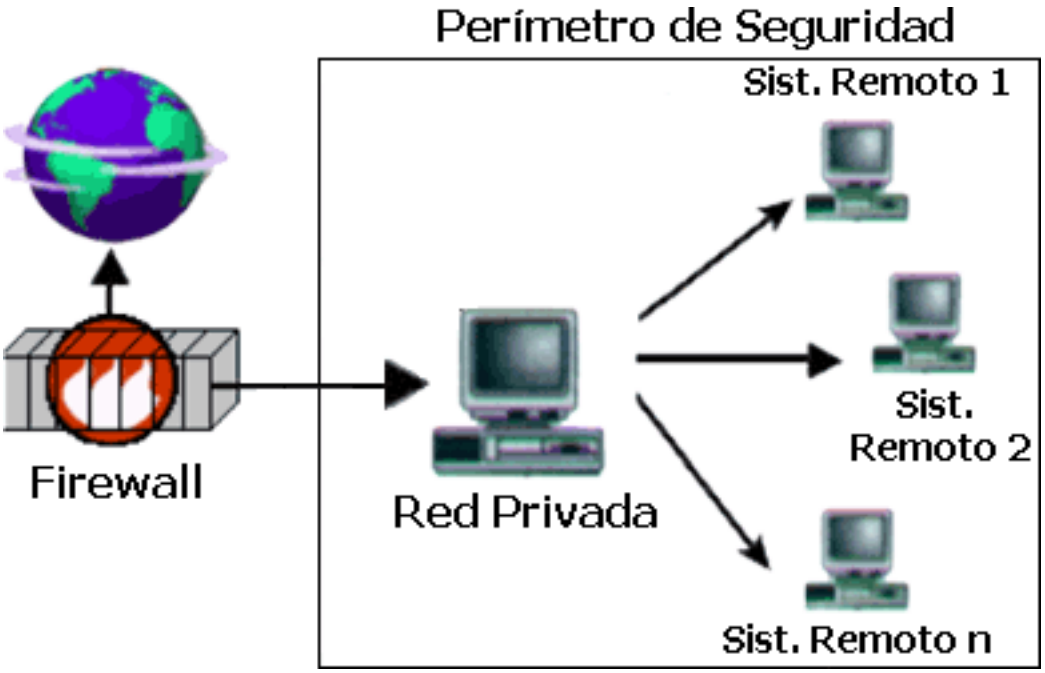


Figura 2.2: Esquema de firewall

Capítulo 3

Seguridad y criptografía

De acuerdo con las definiciones anteriores para que exista seguridad ya sea de la información o informática hay que garantizar las propiedades de: ***confidencialidad, integridad y disponibilidad***. Y es aquí donde se utiliza a la criptografía, ya que mediante el uso correcto de sistemas criptográficos se pretende garantizar las propiedades de confidencialidad e integridad. El siguiente ejemplo ilustra una comunicación.

Primeramente se muestra lo que idealmente es una comunicación normal, en este caso no existe ningún problema de seguridad. El mensaje que se envía se recibe sin alteración alguna.

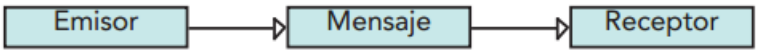


Figura 3.1: Comunicación normal

El segundo caso muestra uno de los problemas más grandes que hay, la interrupción de la transmisión del mensaje, que puede ser ocasionada por fallo del canal o de algún elemento del sistema de comunicación, ya sea de forma natural o intencional. Esto es traducido a un problema de disponibilidad.

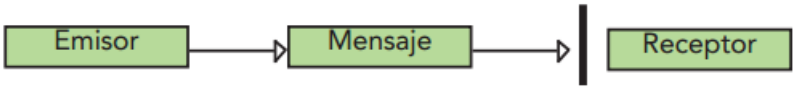


Figura 3.2: Comunicación con interrupción

La interceptación de los datos por un ***intruso*** es algo muy común dentro de las comunicaciones, ya que muchas de las transmisiones son enviadas mediante protocolos que son conocidos por todos y a los mensajes no se les hace ningún tratamiento especial,

en otras palabras, viajan tal cual se generan. Lo único que se hace es escuchar todo lo que pasa por el canal sin alterar nada. Este es un problema de confidencialidad.

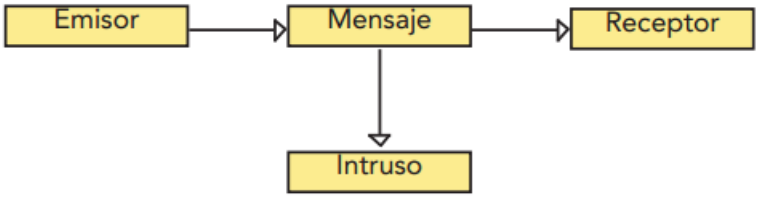


Figura 3.3: Comunicación con interceptación

Otro problema en la comunicación es el problema de la falsificación. Esto se produce cuando el intruso captura un mensaje, se adueña de él y de la identidad del emisor y genera un nuevo mensaje con la identidad del emisor. Este es un problema de integridad y confidencialidad.

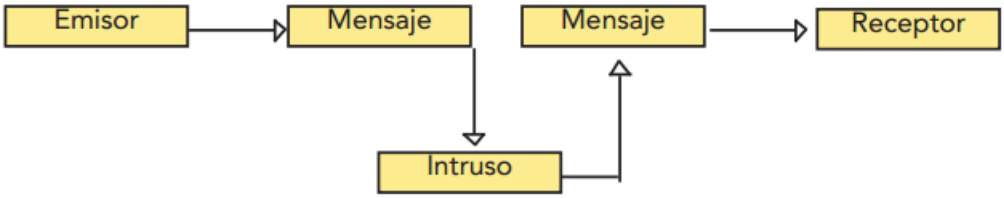


Figura 3.4: Comunicación con falsificación

Finalmente la generación de mensajes se da cuando el intruso genera un mensaje engañando al receptor haciendolo creer que es un emisor válido. Esto se traduce en un problema de integridad.

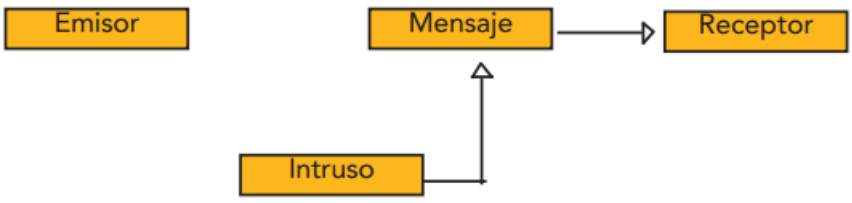


Figura 3.5: Generación de una comunicación falsa

Si pudiéramos de alguna forma evitar los problemas de disponibilidad, integridad y confidencialidad tendríamos un sistema "seguro". Para lograr esto tendríamos que aislar al sistema de los intrusos y hacerlo anti-fallos lo cual es prácticamente imposible. Lo que se hace es crear mecanismos que garanticen en cierta medida las propiedades de disponibilidad, integridad y confidencialidad.

La disponibilidad, generalmente, se trata de solucionar con sistemas redundantes.

La confidencialidad se puede lograr usando un mecanismo que, aunque sea robada la información, permita que no se pueda acceder a ésta o garantice de alguna forma que no se pueda llegar a ella, hasta que pierda su valor.

La integridad es más difícil de lograr y se hace uso varios mecanismos que garantizan que la identidad de un ente que ésta autorizado por el sistema para crear o hacer modificaciones a la información, de tal forma que se puede verificar posteriormente quién creó o modificó la información. Además estos mecanismos permiten ver si la información ya creada ha sufrido o no alguna modificación no autorizada.

Los mecanismos para garantizar la integridad y la confidencialidad se implementan con sistemas criptográficos, de aih la importancia de la criptografía en la seguridad informática en los sistemas actuales.

3.1. Definición de criptografía

La palabra criptografía proviene en un sentido etimológico del griego Kriptos = ocultar, Graphos = escritura, lo que significaría ocultar la escritura, o en un sentido más amplio sería aplicar alguna técnica para hacer inteligible un mensaje.

En su clasificación dentro de las ciencias, la criptografía proviene de una rama de las matemáticas, que fue iniciada por el matematico Elwood Shannon en 1948, denominada: "Teoria de la información". Esta rama de las ciencias se divide en: "Teoría de Códigos" y en "Criptología". Y a su vez la Criptología se divide en Criptoanálisis y Criptografía, como se muestra en la figura 3.6 en la Pag.24

En un sentido más amplio, la criptografía es la ciencia encargada de diseñar funciones o dispositivos, capaces de transformar mensajes legibles o en claro a mensajes cifrados de tal manera que ésta transormación (cifrar) y su transformación inversa (descifrar) sólo pueden ser factibles con el conocimiento de una o más llaves.

En contraparte, el criptoanálisis es la ciencia que estudia los métodos que se utilizan para, a partir de uno o varios mensajes cifrados, recuperar los mensajes en claro en ausencia de la(s) llave(s) y/o encontrar la llaves o llaves con las que fueron cifrados dichos mensajes.

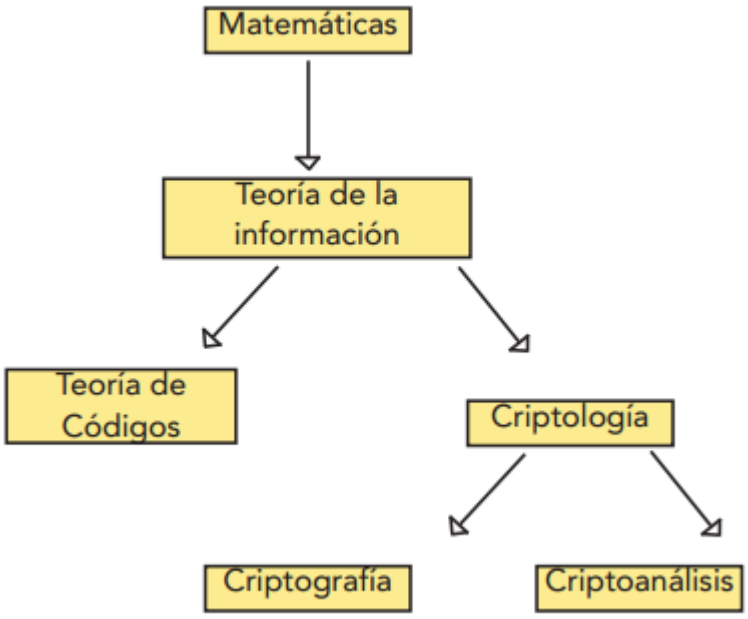


Figura 3.6: Origen de la criptografía

3.2. Clasificación de la criptografía

La criptografía se puede clasificar históricamente en dos: La criptografía clásica y la criptografía moderna.

La criptografía clásica es aquella que se utilizó desde antes de la época actual hasta la mitad del siglo XX. También puede entenderse como la criptografía no computarizada o mejor dicho no digitalizada. Los métodos utilizados eran variados, algunos muy simples y otros muy complicados de criptoanalizar para su época.

Se puede decir que la criptografía moderna se inició después de tres hechos: el primero fue la publicación de la "Teoría de la Información" por Shannon; el segundo, la aparición del estándar del sistema de cifrado DES (Data Encryption Standard) en 1974 y finalmente con la aparición del estudio realizado por Whitfield Diffie y Martin Hellman sobre la aplicación de funciones matemáticas de un solo sentido a un modelo de cifrado, denominado cifrado de llave pública en 1976.

Tanto la criptografía clásica como la moderna se clasifican de acuerdo a las técnicas o métodos que se utilizan para cifrar los mensajes. Esta clasificación la podemos ver en la figura 3.7 en la Pag.25

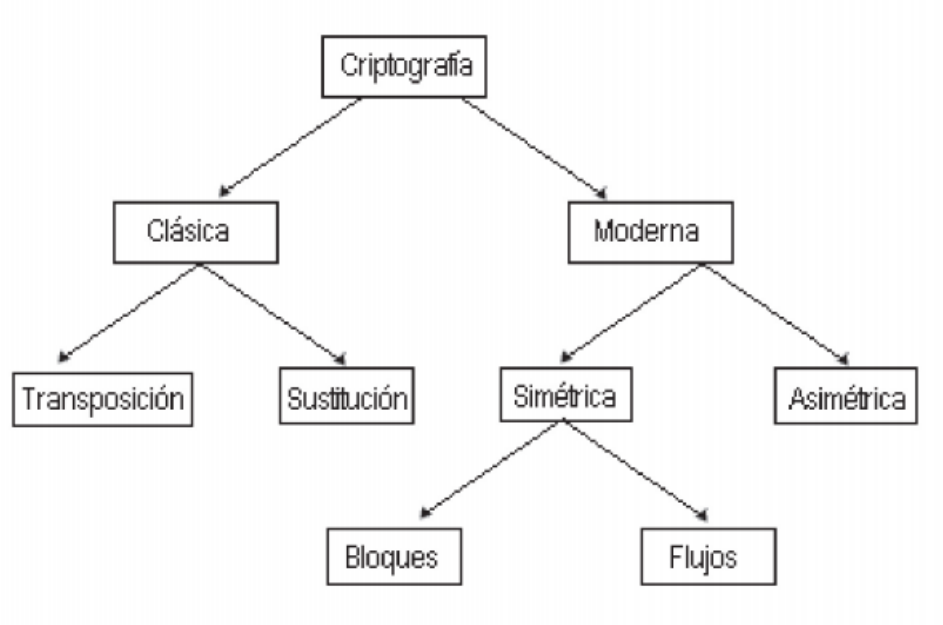


Figura 3.7: Clasificación de la criptografía

3.3. Criptografía clasica

3.3.1. La escítala

Ya en el siglo V antes de Cristo los lacedemonios, un antiguo pueblo griego, usaban el método de la escítala para cifrar sus mensajes. El sistema consistía en una cinta que se enrollaba en un bastón y sobre el cual se escribía el mensaje en forma longitudinal como se muestra en la Figura3.8 en la Pag.25.

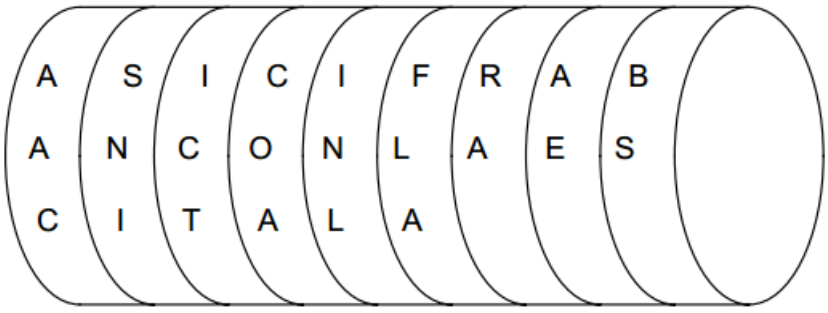


Figura 3.8: Cifrado mediante sistema de escítala

Una vez escrito el mensaje, la cinta se desenrollaba y era entregada al mensajero; si éste era interceptado por cualquier enemigo, lo único que se conseguía era un conjunto de caracteres o letras distribuidas al parecer de forma aleatoria en dicha cinta. Incluso si el enemigo intentaba enrollar la cinta en un bastón con diametro diferente, el resultado obtenido era un conjunto de letras escritas una a continuación de otra sin sentido alguno. Por ejemplo, en el caso de la figura Fig.3.8, la cinta llevará el mensaje M= ASI CIFRABAN CON LA ESCITALA si bien en ella sólo podrá leerse el criptograma C= AACSNIICTCOAINLFLARAAEBS. Para enmascarar completamente la escritura, es obvio que la cinta en cuestión debe tener caracteres en todo su contorno. Como es de esperar, la clave del sistema residía precisamente en el diámetro de aquel bastón, de forma que solamente el receptor autorizado tenía una copia exacta del mismo bastón en el que enrollaban el mensaje recibido y, por tanto, podía leer el texto en claro. En este sistema no existe modificación alguna del mensaje, es decir, éste va en claro desde el transmisor hacia el receptor, por lo que se tratará de un cifrador por transposición.

De esta forma se lograba el objetivo de la confidencialidad, en tanto que la integridad estaba entre dicho y dependía de lo aguerrido y fiel que fuese el mensajero. De estos tiempos tan lejanos se debe la famosa frase de ostenta el "bastón de mando" tan popular entre los políticos, como es de suponer, en aquella épOC no se soltaba por ningún motivo puesto que en él residía la seguridad del sistema de información y la vida política de este pueblo en la antigua Grecia.

3.3.2. El cifrador de Polybios

A mediados del siglo II antes de Cristo, encontramos el cifrador por sustitución de caracteres más antiguo que se conoce. Atribuido al historiados griego Polybios, el sistema de cifrado consistía en hacer corresponder a cada letra del alfabeto un par de letras que indicaban la fila y la columna en la cual aquella se encontraba en un recuadro de 5 x 5 = 25 caracteres, transmitiéndose por tanto en este caso el mensaje como un criptograma. En la Fig.3.9 se muestra una tabla de cifrar de Polybios adaptada al inglés, con un alfabeto de cifrado consistente en el conjunto de letras A,B, C, D y E aunque algunos autores representan el alfabeto de cifrado como los números 1, 2, 3, 4 y 5.

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

	1	2	3	4	5
1	A	B	C	D	E
2	F	G	H	IJ	K
3	L	M	N	O	P
4	Q	R	S	T	U
5	V	W	X	Y	Z

Figura 3.9: Tablas de cifrar de Polybios

Acorde con este método, la letra A se cifrará como AA, la H como BC, etc. Esto significa que aplicamos una sustitución al alfabeto {A, B, C,. . . , X, Y, Z} de 26 letras convirtiéndolo en un alfabeto de cifrado {AA, AB, AC, . . . , EC, ED, EE} de 25 caracteres, si bien sólo existen 5 símbolos diferentes {A, B, C, D, E}.

Ejemplo 3.1. *Usando la tabla del cifrador de Polybios, cifre el mensaje:*
M = QUE BUENA IDEA LA DEL GRIEGO
Solución: C = DADEAE ABDEAECCAA BDADAEAA CAAA ADAECA BBDBBDAEBBCD

El criptograma que se obtiene con este cifrado tiene una extensión de caracteres igual al doble de la del texto en claro, característica que no puede considerarse precisamente como una virtud de éste metodo de cifrado.

3.3.3. El cifrador del César

Unos cincuenta años depués del cifrador de Polybios, en el siglo I antes de Cristo, aparece un cifrador básico conocido con el nombre genérico de cifrador del César en honor al emperador Julio César y en el que ya se aplica una transformación al texto en claro de tipo monoalfabética. El cifrador del César aplica un desplazamiento constante de tres caracteres al texto en claro, de forma que el alfabeto de cifrado es el mismo que el alfabeto del texto en claro pero desplazado tres espacios hacia la derecha módulo n, con n el úmero de letras del mismo. En la Figura3.10 se muestra el alfabeto y por tanto la transformación que utiliza este cifrado por sustitución de caracteres para el alfabeto castellano de 27 letras.

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

Figura 3.10: Alfabeto de cifrado del César para lenguaje castellano

Ejemplo 3.2. *Con el cifrador del César según el alfabeto mostrado en la figura3.10, cifre los siguientes mensajes:*
M_1 = VINI, VIDE, VINCI. (Frase célebre del César: llegué, vi, vencí).
M_2 = AL CÉSAR LO QUE ES DEL CÉSAR.
Solución: Aplicando a cada carácter M_i su equivalente C_i de la tabla de la Figura3.10 se obtienen los siguientes criptogramas:
C_1 = YLPL, YLGL, YLPFL.
C_2 = DÑ FHVDU ÑR TXH HV GHÑ FHVDU.

Al describir el cifrado de César se utilizó un concepto muy usado en las matemáticas y más en criptografía: el módulo.

El módulo es una operación binaria que se realiza en los enteros positivos y se representan de la siguiente forma: $c = amodb$ de tal forma que a,b y c son enteros positivos.

El valor de c al realizar la operación $c = amodb$ es igual al residuo de dividir a entre b. Se puede observar claramente que $0 <= c < b$.

Podemos escribir en forma matemática el cifrado de César de la siguiente forma:

Para cifrar

$C_i = (3 + M_i)mod27$ con $i = 0, 1, \ldots, n; n =$ número de letras del mensaje

donde C_i es la letra cifrada y M_i es la letra a cifrar

el alfabeto comienza con $A = 0, B = 1, \ldots, Z = 26$

Para descifrar

$M_i = (C_i - 3)mod27 = (C_i + 24)mod27$

donde C_i es la letra cifrada y M_i es la letra a cifrar

el alfabeto comienza con $A = 0, B = 1, \ldots, Z = 26$

A partir del ejemplo anterior es fácil apreciar ciertas debilidades en este cifrador como, por ejemplo, la repetición de la cadena de caracteres YL en el criptograma primero y FHVDU en el segundo que entregan demasiadas pistas a un posible criptoanalista.

3.3.4. El cifrador de Alberti

En el siglo XVI Leon Battista Alberti presenta un manuscrito en el que describe un disco cifrador con el que es posible cifrar textos sin que exista una correspondencia única entre el alfabeto del mensaje y el alfabeto de cifrado como en los casos analizados anteriormente. Con este sistema, cada letra del texto en claro podía ser cifrada con un carácter distinto dependiendo esto de una clave secreta. Se dice entonces que tales cifradores usan más de un alfabeto por lo que se denominan cifradores polialfabéticos, a diferencia de los anteriores denominados monoalfabéticos.

Como se aprecia en la Figura3.11, el disco de Alberti presenta en su círculo exterior los 20 caracteres del latín, esto es, los mismos del alfabeto castellano excepto las letras H, J, Ñ, K, U, W e Y, y se incluyen los números 1, 2, 3 y 4 para códigos especiales. Por su parte, en el disco interior aparecen todos los caracteres del latín además del signo & y las letras H, K e Y. Al ser 24 los caracteres representados en cada disco, es posible definir hasta 24 sustituciones diferentes, es decir, dependiendo de la posición del disco interior la cantidad máxima de alfabetos de cifrado es igual a 24. Para cifrar un mensaje, una vez establecida la correspondencia entre caracteres de ambos discos o, lo que es lo mismo, el alfabeto de cifrado, se repasa letra a letra el texto en claro del disco exterior y se sustituye cada una de ellas por la letra correspondiente del disco interior.

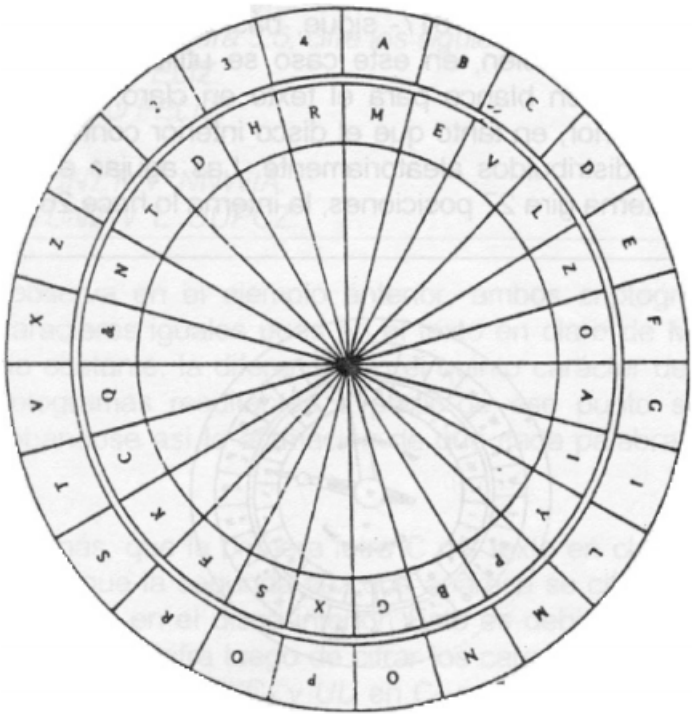


Figura 3.11: Disco cifrador de Alberti.

La innovación que supone este sistema consiste en que el alfabeto de sustitución puede ser cambiado durante el proceso de cifrado, por ejemplo cada K caracteres, simplemente girando el disco interior y por tanto utilizando otro alfabeto de sustitución.

Ejemplo 3.3. *Cifre con el disco de Alberti de la Figura3.11, siendo su posición inicial la de coincidencia entre el número 1 del disco exterior y el signo & del disco interior, el siguiente mensaje:*
M = EL DISCO DE ALBERTI ES EL PRIMER CIFRADOR POLIALFABÉTICO CONOCIDO.

Solución: Desplazamos el disco interior dos espacios en el sentido de las agujas del reloj y leemos el carácter cifrado en el disco interior bajo el carácter correspondiente del texto en claro del disco exterior obteniéndose:
C = VA EOSMP EV HARVXFO VS VA BXOIVX MOLXHEPX BPAOHALHRVFOMP MPYPMOEP.

3.4. Criptografía moderna

La criptografía moderna se puede clasificar en dos grandes grupos: la criptografía de llave secreta o simétrica y la criptografía de llave pública o asimétrica.

3.4.1. Criptografía simétrica

La criptografía simétrica o de llave secreta es aquella que utiliza algún método matemático llamado sistema de cirado para cifrar y descifrar un mensaje utilizando únicamente una llave secreta. Se puede observar en la figura 3.12 que la línea punteada es el eje de simetría: lo mismo que hay de un lado existe exactamente igual en el otro, esto ilustra el porqué se le da el nombre de criptografía simétrica.

Figura 3.12: Criptografía simétrica

Este tipo de criptografía sólo utiliza una llave para cifrar y descifrar, esto es: si uno cifra un mensaje m con una llave secreta k entonces el mensaje cifrado resultante m' únicamente lo va a poder descifrar con la misma llave k. Este tipo de llave conocida como secreta se debe de compartir entre las personas que se desea que vean los mensajes.

Con este tipo de criptografía se puede garantizar la confidencialidad porque únicamente quien posea la llave secreta será capaz de ver el mensaje.

El problema con la criptografía simétrica es que si uno quisiera compartir secretos con m personas, para cada persona tendría que generar una nueva llave secreta y la administración personal de todas las m llaves sería un caos.

Otro problema asociado con este tipo de criptografía es cómo comparto con otra persona de una manera confidencial e integra la llave secreta.

Estos problemas se resuelven de cierta manera con criptografía asimétrica.

Actualmente existen dos métodos de cifrado para la criptografía simétrica, el cifrado de flujo y el cifrado en bloques.

3.4.1.1. Cifrado en bloque

Este tipo de cifrado está basado en el diseño propuesto por Horst Feistel en los años 70.

Diseño de Feistel

Un bloque de tamaño N comúnmente $N = 64$ ó 128 bits se divide en dos bloques de tamaño $N/2$, A Y B. A partir de aquí comienza el proceso de cifrado y consiste en aplicar una función unidireccional (muy difícil de invertir) a un bloque B y una subllave k_1 generada a partir de la llave secreta. Se mezclan el bloque A con el resultado de la

función mediante un XOR. Se permutan los bloques y se repite el proceso n veces. Finalmente se unen los dos bloques en el bloque original. Como se ilustra en la figura 3.13 en la pag.31.

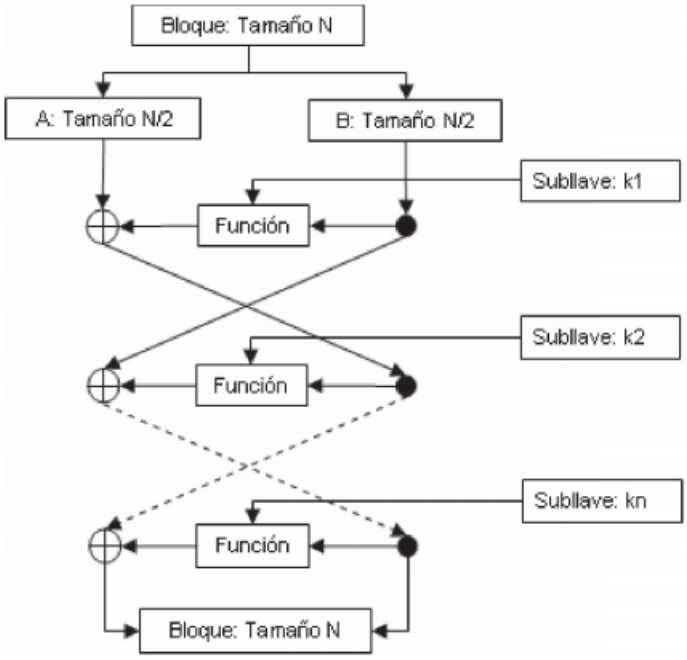


Figura 3.13: Cifrado por bloques de Feistel.

Algunos de los sistemas criptográficos que utilizan esta filosofía son:

Algoritmo	Bloque (bits)	Llave (bits)	Vueltas
Lucifer	128	128	16
DES	64	56	16
Loki	64	64	16
CAST	64	64	8
Blowfish	64	variable	16

Tabla 3.1: Algoritmos de cifrado de bloque

A lo largo de la historia de la criptografía moderna se han usado diversos métodos de cifrado, siendo el más usado el Estantar de Cifrado de Datos por sus siglas en ingles DES (Data Encryption Standard). El problema con este estándar es el tamaño de su llave: 56 bits, para tratar de corregir esto se propuso el tripe DES que únicamente aplica 3 veces el DES, cifrando, descifrando y cifrando con llaves diferentes de tamaño 56 bits, incrementando el tamaño de la llave hasta 168 bits.

A finales de 2001 surge, a partir de un concurso, un nuevo estándar para el cifrado de datos. A este algoritmo conocido como Rindael se le dio el nombre de Estándar

Avanzado de Cifrado o AES (Advanced Encryption Standar). Este algoritmo no sigue la filosofía de Feistel, pero es un cifrador de bloques. Sus características son:

Algoritmo	Bloque (bits)	Llave (bits)	Vueltas
Rijndael	128	128 ó más	flexible

Tabla 3.2: Características del AES

Los cifradores por bloques, como se puede observar en las tablas anteriores, operan con bloques de tamaño fijo, a menudo de 64 o 128 bits. Para cifrar mensajes de mayor tamaño se usan diferentes modos de operación. Estos modos de cifrado son el ECB (Electronic codebook) libro de códigos electrónico, CBC (Cipher-block chaining) cifrado en bloque encadenado, OFB (Output Feedback) cifrado realimentado y CFB (Cipher Feedback) salida realimentada, aseguran la confidencialidad, pero no aseguran la integridad del mensaje.

3.4.1.2. Cifrado de flujo

Este tipo de criptografía se basa en hacer un cifrado bit a bit, esto se logra usando la operación XOR, representada con Å. Se utiliza un algoritmo determinístico que genera una secuencia pseudoaleatoria de bits que junto con los bits del ménsaje se van cifrando utilizando la operación XOR.

Figura 3.14: Criptografía simétrica de flujo.

Algunos ejemplos de este tipo de criptografía son RC4 (usado en redes inalámbricas), A5 (usado en telefonía celular.)

3.4.2. Criptografía asimétrica

Si se observa la figura 3.15, que ilustra la idea de criptografía de llave pública, se puede ver claramente que no existe simetría en ella, ya que de un lado de la figura se cifra o descifra con una llave pública y en otro lado con una privada. De este hecho es de donde la criptografía asimétrica recibe su nombre

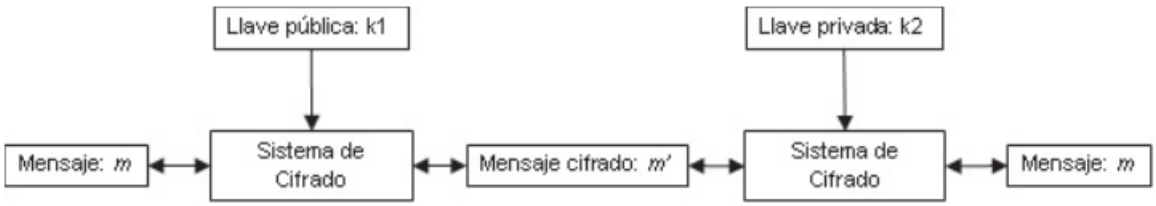


Figura 3.15: Criptografía asimétrica.

Es importante destacar que para este tipo de criptografía lo que se cifra con una llave se puede descifrar con la otra llave. Es decir, uno puede cifrar con la llave pública y descifrar con la privada y viceversa. Esto es de gran ayuda ya que el número de llaves que uno debe de poseer se reduce considerablemente. Si alguien quisiera enviar un mensaje cifrado a n personas, necesitaría saber n llaves públicas una de cada persona, pero si n personas le quieren enviar un mensaje cifrado sólo es necesario que los demás conozcan su llave pública. Así, uno sólo tiene que preocuparse de que la llave pública sea la persona que dice ser. Este es el problema de la criptografía asimétrica, la autenticidad de las llaves públicas.

Algunos ejemplos de este tipo de criptografía son RSA, El Gamal Y Curvas elípticas.

Solución al problema de intercambio de llaves secretas usando criptografía asimétrica: se supone que alguien va a enviar la llave secreta k a una persona para que puedan cifrar entre ellos mensajes. Lo que se hace es que se toma la llave pública de la persona a la que se le va a enviar el mensaje y se cifra con un sistema asimétrico la llave secreta, esto implica que sólo la persona poseedora de la llave privada pueda descifrar lo que se está enviando y con ellos tener la llave secreta, tal y como se muestra en la fig.3.16 en la pag.33.

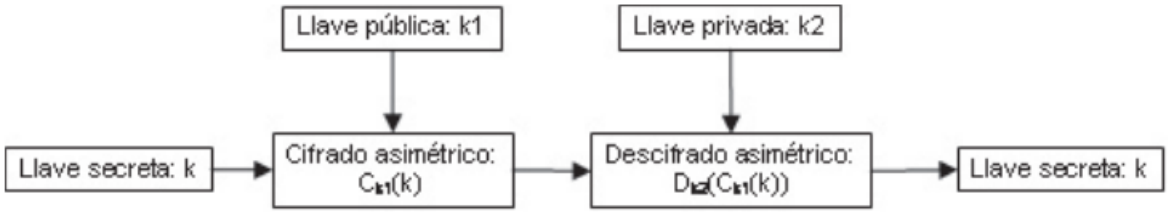


Figura 3.16: Intercambio de llaves secretas.

3.4.3. Algoritmos estandar de cifrado

3.5. Documentos digitales

En criptografía existen diferentes documentos digitales que se usan para garantizar las propiedades de confidencialidad e integridad, estos documentos son la integración de los dos tipos de criptografía: la simétrica y la asimétrica. Al hacer esta integración se compensan las desventajas de los tipos de cifrado y se utilizan las mejores características de cada uno, combinando rapidez del cifrado simétrico con la facilidad de la administración de llaves del cifrado asimétrico.

3.5.1. Firmas digitales

Una firma digital es un documento que permite garantizar la integridad de un documento y se puede relacionar de manera única al firmante con su firma, ya que realiza ésta con la llave privada y únicamente el firmante posee esa llave, ésto se traduce en que se verifica la autenticidad del firmante.

Antes de entrar más en detalle de cómo se realizan las firmas digitales, es importante hablar de una función denominada "Hash" o resumen del documento. Esta función lo que hace es que a partir de un documento de tamaño N bits entrega una cadena de M bits. No hay límite para el tamaño N, pero M siempre es de tamaño constante de acuerdo con el algoritmo usado, normalmente es de 128 o 256 bits. Una de las características de este tipo de funciones es que son unidireccionales, es decir, que debe de ser imposible a partir del resumen del documento encontrar el mensaje original. También deben cumplir la propiedad de dispersión, lo que significa que si se cambia al menos un bit del documento, su resumen debe de cambiar la mitdad de sus bits aproximadamente.

La firma de un documento d se realiza tomando un documento digital, se extrae el resumen del documento $H(d)$ y este resumen se cifra asimétricamente con la llave privada del firmante $Ck_1(H(d))$, esto es lo que vendría siendo la firma digital, ahora hay que ponérsela al documento, para eso se concatenan el documento y su resumen cifrado.

Ahora hay que verificar la firma, para eso se separan el documento d del resumen cifrado. Se descifra asimétricamente con la llave pública k_2 del firmante el resumen cifrado $Dk_2(Ck_1(H(d)))$ obteniéndose el resumen del documento original $H(d)$. Se obtiene el resumen del documento enviado $H(d)'$ se comparan las dos digestiones $H(d) = H(d)'$ y si estos son iguales, se dice que la firma es válida, de lo contrario es inválida. Si la firma es inválida puede deberse a dos causas: una es que se está usando una llave pública que no corresponde con la privada del firmante (problema de autenticación) o la otra es que el documento que se envió fue alterado (problema de integridad). La figura 3.17 ilustra el proceso descrito de firmar y validar la firma digital.

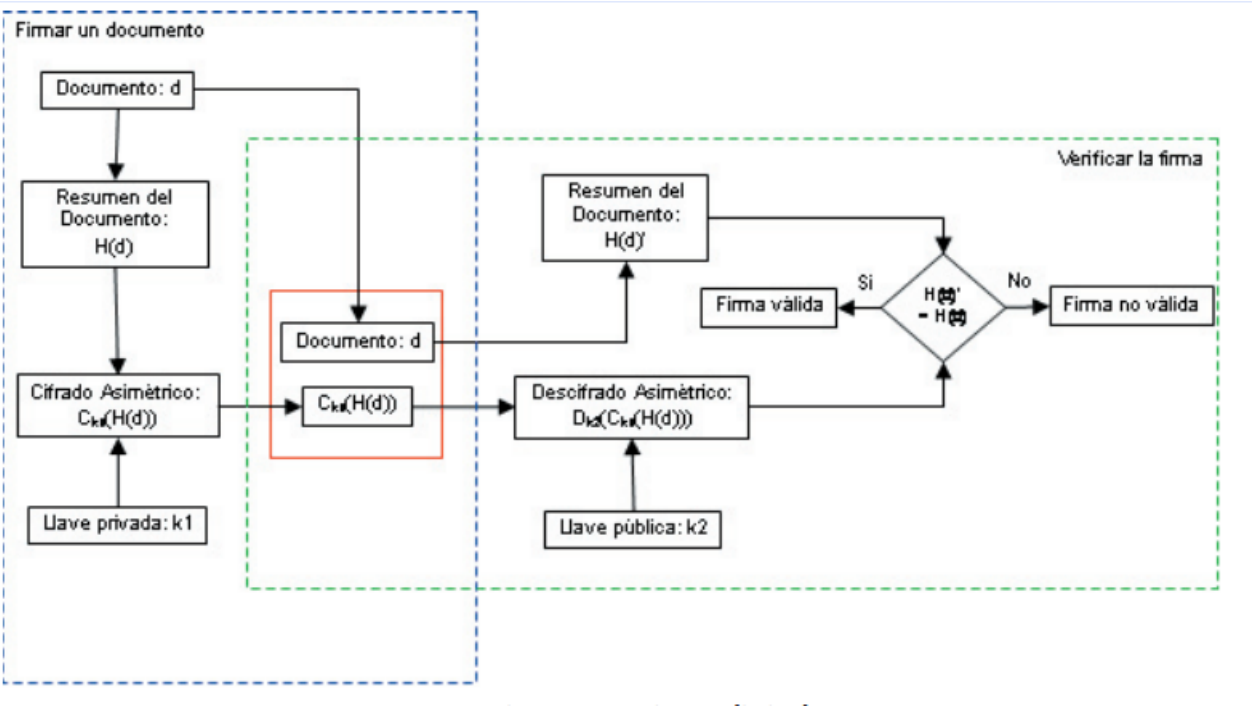


Figura 3.17: Firma digital

3.5.2. Sobres digitales

Con un sobre digital se puede garantizar las propiedades de confidencialidad de un documento. El sobre digital usa criptografía simétrica y asimétrica. Un sobre digital se genera a partir de un documento d y una llave secreta k que se genera de forma aleatoria, se cifra simétricamente $Ck(d)$ el documento d con la llave secreta k, luego la llave secreta k se cifra asimétricamente con la llave pública k_2 de la persona a la que se le va a enviar el sobre $Ck_2(k)$ y finalmente se concatenan el cifrado del documento $Ck(d)$ con el cifrado de la llave secreta $Ck_2(k)$ dando origen al sobre digital.

Para abrir el sobre digital se toma el cifrado de la llave secreta $Ck_2(k)$ y se descifra $Dk_1(Ck_2(k))$ con la llave privada k_1 de la persona a la que va dirigida el sobre, obteniendo la llave secreta k. Con la llave k se descifra el el cifrado del documento $Dk(Ck(d))$ obteniendo así el documento d original.

Ahora bien, se pueden combinar los sobres digitales con las firmas digitales dando lugar a un sobre firmado y así se garantizan las propiedades de integridad, confidencialidad y autenticación.

3.5.3. Certificados digitales

Un certificado digital básicamente es un documento digital expedido por una autoridad de confianza que contiene los datos que identifican al dueño del certificado, su llave

pública, fecha de expedición, fecha de caducidad, los datos de autoridad de confianza y finalmente todo esto está firmado por la misma autoridad.

Los certificados sirven para establecer lazos de confianza entre sistemas o personas, ya que si confían en la autoridad de confianza entonces se puede confiar en la llave pública del dueño del certificado. Tratando así de resolver el problema de relacionar las identidades con las llaves públicas.

Como podemos observar, la criptografía no es la solución a todos los problemas, pero bien usada puede ser de gran ayuda para mantener la seguridad informática.

Capítulo 4

Introducción al caos

4.1. Introducción

El caos es un fenómeno fascinante que puede ser observado en la naturaleza (estado del tiempo y clima, la dinámica de los satélites en el sistema solar, tiempo de evolución del campo magnético de los cuerpos celestes, y el crecimiento de la población en la ecología) y en laboratorio (circuitos eléctricos, láseres, reacciones químicas, dinámica de fluidos, sistemas mecánicos y dispositivos magneto-mecánicos). El comportamiento caótico también ha encontrado numerosas aplicaciones en la ingeniería eléctrica y comunicaciones, la información y las tecnologías de comunicaciones, la biología y la medicina. Esto se debió principalmente a la característica de banda ancha de las señales caóticas, fácil control experimental del caos y todo lo que se consigue con un ejercicio de laboratorio de bajo costo de cualquiera de los circuitos eléctricos o los algoritmos correspondientes si solamente se enfocan a los números de serie. Comunicaciones y aplicaciones de procesamiento de señales de caos, como áreas de interés permanente, se establecieron más o menos desde 1990, después las teorías de la sincronización de caos y control del caos se trabajó en más detalles. Hoy en día, las aplicaciones de ingeniería de sonido de la generación de secuencia cuasi aleatoria, modelando canales de comunicación usando caos, la criptografía caótica, la codificación de imagen digital, y los fenómenos de transporte caóticos en redes complejas representan todas las áreas de investigación permanente con las soluciones de ingeniería comercialmente viables [4] .

Aquí se va tratar de dar una breve introducción a los conceptos y la teoría básica de la dinámica no lineal y el caos de manera que sea mas fácil la explicación del esquema que se presentará mas adelante. Se hará énfasis en explicar las características básicas de las dinámicas complejas y los mecanismos por los que los cambios de los parámetros y condiciones iniciales en los sistemas simples tienen soluciones predictivas y oscilaciones simples en las más complejas.

4.2. Sistemas dinámicos

4.2.1. Concepos básicos

Nos acercamos a un modelo o sistema experimental mediante la definición o selección de un observable. Un observable es generada por un sistema dinámico - conjunto de ecuaciones que describen la evolución dinámica de las cantidades u observables que estamos tratando de modelo y diferentes sistemas dinámicos pueden generar el mismo observable. Los matemáticos suelen insistir en la dimensión más baja del sistema al definir sus propiedades dinámicas genéricas. Como ejemplo, vamos a tratar de averiguar qué ley dinámica unidimensional se realiza con la función observable es decir, $f(x) = e^{\lambda x}$(en general, puede ser cualquier ley N-dimensional. Para $x \in \{1, 2, \ldots\}$ tenemos:

$$f(1) = e^{\lambda}, f(2) = e^{2\lambda}, etc. \tag{4.1}$$

Elegimos dos valores adyacentes para x, K y $k + 1$, por lo que podemos escribir:

$$\frac{f(K+1)}{f(K)} = e^{\lambda} \rightarrow \lambda = ln\frac{f(K+1)}{f(K)} \tag{4.2}$$

donde $K = x \in [1, 2, \ldots]$. Así podemos reconstruir la ley dinámica $\dot{x} = \lambda x$ que genera la funcioń. Ahora, podemos explicar el significado de la relación $\dot{x} = F(x)$. Para un fenómeno observable continuo, o que cambia constantemente, un sistema dinámico es un conjunto de ecuaciones diferenciales ordinarias acopladas que determinan cómo el estado de un sistema evoluciona con el tiempo. Cuando el tiempo es de valor entero, es decir, se observa que el sistema en tiempo discreto, la evolución de un sistema dinámico se rige por un conjunto de ecuaciones diferenciales. Un sistema dinámico continuo puede ser descrito como el conjunto de ecuaciones diferenciales de primer orden:

$$\dot{X}(t) = F(x(t)) \tag{4.3}$$

donde $\dot{x} = \frac{d}{dt}X(t), x \in \mathbb{R}$. El mapeo $F : \mathbb{R}^n \rightarrow \mathbb{R}^n$ rige la evolución del sistema y es llamado campo vector. Este campo vectorial es tal que en cada punto $X_s(t)$ el vector $F(X_s(t))$ es tangente a la curva de la solución $X_s(t)$. En pocas palabras, la ecuación (1) describe un flujo de N-dimensiones. Si tuviéramos que definir un mapa (donde el tiempo es de valor entero), la ecuación (1) se convierte en:

$$X_{(n+1)} = F(x_n), \tag{4.4}$$

donde $n \in \mathbb{Z}$ o $n \in \mathbb{Z}^+$. El vector N-dimensional $X(t) \in \mathbb{R}^n$ representa el estado del sistema y los componentes de $X(t) = (x_1, x_2, x_3, \ldots, x_m)$ son llamados estados variables. Usualmente F depende de un conjunto de parámetros $P = (p_1, p_2, \ldots, p_k), P \in \mathbb{C}^k, \mathbb{C}^k \subseteq \mathbb{R}^n$ pero la mayoría de las veces no es necesario indicar explícitamente esta dependencia. El espacio determinado por x se llama espacio de estado o espacio de fases y se considera que es el espacio euclidiano, pero en general podría ser un colector de N-dimensiones.

El estado del sistema de $x(0)$ ó x_0 cuando $t = 0$ se denomina a las condiciones iniciales y el conjunto de todos los puntos de partida de éste estado se llama una trayectoria o una órbita. Tenga en cuenta que existen diferencias entre las órbitas de un mapa y los de un flujo. Para un flujo, la órbita es una curva continua, pero para un mapa, la órbita es un conjunto de puntos desconectados, como un conjunto de instantáneas estroboscópicos consecutivos de una órbita de un flujo con la misma regla de evolución. (ver Fig4.1 en la Pag.40)

El sistema que nos encontramos anteriormente: $x(t) = \lambda x$ es un ejemplo clásico de un sistema lineal. Resolviendo $\frac{dx}{dt} = \lambda x$,obtenemos $x(t) = X_0 e^{\lambda t}$ y explica por qué se utilizó la función exponencial como ejemplo. Sin embargo, lo que nos interesa es el estudio de sistemas no lineales, simplemente porque la mayoría de los sistemas que observamos, o tratamos de modelo son, de hecho, no lineal: la dinámica de fluidos, dinámica neural, la relatividad general, etc.

El agudo lector ya habrá notado que no se ha dicho si F cambia con el tiempo. Así, se define un sistema dinámico autónomo como un sistema donde $F(x)$ no depende explícitamente del tiempo. Si F es de hecho no autónomo, tal que $F(x, t)$, el sistema debe ser considerado como un sistema de segundo orden, y su análisis está más allá del alcance de este capítulo.

Mapas de Poincaré nos dan la capacidad de analizar el flujo de N-dimensional utilizando el asociado $N - 1$ mapa tridimensional. $N - 1$ dimensiones de Poincaré de un sistema discreto es un mapeo invertible entre los puntos sucesivos de la superficie de la sección obtenido usando $N - 1$ hiperplano tridimensional para intersectar el flujo de N-dimensional en $\mathbb{R}^n$. En otras palabras, el mapa Poincaré mapas de la $N - 1$ coordenadas de la $n-$énesima cruce, a los de la $(n - 1)-$énesimo paso del flujo del sistema continuo. Asumiendo que podemos reconstruir el mapa o aproximadas con una que se conoce, desde el punto de vista del análisis, se beneficiará de:

- Simplicidad en su mayoría de manipulaciones algebraicas del mapa.
- tratar con el sistema de menor dimensión.
- Mapa invertible, si tenemos que iterar hacia atrás en el tiempo

Una propiedad interesante y útil de los mapas de Poincaré es que los multiplicadores característicos del mapa, que corresponde ya sea a un punto fijo o una órbita periódica del flujo de N dimensiones, no depende de la selección de la superficie de la sección S, o las coordenadas locales en él.

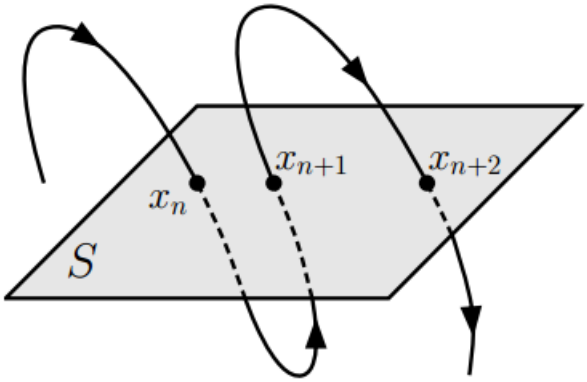


Figura 4.1: La curva representa la órbita de un flujo de arbitrario y los puntos de intersección son las capturas estroboscópicas de la órbita del flujo, que nos da la órbita del mapa. La superficie de la seccińn de Poincaré (denotado S) en este ejemplo es un plano 2-dimensiones que se cruza con el flujo 3-dimensiones arbitraria. Así, desde un flujo dimensional N, obtenemos un mapa $N-1$ dimensiones.

4.2.2. Comportamiento complejo en mapas unidimensionales

Los sistemas discretos unidimensionales han sido extensamente estudiados, ya que son los sistemas más simples, capaces de un comportamiento complejo y caótico. La notación usual para un mapa unidimensional es:

$$x_{n+1} \;=\; f(x_n) \tag{4.5}$$

donde $f : X \to X$ es la asignación del estado espacial a sí mismo. Entonces un mapa unidimensional es un sistema discreto que se puede decir que puede ser **iterado**. Esto significa que para un estado inicial x_0, simplemente aplicacando la regla de evolución una y otra vez, es decir iterando, podemos obtener todos los demas estados futuros del sistema que siguen de las condiciones iniciales específicas: $x_1 = f(x_0),\ x_2 = f(x_1) = f(f(x_0)), ...,\ x_n = f^n(x_0)$, donde f^n es la n-ésima iteración del mapa o equivalentemente la composición:

$$f^n = \underbrace{f \circ f \circ f \cdots \circ f}_{n\ veces} \tag{4.6}$$

Como ejemplo, intruducciendo un valor en la calculadora y presionando el boton seno una y otra ves representa la iteración del mapa $x_{n+1} = seno(x_n)$. Los resultados de la iteraciión n-iésima, se convierten en la entrada para el n +1 iteración y así sucesivamente (un sistema dinámico es como una función recursiva en programación).

Antes de poder definir con precisión lo que es una órbita de un mapa es, es importante que definamos las nociones de homeomorfismos y difeomorfismos (que también

son válidas para sistemas dinámicos en general). Para un mapeo $M : X \to Y$ para ser un homeomorfismo esto debe ser una biyección (todos los elementos del conjunto de salida tienen una imagen distinta en el conjunto de llegada, y a cada elemento del conjunto de llegada le corresponde un elemento del conjunto de salida), ser continua y también M^{-1} debe ser continua. Del mismo modo, decimos que $M : X \to Y$ es un difeomorfismo si M es un homeomorfismo y diferenciable y la inversa de M^{-1} también es diferenciable. Una observación interesante es que si existe un difeomorfismo entre dos sistemas dinámicos n-dimensionales entonces son equivalentes.

Ahora, para un mapa dado $x_{n+1} = f(x_n)$, podemos definir la órbita hacia adelante $\mathcal{O}^+(x)$ del estado x como el conjunto de puntos $\mathcal{O}^+(x) = \{x, f(x), f^2(x) \cdots\} = f^n(x), n \in \mathbb{Z}^+$. Si f es un homeomorfismo podemos definir la órbita completa de $x, \mathcal{O}(x)$, como el conjunto de puntos $f^n(x)$ para $n \in \mathbb{Z}$ y las órbitas inversas $\mathcal{O}^-$, como el conjunto de puntos $x, f^{-1}(x), f^{-2}(x), \cdots$ La razón por la que se disingue entre estos tipos de órbitas es que cuando estudiamos mapas, no siempre podemos seguir las órbitas atrasadas, es decir, atrás en el tiempo. Mapas no reversibles son un ejemplo de tales sistemas y en la siguiente sección proporciona una discusión a fondo de la cuestión de la invertibilidad y cómo se relaciona con el caos.

Una forma muy conveniente de describir las órbitas de los mapas discretos iteradas es la telaraña. Esta técnica gráfica consiste de la superposición del argumento $y = x$ encima del argumento del mapa $x_{n+1} = f(x_n)$. A partir de unos valores iniciales x_0 trazamos una línea vertical a la gráfica del mapa (la parábola en nuestro caso) y desde este punto se traza una línea horizontal en el gráfico de $y = x$. Así se obtienen los resultados de la primera iteraciín del mapa y el punto de partida para el siguiente. Repetimos el mismo procedimiento tantas veces como se necesite. Por ejemplo, véase Fig.4.2 en la Pag.42

4.2.3. Ejemplo: mapa logístico

Uno de los ejemplos más estudiados de un sistema unidimensional capaz de distintos regímenes dinámicos, incluyendo el caos es el mapa logístico:

$$x_{n+1} = rx_n(1 - n) \tag{4.7}$$

donde r es el parámetro control. El mapa logístico representa nada más que un modo idealizado de la población. Es crucial para el comportamiento del mapa el parámetro control r y vamos a examinar los cambios cualitativos en los mapas dinámicos, variando el valor de r.

4.2.4. Problema de invertibilidad

Decimos que un mapa $f(x_n)$ es invertible si para un estado dado x_{n+1} existe una única preimagen x_n tal que $x_n = f^{-1}(x_{n+1})$ donde f^{-1} es la inversa de f. El mapa logístico es claramente no invertible porque existen dos pre-imágenes x_n para cualquier

x_{n+1} arbitrario (excepto para el punto crítico $x = 0{,}5$). De la relación $x_{n+1} = rx_n(1-x_n)$ para x_n obtenemos:

$$x_n = \frac{r \pm \sqrt{r^2 - 4rx_{n+1}}}{2r} \tag{4.8}$$

lo que significa que tenemos dos preimages para x_{n+1} por lo tanto podemos confirmar nuestra declaración acerca de la no invertibilidad del mapa logístico.

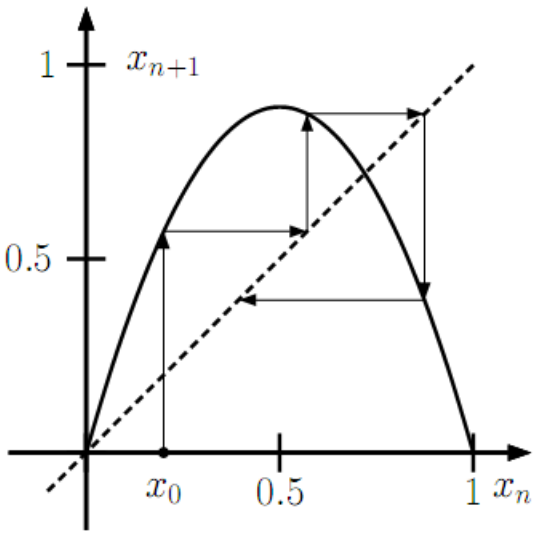


Figura 4.2: Telaraña del mapa logístico para $r = 3{,}565$ y las condiciones iniciales $x0 = 0{,}2$. Representa un alto periodo de la órbita para el mapa logístico y el mapa se encuentra todavía en el régimen periódico. Las tres primeras iteraciones es decir, cómo se crea la telaraña, están marcados con flechas.

Veremos más adelante que, además de la invertibilidad, la dimensionalidad del mapa también pone un límite en el repertorio de soluciones que podemos observar. Del mapa logístico nosotros que la dimensionalidad mínima para mapas no-invertibles para ser caótico es $N = 1$. Para flujos, es decir, sistemas de ecuaciones diferenciales $N \geq 3$ debe cumplirse para que el sistema sea caótico [5].

4.2.5. Puntos fijos y conjuntos de Atracción

Un punto fijo x_* de el mapa $x_{n+1} = f(x_n)$ es tal que $f(x_*) = x_*$. Si existe $n > 0$ tal que $f^n(x_*) = x_*$ y $f^k(x_*) \neq x_*$, donde $0 \leq k \leq n$ entonces se dice que el punto x_* es un ***punto periódico*** de periodo n. El periodo mínimo para que un punto sea periódico se denomina ***periodo primo***. Una órbita de periodo-n se compone de n puntos periódicos y si el estado del sistema pertenece a una órbita periódica, que se alternará entre esos puntos periódicos en sucesión. Existe otro tipo de órbitas uno que ni corresponde a los

estados estacionarios (puntos fijos) del sistema, ni son periódicas. Este tipo de órbitas caóticas, irregulares será discutido en detalle en la siguiente sección.

Encontrar los puntos fijos en un mapa unidimensional es bastante simple, para el mapa logístico sólo tenemos que resolver $x = rx(1-x)$, por lo que las soluciones son:

$$x_{1/2} = \frac{-(1-r) \pm (1-r)}{2r} \tag{4.9}$$

Una forma alternativa de encontrar los puntos fijos de un mapa unidimensional es encontrar los puntos de intersección de su gráfica con la diagonal $y = x$. Un aspecto central de puntos fijos es su carácter: se pueden atraer o repeler, lo que significa que las órbitas cercanas respectivamente convergen o divergen a partir de ellos.Para los mapas iterados discretas, atrayendo puntos fijos son los ejemplos más simples de conjuntos para atraer o atractores (los próximos simples son de periodo n órbitas). Un conjunto para atraer un mapa se describe mejor como un subconjunto cerrado de espacio de fases del mapa, tales que las soluciones para muchos diferentes condiciones iniciales convergen/asíntota a medida que incrementa el tiempo.

4.2.6. Estabilidad de puntos fijos

En términos generales, la estabilidad de un punto fijo depende de la derivada del mapa en ese punto fijo. Para ver esto, se inyecta un pequeño cambio o alteración del punto fijo $x_* = f(x_*)$, que etiquetamos δ_n. Queremos calcular la perturbación en la siguiente iteración, por lo que tenemos $\delta_{n+1} = f(x_* + \delta_n) - x_*$. Mediante el uso de expansión de Taylor se obtiene:

$$\delta_{n+1} = f(x_* + \delta_n) - x_* = f(x_*) + f'(x_*)\delta_n - x_* + O(\delta_n^2) \tag{4.10}$$

Debido a que δ es suficientemente pequeña, el término $O(\delta_n^2)$ no influye en el carácter de la estabilidad por lo que es correcto para que se aproxima a cero. Por lo tanto, la perturbación despues de n iteraciones es: $\delta_n \approx (\mu_*)^n \delta_0$, donde μ_* es el multiplicador del punto fijo:

$$\mu_* = f'(x_*) \tag{4.11}$$

Para $\mid \mu_* \mid < 1$ se dice que el punto fijo x_* es estable y nos referimos a μ como multiplicador del punto fijo. Si $\mid \mu \mid \neq 1$, entonces se dice que el punto fijo es hiperbólico. Por otro lado, para $\mu = 1$ no estamos seguros del carácter del punto fijo y para este valor de μ el sistema sufre un cambio explicaremos en detalle más adelante. La noción de hiperbolicidad también se aplica para los puntos periódicos y se explica en la siguiente sección. Para $\mu = 0$ se dice que el punto fijo es superestable ya todas las perturbaciones se amortiguan más rápido que exponencialmente.

4.2.7. Estabilidad de órbitas periódicas

Vamos a considerar el periodo-p órbita $O(x_p)$ y sabemos que $x_i = f^p(x_i)$ se cumple para $i = 0, 1, \ldots, p-1$ De nuevo, utilizando la misma técnica, se introduce una ligera perturbación a x_i y tomamos ese valor como condiciones iniciales para nuestro análisis: $x_0 = x_i + \delta_0$. A causa de estas perturbaciones, la p-ésima iteración será diferente de x_i por alguna δ_p así que tenemos $x_i + \delta_p = f^p(x_i + \delta_0)$. Una vez más, utilizamos expansión de Taylor para obtener:

$$x_i + \delta_p = f^p(x_i + \delta_0) = f^p(x_i) + (f^p)'(x_i)\delta_0 + O(\delta_0^2) \tag{4.12}$$

Usando de la regla de la cadena de derivados (ya que f^p es una composición de funciones) y no teniendo en cuenta las órdenes superiores de δ_0 (porque δ_0 es suficientemente pequeño), se llega a: $\delta_p = \lambda_p \delta_0$, donde λ_p es:

$$\lambda_p = f'(x_0)f'(x_1)\ldots f'(x_{p-1}) \tag{4.13}$$

y es igual para todos los puntos $x_i, i = 0, 1, \ldots, p-1$ que pertenece a la órbita periodo-p. Cuando seguimos el punto perturbado $x_i + \delta_p$ otra p repite alrededor de la órbita, el resultado es $x_i + \lambda_p \delta_p = x_i + \lambda_p^2 \delta_0$, por lo que podemos generalizar a: $\delta_{np} = \lambda_p^n \delta_0$. Esta ecuación cuantifica la desviación a medida que avanzamos sobre la órbita periódica. Para $\mid \lambda_p \mid > 1$. Esta desviación crece en un factor de λ_p para cada círculo alrededor de la órbita y representa las órbitas periódicas como repelente, que nos alejamos de ella. Cuando $\mid \lambda_p \mid < 1$ cada vez alrededor de la órbita periódica la desviación disminuye, es decir, la órbita actúa como un atractor ya que todas las condiciones iniciales de sus vecinos es asíntota a la misma. Para $\lambda_p = 0$ decimos que la órbita es muy estable, ya que no hay convergencia a la misma, ni divergen de ella. λ_p puede ser referido como el coeficiente de estabilidad o multiplicador para la órbita periódica. Generalizar la discusión de hiperbolicidad, podemos decir con seguridad que un punto fijo es un caso especial de un punto periódico, es decir, su primer periodo es 1. Si el multiplicador $\mid \lambda_p \mid \neq 1$, para un punto de periodo p, entonces se dice que es hiperbólica y si $\mid \lambda_p \mid = 1$ se cumple, entonces ese punto periódico no es hiperbólica y no se puede decir a ciencia cierta si es estable o no. Cuando se cumple esta última condición, un sistema se encuentra en un "punto de inflexión", que es seguido por un cambio en la dinámica.

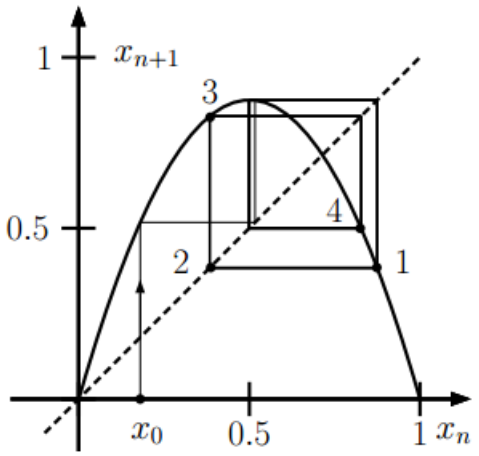


Figura 4.3: 4° periodo estable de la órbita del mapa logístico, para condiciones iniciales $x = 0{,}15$. Un ejemplo de un conjunto asintótico, después de los transitorios se han extinguido, como el mapa se itera aún más, la órbita sucesivamente visitadas los puntos $1 \to 2 \to 3 \to 4$ que también puede ser marcado en la diagonal $y = x$. Tenga en cuenta que para todas las condiciones iniciales, el conjunto asintótico sería el mismo, ya que sólo depende del parametro r de control

4.3. Comportamiento caótico

4.3.1. Características principales de la dinámica caótica

En lugar de entrar en estricta definición matemática del comportamiento caótico, discutiremos con más detalle las características principales y manifestaciones del caos, así como las transiciones de las soluciones caóticas mas regulares. Características principales de un sistema dinámico no lineal, que exhiben caos determinístico para valores dados de los parámetros, son los siguientes:

- ***Dependencia sensible de las condiciones iniciales***, donde pequeños cambios en los valores iniciales de las variables crece en el tiempo, y producir cambios impredecible como cálculos más lejanos, la órbita o la ruta.

- ***Movimiento irregular en la fase espacial***,ilustrado por muy complejos, a veces el ruido como patrones de oscilaciones de las soluciones dentro de un limitado, conjunto compacto. La particularidad del caos es que tales oscilaciones complejas son totalmente reproducibles para la misma precisión numérica en las condiciones iniciales y los valores de los parámetros. Tal comportamiento cuasi-estocástico puede ser calificado por el carácter específico de las medidas asociadas y densidades invariantes.

- ***Cambio cualitativo del carácter de las soluciones*** ilustrada por una o más bifurcaciones posteriores, los cambios estructurales de la fase a la que establecen soluciones caóticas convergen a medida que evolucionamos el sistema en el tiempo. Estos atractores, es decir, atractores caóticos algunas veces no se asemejan en absoluto a la estructura topológica de otras soluciones, por ejemplo, las órbitas periódicas. Esto es resultado de un cambio estructural global del espacio de fases. Compacto, simplemente subconjuntos conectado, a lo largo de ciertos rangos de parámetros someterse a una serie de cambios no regulares en su geometría y la topología, principalmente debido a sucesivas cascadas de estiramiento y plegado. Estos atractores también se llaman atractores extraños, debido a su geometría específica y la estructura auto-similar en diferentes escalas de tiempo.

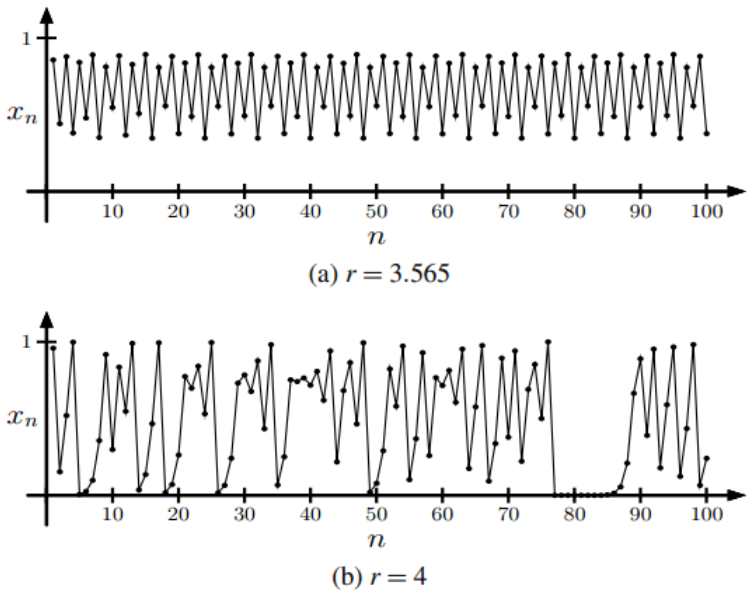


Figura 4.4: Dos reglas distintos para el mapa logístico se muestran a través de 100 iteraciones del mapa logístico (x_n, n), donde n es el número de iteraciones. Los gráficos revelan los diferentes comportamientos propios del mapa. En el régimen periódico, una repetición de los estados, es decir, periodicidad puede verse claramente, mientras que en el régimen caótico presenciamos movimientos no periódico irregulares, al azar.

Comencemos con un ejemplo de la órbita caótica en mapa logístico. Vamos a definir términos y conceptos necesarios, relacionados con el caos, ya que nos encontramos con ellos. Como hemos mencionado anteriormente estamos interesados en ver cómo las órbitas cambian cualitativamente como parámetro de control r es variada y nos explicarán los conceptos de bifurcaciones y de movimiento caótico. Como dijimos anteriormente, el mapa logístico tiene dos puntos fijos:

$$x_1 = 0, x_2 = 1 - \frac{1}{r} \tag{4.14}$$

y los correspondientes multiplicadores de estos puntos fijos son:

$$\mu_1 = r, \mu = 2 - r \tag{4.15}$$

Usando la formula de los multiplicadores podemos determinar como r afecta la dinámica del mapa logístico. Para $r > 1$ podemos ver claramente que x_1 es inestable. Por otro parte, para $1 < r < 3$ tenemos que x_2 es estable desde $|2 - r| < 1$. Así que x_2 es un punto fijo de atracción en este rango de r. Se puede demostrar fácilmente que para $r \in (1, 3)$, las órbitas con periodo $p \geq 2$ no existe, y que cualquier condición inicial $0 \leq x_0 \leq 1$ converge hacia el atractor x_2, por lo que podemos decir que el intervalo $x \in [0, 1]$ es la cuenca de atracción de x_2. ¿Qué sucede para $0 < r < 1$? O ¿ con las condiciones iniciales $x_0 < 0$ ó $x_0 > 1$? Para $r \in (0, 1)$ tenemos que el punto fijo $x = 0$ es estable ($|\mu_x = r| < 1$), de modo que las condiciones iniciales arbitrarias en el intervalo $[0, 1]$ convergerán a ella. Para este valor, el punto fijo $x = 1 - r^{-1}$ es inestable. Además, si $r > 1$, y tenemos las condiciones iniciales negativas $x_0 < 0$ o condiciones iniciales como $x_0 > 1$, entonces $M(x) < x$, por lo que llegamos a la conclusión de que tales condiciones iniciales generan órbitas que tienden a $-\infty$ a medida que avanzamos en el tiempo.

Vamos a discutir los diagramas de la figura4.5 de la Pag.48 por un tiempo. Como se señaló anteriormente, podemos afirmar explícitamente la dependencia de los parámetros de un sistema dinámico, por lo que alternativamente puede representar en la forma:

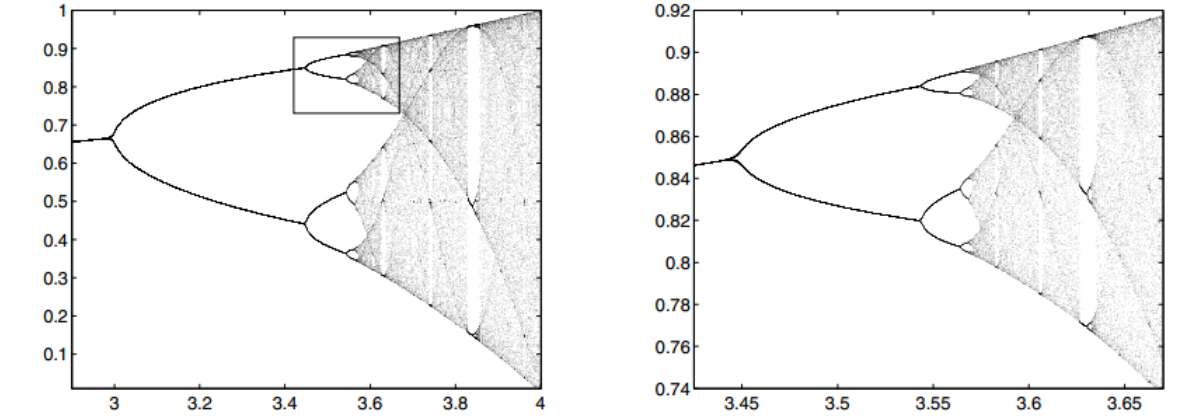
$$(x, r), x \in \mathbb{R}^n, r \in \mathbb{C}^k. \tag{4.16}$$

Para el mapa logístico:

$$(x, r), x \in \mathbb{R}^1 x \mathbb{C}^1. \tag{4.17}$$

por lo tanto, el diagrama de dos dimensiones. Una de las cosas más notables de los diagramas es que al aumentar el parámetro r, las órbitas se dividen de una manera ordenada. Esta división/bifurcación representa un cambio cualitativo en el comportamiento dinámico del mapa. La órbita periodo-1 "bifurca" en una orbita de periodo-2 (diagrama de la derecha). La razón por la que usamos comillas es que mediante bifurcación no nos limitamos a asumir la división, pero cualquier cambio cualitativo en la dinámica de un sistema y una definición apropiada seguirá poco después. Este tipo de diagramas, donde observamos los cambios en el carácter de la solución, como cambio de parámetro, se llaman diagramas de bifurcación. Tenga en cuenta que esta división continúa indefinidamente, y se llama el periodo de duplicación en cascada, es decir, como $n \in \mathbb{Z}, n \to \infty$ se crea una órbita de periodo-2^n . Esta cascada es el principal culpable de la aparición del caos en el mapa logístico. Un vistazo rápico al diagrama de la izquierda, en $r_\infty = 3{,}57$, el diagrama se vuelve borrosa. En este punto, se crean órbitas de período infinito, implicando infinitas bifurcaciones. Este punto de acumulación está estrechamente relacionada con una constante llamada al número Feigenbaum

que se explica en una sección posterior (donde se explica la bifurcación del periodo de duplicación con más detalle).



(a) Bifurcation diagram of the logistic map for $r \in [2.9, 4]$

(b) A zoomed in section of the chaotic region $r > 3.57$ from (a), exposing the self-similar structure of the diagram

Figura 4.5: El periodo de doble bifurcación conduce al caos en el mapa logístico. Desde el propio diagrama podemos ver cómo se comporta el mapa para diferentes valores de r y la transición del régimen periódica al caos: la dependencia sensible de las condiciones iniciales se manifiesta en la asignación de un pequeño intervalo en el dominio completo del mapa, junto con la existencia de órbitas de período infinito (movimiento irregular). Por último, la estructura auto-similar explica la disposición fractal y el proceso de plegado del atractor caótico.

Desde r_∞ a $r = 4$ se dice que el mapa está en régimen caótico. Pero como podemos ver hay más para el diagrama de bifurcación que la simple distinción entre comportamiento periódico y caótico. Hay "ventanas" que son fácilmente perceptibles y se correlacionan con intervalos enteros de $[0, 1]$ se relacionan a sólo 3 valores, ventanas periodo-3. Hay una cascada sin fin de órbitas periodo-3^n y, finamente entrelazado con las órbitas periodo-2^n órbitas. Para los valores de $r > 4$ todas las órbitas tienden al infinito. Ahora nos centramos en mayor detalle de las principales características del caos, como señalamos al comienzo de este capítulo.

4.3.2. Dependencia sensible de las condiciones iniciales

Decimos que el mapa $f : X \to X$ tiene dependencia sensible de las condiciones iniciales si existe $\delta > 0$ tal que, para cualquier $x \in X$ y cualquier vecino de σ de x, existe $y \in \sigma$ y $n \geq 0$ tal que $|f^n(x) - f^n(y)| > \delta$. Para un mapa esto significa que para un punto x dado, existe por lo menos un punto arbitrariamente cerca cuya imagen después

de n iteraciones serán diferentes por δ de la imagen de x. Utilizando el análisis previo de la estabilidad se concluye que una perturbación infinitesimal se comportará como $|\delta x_n| = |\mu^n \delta x_0|$.

4.3.3. Bifurcaciones

La bifurcación es un cambio cualitativo en la dinámica de un sistema dinámico dado como un parámetro de control es variado. El mapa logístico exhibe dos bifurcaciones característicos: una transcrítica y una bifurcación duplicación de periodo, que también se conoce como una bifurcación flip y esta bifurcación es el principal culpable del movimiento caótico en el mapa logístico. Decimos que una bifurcación se produce si el diagrama de fase de un sistema dinámico cambia para algún parámetro r. En otras palabras, si la dinámica del sistema para $r_1 = r - \psi$, $\psi > 0$ ya no son los mismos que los de $r_2 = r + \psi$, $\psi > 0$, decimos que para ese valor distinto de r ocurrió una bifurcación , resultando en diferentes diagramas de fase cualitativo para los respectivos valores de r. Para que bifurcación suceda, un cierto número de condiciones tiene que cumplir, de los cuales los más importantes son los valores de los parámetros. En el mapa logístico, sólo modificamos un parámetro, por lo que podemos decir que las bifurcaciones en el mapa logístico son de primer orden o codimension-1.

Una manera muy conveniente para describir la dinámica de un sistema es a través de diagramas de fase. Este tipo de representación gráfica no requiere el cálculo de iteraciones más altas y el trazado subsiguientes de cada uno de sus gráficos. Para mapas unidimensionales podemos representar las órbitas en $\mathbb{R}^1$ en lugar de $\mathbb{R}^2$ para el diagrama gráfico. En esta línea de fase (obviamente para los sistemas de 2 dimensiones tendríamos un plano de fase) toda la información acerca de todas las iteraciones se puede mostrar al mismo tiempo.

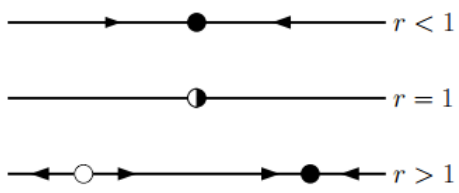


Figura 4.6: Retratos de fases del mapa logístico para varios valores de r. El círculo blanco representa un punto fijo inestable (repelente) y el círculo negro indica un punto fijo estable (atractor). A medida que aumenta el parámetro, podemos ver como ($x_* = 0$) pierde su estabilidad y otro punto fijo ($x_* = 1 - r^{-1}$) es creado. De hecho, esto representa la bifurcación transcrítica, el cual se explica en la siguiente sección.

Además del diagrama de bifurcación del mapa logístico, los retratos de fase ayudarán a ilustrar los diferentes tipos de bifurcaciones que se producen en el mapa y ver como

puntos fijos se "mueven" a lo largo de la curva del mapa resultando en dinámicas diferentes.

4.3.4. Bifurcación transcritica

Cuando hablamos de la estabilidad del punto fijo $x_* = 0$, vimos que para $r = 1$, que denotaremos (r_{tc}), pierde su estabilidad. A medida que aumentamos r desde 0, a $r_{tc} = 1$ la diagonal es tangente a la parábola y hasta este punto el multiplicador $\mu = f'(x_*) = r - 2rx_*$ para $x_* = 0$ es inferior a 1. A medida que aumentamos aún más r, la parábola se eleva por encima de la diagonal y esto da a luz a un nuevo punto fijo $x_* = 1 - r^{-1}$ (que es estable para $1 < r < 3$), a expensas de la estabilidad de $x_* = 0$ que se convierte en inestable. Este es un ejemplo de una bifurcación transcrítico. Vear figura 4.7 como un ejemplo.

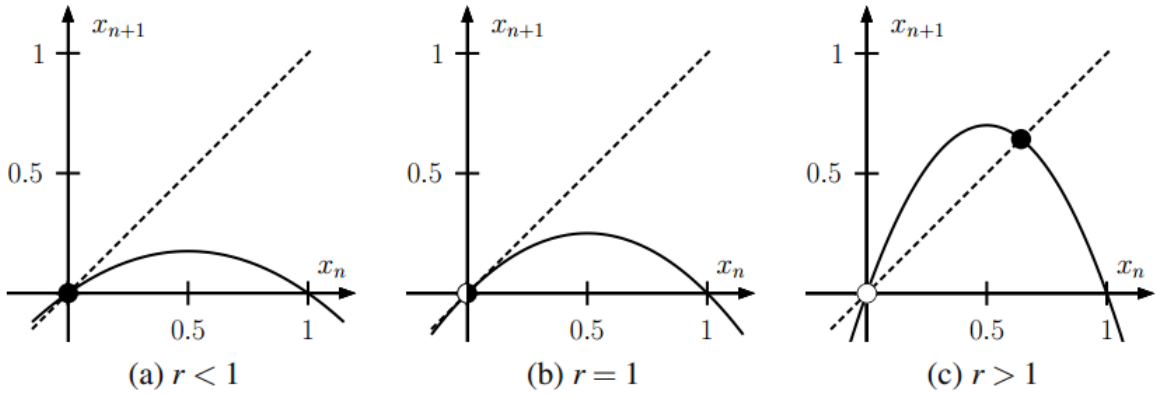


Figura 4.7: Bifurcación transcrítica: (a) para $r < 1$ la parábola está bajo la diagonal $y = x$ y el punto fijo x_* es estable. (b) La diagonal es tangente a la parábola y x_* es no-hiperbólica. (c) Para $r > 1$, la parábola está sobre la diagonal y hay dos puntos fijos: $x_* = 0$ éste pierde estabilidad y hay un nuevo punto fijo estabable $x_* = 1 - r^{-1}$.

4.3.5. Bifurcación de duplicación de periodo

El análisis de la bifurcación duplicación de periodo comenzó cuando se hablo de la aparición de un comportamiento caótico anteriormente. Vimos que para $r < 3$ un régimen periodo-1 está presente, pero en realidad hay dos órbitas de perÃodos-1 en $r = 3$ algo interesante sucede. La órbita de periodo-1 pierde su estabilidad y da lugar a una órbita estable de periodo-2. Esto es llamado como bifurcación de duplicación de periodo. En general, en una bifurcación de duplicación de periodo de cualquier órbita de periodo-n se volverá inestable y dará luz a una órbita de periodo-2n, y permanecerá presente como una ŕbita inestable. Vamos a analizar los coeficientes de estabilidad para $r = r_{pd} = 3$. $\mu_1 =$ y $\mu_2 = -1$. Lo que esto significa es que nos quedamos con dos órbitas,

siendo el primero un órbita inestable periodo-1 y siendo el último una órbieta estable de periodo-2. Esta órbita inestable no desaparece, pero está presente en todos los regímenes dinámicos posteriores. La órbita perÃodo-2 también puede ser considerado como un punto fijo de la doble iteración del mapa logístico es decir:

$$f^2 = f(f(x)) = [rx(1-x)][1-rx(1-x)] \tag{4.18}$$

Así que para calcular los dos puntos periódicos x_1, x_2 tenemos que resolver:

$$x = f^2(x) = r[rx(1-x)][1-rx(1-x)] \tag{4.19}$$

Los dos puntos fijos de f son soluciones de la ecuación, por lo que reducimos de cuarto grado a segundo grado

$$b(x) = \frac{f(f(x))-x}{f(x)-x} = r^2x^2 - (r^2+r)x + r + 1 = 0 \tag{4.20}$$

para lo cual podemos calcular las soluciones:

$$x_{1/2} = \frac{r+1 \pm r\sqrt{(r-3)(r+1)}}{2r} \tag{4.21}$$

Dado que r debe ser positivo para el mapa logístico, que nos deja con $r_{pd} = 3$ para el valor en el que la órbita de periodo-2 aparece y existe la órbita para cualquier valor $r > r_{pd}$ pero má tarde veremos, sus cambios de estabilidad. El multiplicador de la órbita de periodo-2 se puede obtener como el calculando cualquiera de los multiplicadores de los puntos fijos de f^2, ya que son iguales:

$$\mu_{1,2} = [f^2(x_1)]' = [f^2(x_2)]' = f'(x_2)f'(x_1) = r^2(1-2x_1)(1-2x_2) \tag{4.22}$$

que para $r = r > pd = 3$ se evalúa como $\mu = 1$. El mapa logístico sigue actuando de esta manera conforme incrementamos r. Como vimos anteriormente, x_{fix2} es estable para $1 < r \leq r_{pd} = 3$ y como r va más allá de 3, la órbita gradualmente pierde su estabilidad, los multiplicadores cambian de 1 a -1, y en algún punto r_{pd4} se bifurca, dando así nacimiento a una órbita estable de periodo-4. Así que este comportamiento se repite una y otra vez, como una cascada interminable, generando órbitas de manera que para el rango $r_{n-1} < r \leq r_n$, la órbita de perido 2^n es estable, para $n \to \infty$. El rango de r para el cual una órbita de periodo 2^n es estable disminuye casi geométricamente con n. Curiosamente, es una constante y se llama el número Feigenbaum [6].

$$\frac{r_n - r_{n-1}}{r_{n+1} - rn} \to 4{,}669201 = \delta \to \infty \tag{4.23}$$

La secuencia de bifurcación de duplicación de periodo que conduce a una solución caótica, representa sólo uno de los escenarios de transición al caos, o una ruta al caos, también llamada ruta Feigenbaum o escenario.

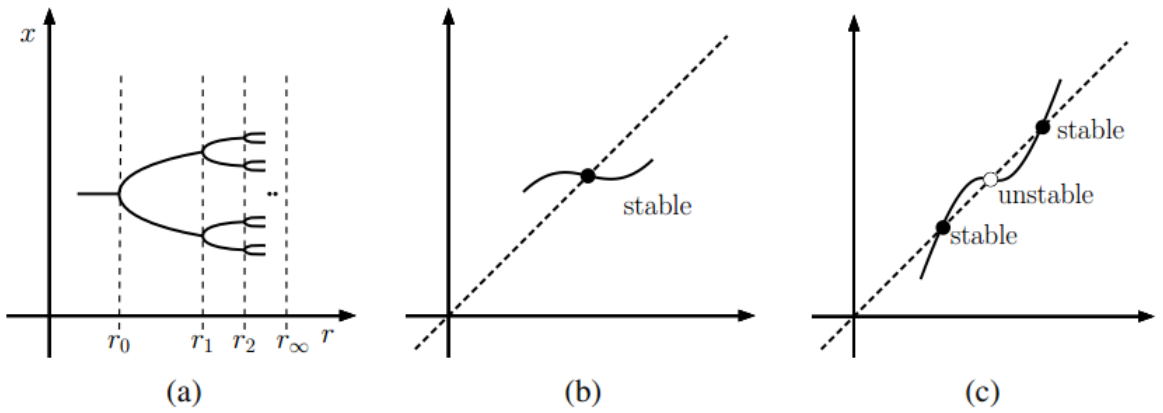


Figura 4.8: (a) Una muestra simple de la bifurcación de duplicación de periodo. (b) y (c) Representan una sección de la segunda iteración del mapa logístico f^2. Se hace más curva mientras r aumenta y corta a la diagonal en tres puntos en lugar de uno. En $r = 3$ el punto fijo $x_* = 1 - r^{-1}$ pierde su estabilidad y da a luz a dos puntos fijos estables. Como el parámetro se incrementa aún más, lo mismo sucede con los puntos fijos también.

Capítulo 5

Esquema de transmisión segura basa en caos con tolerancia a perdida de información

5.1. Introducción

El incremento de dispositivos móviles tales como PDAs, SmartPhones, Netbooks y Notebooks exige la producción de sistemas de cómputo eficientes de alta seguridad para satisfacer las comunicaciones multimedia en tiempo real actuales. Mientras que la seguridad de los estándares de cifrado actuales es importante, es igualmente importante el rendimiento y ahorro de consumo de energía proporcionada por sistemas de cifrado caóticos. Al-Masalha en [7] demostró que el esquema de cifrado caótico propuesto en [8] consume en el promedio de 300 % - 400 % menos de energía y trabajó 250 % más rápido que la aplicación actual del estándar de cifrado avanzado (AES) [9] en diferentes plataformas (Desktops, Notebook, Netbooks y SmartPhones). Estos resultados de rendimiento son alentadores y abren nuevos horizontes para el desarrollo de esquemas de cifrado caóticas competitivos para diferentes plataformas, que van desde las estaciones de trabajo de gama alta hasta los dispositivos móviles.

Sin embargo, la criptografía basada en caos (CBC) todavía se enfrenta al reto principal de la seguridad (robustez a ataques conocidos, la sensibilidad a los cambios de clave y planitext, buen comportamiento estadísticos, gran espacio de claves, etc). Eso depende de un pequeño ciclo Sistemas Dinámicos Caóticos Digitales (DCDS) con distribución de probabilidad degradado y un intervalo reducido del parámetro de control que hace que el sistema sea vulnerable a los ataques [10, 11]. A pesar de las luces de advertencia emitidos en varias obras [12, 13], muchos esquemas propuestos en la literatura ya se han roto [14, 15, 16, 17, 18],causando la pérdida de credibilidad, particularmente en la comunidad de criptografia convencional. Esto significa que la arquitectura de cifrado caótica es en sḿismo otro problema en el sentido de que la difusión y confusión necesaria para esquemas de cifrado no son robustos no se están aplicando correctamente. Otro reto importante para superar es la complejidad computacional de CBC. Es decir,

se debe ser capaz de producir sistemas de baja complejidad con capacidad para hacer frente a las aplicaciones de vÃdeo de alta definición en tiempo real actuales en diferentes plataformas.

Se han propuesto varios métodos para aumentar la seguridad de los esquemas de cifrado caóticas, entre los más importantes son la extensión de la longitud de ciclo (CLE)[19, 20, 21], sistemas caóticos de altas dimensiones-HDCS (incluyendo mapas caóticos acoplados)[22, 23, 24, 25] , múltiples y multidimensionales mapas caóticos [26, 27, 28]y esquemas de perturbaciones [29, 30, 31, 32]. La idea en CLE es incrementar la longitud del ciclo del mapa caótico a costa de aumentar la complejidad del mapa. HDCS por otro lado combina múltiples y/o mapas multidimensionales para aumentar tanto la longitud del ciclo y robustez de todo el sistema (no mapas individuales), por lo tanto, el atacante tiene que romper una red de mapas caóticos. Los esquemas anteriores se pueden combinar con perturbaciones periódicas y/o procesamiento iterativo para aumentar potencialmente la seguridad que proporciona.

Los esquemas mencionados anteriormente aumentan la robustez de los sistemas criptográficos, pero no son la solución para todo. Algunas propuestas son débiles [32](criptoanalizado en [14]), y la mayoría de ellos son demasiado lentos para competir con AES (ver [7] para un rendimiento AES bajo diferentes plataformas). Creemos que el sistema de cifrado en sí, además de HDCS deben trabajar juntos para proporcionar seguridad, es más, la vulnerabilidad de los mapas caóticos deben ser independientes del sistema. Es decir, el propio régimen debe tener la capacidad de trabajar en torno a estas vulnerabilidades cuando se presenten.

En [8], Hasimoto Beltrán presenta un esquema simple basado en una combinación de múltiples mapas caóticos junto con perturbaciones periódicas de plaintext dependientes que rompen la barrera de Gb/s (Gigabit por segundo), superando por mucho el rendimiento (velocidad y consumo de energía) del AES en diferentes plataformas, como se muestra en [7]. A pesar de que el sistema es seguro bajo los ataques conocidos actuales, reacciona lentamente a los cambios de plaintext. Nuestro objetivo en este trabajo es desarrollar un esquema altamente segura con la reacción rápida a los cambios de plaintext sin disminuir el rendimiento general. Para ello, se propone una combinación de múltiples mapas caóticos con un nuevo aspecto dinámico una tabla de búsqueda(LUT) modelo que genera un sistema de cifrado robusto con excelente efecto de avalancha (textos cifrados de plaintext ligeramente diferentes difieren significativamente).

Nuestro sistema es un concepto totalmente diferente de la clase de los esquemas de cifrado caóticos basado en búsqueda [12] propuesto por Baptista en 1998 y variantes [34, 35, 36, 37, 38, 39, 40], que han estado en el ojo del huracán desde hace más de 10 años sin alcanzar todavía una propuesta final efectivo. Baptista [33], propone un esquema de cifrado caótica iterativo en el que un subconjunto del espacio de fases se divide en M subintervalos de igual tamaño que corresponden al número de elementos del alfabeto del plaintext. Cada alfabeto del plaintext se asigna a un subintervalo y se cifra como el número de iteraciones en el mapa logístico necesario para visitar el subintervalo correspondiente que parte de una condición inical privada. El esquema decide si el texto cifrado actual toma en el número mínimo de iteraciones o mantiene la iteración hasta

que alguna condición aleatoria es satisfecho.

El esquema de Baptista fue criptoanalizado por Jakimoski, et.al. [34], Alvarez, et.al [16, 35] y Li, et.a. [36], quienes ponen en evidencia importantes deficiencias en el sistema, tales como los tipos de cifrado pobres debido a su naturaleza iterativa, distribución de texto cifrado no uniforme, y el archivo de texto cifrado expandida. Wong[37] reduce la complejidad computacional e incrementa la robustez del esquema de Baptista disminuyendo el número de iteraciones en el proceso de cifrado y la actualización dinámica de la LUT respectivamente. Wong [38] se dio cuenta de que su LUT dinámica propuesto en [37] no era lo suficientemente robusta contra los ataques criptográficos y aumentó la frecuencia de intercambio de múltiples pares de entradas en la LUT durante el proceso de cifrado. Alvarez et.al [16] reporto la inseguridad del esquema de Wong [38], debido a la independencia entre la LUT y la llave del sistema, lo que facilita el criptoanálisis. Ariffin, et.al. [39], propuso una variante del esquema de Baptista utilizando una matriz de clave secreta con el fin de que sea robusto a ataques de Alvarez [16, 35], en particular, el ataque almohadilla de una sola vez(un tipo de ataque de texto elegido).

Rhouma, et.al. [40], encontró que el esquema de Ariffin, implica una transformación de cifrado no invertible, entre otras debilidades estructurales que dan lugar a más problemas de seguridad. Enfoques adicionales al esquema de Baptista se han propuesto para incrementar su rendimiento y seguridad, como en [41, 42]. En [41], dos mapa logístico están incluidos, uno para cifrar cada bit en el plaintext como en [33] y otra para generar las claves caóticas dinámicas en diferentes sesiones de transmisión. Los principales cambios que se proponen por Formolo ,[42],de 2010, que sustituyó el mapa logístico con un mapa caótico basado en la raíz p^{th} de número racional positivo, eliminar la dependencia de la máquina del mapa utilizando la representación simbólica de las trayectorias caóticas y mejorar los problemas de eficiencia del esquema para su uso comercial.

Una solución eficaz del esquema original de Baptista sigue en el limbo, por lo tanto, el esquema tiene que ser redefinido a partir de cero. Creemos que nuestra propuesta de modelo (ver figura 1), no sólo contribuye a la mejora de los sistemas basados en LUT, sino también retos actuales en CBC mencionamos anteriormente. Ofrecemos una solución de cifrado caótico simple pero seguro, basado en una transformación de plaintext no iterativo y una LUT dinámica poblado con K mapas caóticos evaluados al azar. La seguridad se ve agravada por el hecho de que la transformación de nuestro plaintext se basa en una función biyectiva, creando índices que recuperan trayectorias caóticas al azar de la LUT que pueden pertenecer a diferentes mapas (pueden venir de 1, 2, 3 ... K mapas diferentes) y pueden aparecer sin ningún orden específico de tiempo, lo que aumenta considerablemente la complejidad de los ataques.

La arquitectura de los esquemas propuestos es independiente (en cierta medida) del mapa caótico utilizado en el proceso. Por último, la baja complejidad del sistema lo hace adecuado para las comunicaciones multimedia en tiempo real actuales. Nuestra propuesta tiene las siguientes características generales: a) robustez para cifrado de plaintext, ataques de texto elegido y ataque diferencial, b) reacción rápida a los cambios de bits del plaintext y/o la clave del sistema en caso de ataques, c) la degradación del ciclo

caótico conscientes, es decir, se tiene en cuenta la degradación dinámica sufrida por los mapas caóticos digitales, y d) la selección aleatoria de los mapas y las correspondientes trayectorias caóticas, lo que aumenta naturalmente el ciclo caótico sistema.

5.2. Esquema propuesto

El esquema propuesto es un cifrado simétrico de bloques, compuesto por tres módulos principales: 1) Una transformación del plaintext $T(P)$ biyectiva no iterativa, 2) Una LUT dinámica que contiene L trayectorias caóticas de K mapas seleccionadas al azar (almacenando $TR = L/K$ trayectorias por mapa), y 3) Un esquema de perturbación para evitar ciclos en el sistema dinámico.

La salida del primer módulo es un conjunto de ídices aleatorios distribuidos uniformemente que proporciona un acceso directo a la LUT. Esto significa que las TR trayectorias del i^{th} map$X_{l+n+1} = f_i(X_{l+n}), n = \{1, 2, 3 \ldots, TR\}$ y el propio mapa $(f_i, i = \{1, 2, \ldots, K\})$ se recuperan sin ningún orden en particular en el tiempo (trayectorias recientes pueden ser recuperados antes de trayectorias previas) o secuencia respectivamente, produciendo un enorme número de posibles caminos de cifrado, que a su vez aumenta la seguridad del sistema. En el segundo módulo, la LUT se inicializa de forma secuencial con trayectorias caóticas de K mapas caóticos iterados circularmente; una vez que comienza el proceso de cifrado, la LUT se rellena al azar basado en la transformación plaintext. El tercer módulo es importante para incrementar el ciclo caótico del sistema al perturbar periódicamente el parámetro caótico y orden de evaluación de mapas que participan en el proceso de cifrado.

Nuestro sistema propuesto se puede considerar que consiste de módulos de transformaciones generalizadas para diferentes funciones T y diferentes esquemas conscientes de degradación (perturbación) caótica. Del mismo modo, nuestro sistema no se limita a un mapa caótico específico, sino que debe ser robusto sin importar las propiedades caóticas de la elegida. En las siguientes secciones se discuten cada módulo en detalle, y La figura 3 presenta un algoritmo paso a paso para su reproducción.

5.2.1. Renyi map y tabla de búsqueda (LUT)

Un paso importante en cualquier cifrado caótico digital es la selección del mapa. Los mapas caóticos tienen diferentes comportamientos con respecto a la complejidad, propiedades caóticas (periodo del ciclo, intervalo caótico, ventanas periódicas, etc.), sensibilidad a las condiciones iniciales, la reacción a las perturbaciones de trayectoria, etc, que influyen en la estructura o el comportamiento del sistema de cifrado caótico. De hecho, algunos sistemas se han roto por no considerar las debilidades del mapa caótico elegido. Es deseable el proporcionar alguna independencia entre el sistema de cifrado y el mapa caótico bajo consideración. Afirmamos que el pleno conocimiento del mapa caótico seleccionado no es necesario siempre y cuando el sistema de cifrado proporcione los mecanismos para superar la presencia de punto fijo, ventanas periódicas, la reducción

del periodo del ciclo y e intervalo caótico, etc. con el fin de cumplir con la seguridad de la aplicación correspondiente.

Teniendo esto en mente, la arquitectura del sistema propuesto (no el mapa caótico) es el punto clave en la seguridad del sistema criptográfico. Cualquier mapa puede ser incorporado en el sistema, pero teniendo en cuenta la simplicidad y buenas propiedades dinámicas hemos seleccionado el *Renyi map* defidinido como:

$$X_n = f(X_{n-1}) = \lfloor \lambda * X_{n-1} \rfloor \bmod 2^{PR} \tag{5.1}$$

donde $1 < \lambda \in \mathbb{R}$ es el parámetro y $X \in \{1, 2, \ldots, 2^{PR} - 1\}$ la variable caótica con $PR - bit$ de precisión (32 o 64 bits). Considerando a λ como la suma de un entero (b) y parte fraccional ($\gamma = 2^{-j}$, $j \in \{1, 2, \ldots, PR - 1\}$), podemos reescribir la ecuación 5.1 como:

$$X_n = f(X_{n-1}) = \left(b * k + \left\lfloor \frac{X_{n-1}}{2^j} \right\rfloor \right) \bmod 2^{PR} \tag{5.2}$$

que se puede implementar utilizando aritmética de enteros que requiere una multiplicación, una adición y una operación de desplazamiento a la derecha. Un total de $PR*2^{PR}$ mapas diferentes mapas están definidas en la ecuación 5.2 con máximo periodo alcanzable $2^{PR} - 1$.

Consideramos una red $8 \leq K \leq 32$ Renyi maps, donde los mapas y trayectorias para el proceso de cifrado se seleccionan al azar y se transforman (junto con el plaintext) para convertirse prácticamente en un mapa complejo con distribución uniforme y ciclo extremadamente largo, como se discute en la siguiente sección. K es un número aleatorio obtenido de lo dado B-bits del sistema de claves Key por la siguiente expresión:

$$\begin{aligned} K &= [KS \bmod (MAX_B + 1)] \\ KS &= \sum_{i=1}^{B/8} KEY_8(i) \end{aligned} \tag{5.3}$$

donde KS es la suma de los $B/8$ elementos en el sistema de claves KEY cuando es considerado como un arreglo de números de 8 bits $KEY_8(i)$. El número de K mapas, es forzado a caer en el intervalo $[MIN_B, MAX_B]$ donde MIN_B y MAX_B representan el mínimo y máximo valor de K como una función de la longitud del sistema de claves B ($B \geq 128 bits$ como se muestra en la tabla 1. Para $K < MIN_B$, el sistema caótico se establece en el número mínimo de mapas $K = MIN_B$. En particular, cuando $B = 128$ bits, el número de mapas varía entre 8 - 16, requiriendo $S = \text{piso}\,(B/K)$ bits de KEY para inicializar cada mapa individual (o similarmente, $S/2$ bits por parámetro o variable). como la longitud de KEY se incrementa, se pueden agregar más mapas al sistema, hasta un máximo de 32 cuando $B = 512$ (que es lo suficientemente extremo para mantener el sistema asegurado como se discute en la siguiente sección). Para valores de longitud KEY intermedios, los limites de K se calculan linealmente.

El arreglo K de mapas caóticos es inicializado de la siguiente manera (Ver figura 4):

$$
\begin{aligned}
&X_{i,0} = KEY_{S/2}(2i-1)/2^{S/2}, \\
&\lambda_i = KEY_{S/2}(2i)/2^{S/2}, \\
&i = 1, 2, \ldots, K
\end{aligned}
\tag{5.4}
$$

donde i denota el númer de mapa, $KEY_{S/2}(m)$ representa el sistema de claves como un arreglo de $2KS/2$ bit elementos.

Con el fin de aumentar la sensibilidad del sistema de cifrado a los cambios en las claves del sistema, cada variable caótica $X_{i,0}$ está influenciada por todos las demás a través del siguiente sistema de acoplamiento caótico.

$$
\begin{aligned}
&X_{i,j} = (1-\varepsilon) f(X_{i,j-1}) + \varepsilon H(X_{1,j-1}, \ldots, X_{K,j-1}), \\
&H(X_{1,j-1}, \ldots, X_{K,j-1}) = \frac{1}{k} \sum_{i=1}^{K} X_{1,j-1}
\end{aligned}
\tag{5.5}
$$

donde j es el estado de iteración del mapa actual, $0{,}0001 < \varepsilon \leq 0{,}1$ es el parámetro de acoplamiento y H es la función de acoplamiento o mezcla representada por la media de las variables de iteraciones previas sobre todos los mapas. La salida de la Ec.5.5 después de un número aleatorio de iteraciones $20 \leq RT \leq 50$ iteraciones se convierte en la variable de estado inicial en el proceso de cifrado, es decir, $X_{i,0}, 1 \leq i \leq K$. $RT = 20$ representa el número mínimo de iteraciones necesarias para que una trayectoria caótica perturbada diverja de su trayectoria original cuando la magnitud de la perturbación es $\sim 1/2^{16}$ (donde 16 es el número máximo de bits involucrados in la inicialización de cualquiera de los parámetros caóticos o variables cuando $B = 512$), garantizando que un cambio de bit en KEY, afectará a toda la salida del sistema (ciphertext). Los K mapas creados están iterados circularmente para poblar la LUT como se muestra en la fig.2. Durante el proceso de cifrado, sólo se realizan inserciones aleatorias y recuperaciones.

Como se mencionó anteriormente, sólo el proceso de inicialización se toma de [8], el resto del esquema es totalmente diferente. En el esquema propuesto, hemos introducido una nueva transformación del plaintext T, un acceso aleatorio a la LUT, selección aleatoria de mapas y una perturbación para el reodenamiento del mapa (para las sesiones de transmisión muy largos) que no aparece en [8]. Particularmente, las trayectorias en [8] son independientes del plaintext, secuencialmente evaluados y aplicados directamente al plaintext, en ésta propuesta el plaintext toma el papal principal en la determinación de las trayectorias seleccionadas, índices para la LUT y el esquema de perturbación.

5.2.2. Transformación del plaintext (T(P))

La transformación del plaintext ha sido usado para incrementar la sensibilidad de los sistemas de cifrado a cambios de bit, y consecuentemente la robustez del sistema a ataques comunes tales como: texto plano elegido y ataques a textos cifrados (incluyendo

el ataque diferencial). Algunas transformaciones son de naturaleza iterativa [32, 33], es decir, un mapa caótico es iterado un número aleatorio de veces para obtener un valor final de transformación, que se convierte o bien en el propio texto cifrado o un valor involucrado en el proceso de cifrado. Estos esquemas iterativos reaccionan muy rápidamente a los cambios de bits, pero son demasiado lentos para aplicaciones en tiempo real. Otros esquemas [8], evitan la transformación iterativa del plaintext, mediante la implementación de operaciones adicionales para manejar el cambio de bit del plaintext tal como la retroalimentación espacio-temporal. A pesar de que la difusión de un cambio de bits es más lento que los esquemas iterativo, la seguridad no se ve afectada y la ganancia de velocidad se mejora considerablemente en varios órdenes de magnitud (ver [7]). En este trabajo, nuestro módulo de transformación de plaintext (T(P)) es un generador de índices aleatorios no iterativo, que es extremadamente sensible a los cambios de bits del plaintext o permutaciones de byte. Se requiere una iteración después de un cambio de plaintext(o ataque) para diverger completamente de la trayectoria original.

La transformación $T(P)$ acepta un $PL = B-bit$ plaintext P (misma longitud que la clave del sistema), que se divide en $N = PL/8$ byte índices $P = \{P_1, P_2, \ldots, P_N\}, 0 \leq P_i \leq 255$ que apunta a una ubicación específica en la LUT. Sin transformación del plaintext, el sistema es robusto a cualquier cambio de bit en P (el byte afectado apuntará a una posición diferente de la LUT con una trayectoria diferente), pero no robusto para permutaciones de byte (mismos índices se generan $N!$ veces). Sin embargo, si se hace referencia a cada índice con respecto al índice anterior, cualquier permutación de bytes en P será detectado por al menos un índice en la nueva transformación $P' = \{P'_1, P'_2, \ldots, P'_N\}$ donde $P'_1 = P_1, P'_2 = P_1 + P_2, P'_3 = P_1 + P_2 + P_3$ y $P'_N = P_1 + P_2 + \ldots + P_N$. El número total de referencias a índices modificados cuando permutando P_i y P_j en P para $j > i$ es $j - i$. En el índice j, la suma acumulativa se sincroniza con respecto al original. Los índices de referencia son, de hecho, un conjunto de transformaciones lineales representada por $P'_i = \sum_{j=1}^{i} P_j, i \in \{1, 2, \ldots, N\}$, lo que garantiza que al menos un índice va a cambiar en caso de un cambio de bits o permutación de un byte en el plaintext. Para que la transformación sea biyectiva, se requiere una LUT de tamaño $\leq N * 255$. Nuestro objetivo es encontrar una transformación simple que requiera una LUT pequeña de tal manera que, al menos la mitad de los índices modificados sean diferentes después de cualquier cambio de bits o permutación de un byte en el plaintext. Para lograr esto, pasemos a analizar la combinación de un par de índices en P usando la operación de suma.

$$P' = T(P) = \left\{ \begin{array}{c} P'_1 \\ P'_2 \\ \vdots \\ P'_{N/2} \\ P'_{N/2+1} \\ \vdots \\ P'_N \end{array} \right\} = \left\{ \begin{array}{c} P_1 + P_N \\ P_2 + P_{N-1} \\ \vdots \\ P_{N/2} + P_{N/2+1} \\ P_{N/2+1} + P_{N/2} \\ \vdots \\ P_N + P_1 \end{array} \right\} \tag{5.6}$$

Un cambio de bit en P_i es reflejado en dos índices P'_i y P_{N-i+1} (correspondiendo a la primera y la segunda mitadd de P'). Sin embargo, un cambio apropiado de bits de dos índices o permutación de byte en P no causa cambios en P' cuando $P_i + P_{N-i+1} = P'_i = A$, para un número constante $0 \leq A \leq 510$. Para cada valor A, hay $J = A+1$ diferentes maneras de obtener el mismo resultado variando apropiadamente P_i y P_{N-i+1} entre 0 y 255. J se distribuye simétricamente alrededor de $A = 255$, donde se encuentra el mayor número de posibilidades. Para tener por lo menos 2 índices cambiados en la Ec.5.6 una representacion de $2\sum_{J=1}^{255} J$ numeros es necesario para P'_i, representando una LUT poco prácticco con $\sim 2^{16}$ posiciones.

En su lugar, vamos a considerar un transformación T en 2-D, $T : Z \to Z^2$:

$$P' = T(P) = \begin{Bmatrix} P'_1 \\ P'_2 \\ \vdots \\ P'_{N/2} \\ P'_{N/2+1} \\ \vdots \\ P'_N \end{Bmatrix} = \begin{Bmatrix} (P_1 + P_N, P_1) \\ (P_2 + P_{N-1}, P_2) \\ \vdots \\ (P_{N/2} + P_{N/2+1}, P_{N/2}) \\ (P_{N/2+1} + P_{N/2}, P_{N/2+1}) \\ \vdots \\ (P_N + P_1, P_N) \end{Bmatrix} \tag{5.7}$$

Ahora, cualquier cambio en P puede ser detectado por al menos dos índices del vector en P' con al menos un componente por vector.

La Ec.5.7 satisface la propiedad biyectiva:

Sea $(P, Q) \in Z$ dos plaintexts con las transformaciones correspondientes $(T(P), T(Q)) \in Z^2$, si $P \neq Q$ entonces $T(P) \neq T(Q), \forall P, Q \in Z$.

Demostracion: sin pérdida de generalidad, consideremos sólo un índice de la Ec.5.7, decimos $P'_1 = (P_1 + P_N, P_1)$ y su correspondiente plaintext modificado $Q'_1 = (Q_1 + Q_N, Q_1)$. Tenemos los siguientes 2 casos: i) si $P_1 \neq Q_1$ es fácil ver que $P'_1 \neq Q'_1$ en al menos el segundo componente $\forall(P_N, Q_N)$; y ii) si $P_1 = Q_1$, entonces $P_1 + P_N \neq P_1 + P_N, \forall(P_N \neq Q_N)$, por lo tanto, el primer componente es diferente.

La Ec.5.7 requiere un sistema de coordenadas 2D o una LUT-2D con dimensiones 512x255 (2^{17} posiciones), donde sólo una trayectoria caótica se recupera por índice P'_i. Al igual que en el caso anterior (Ec.5.6), este tamaño de LUT puede causar problemas en pequeños dispositivos como teléfonos inteligentes o PDAs. Sin embargo, es posible hacer uso de sólo una LUT (LUT-1D) al permitir la recuperación de dos trayectorias caóticas por índice P'_i del vector (en lugar de uno como en el caso de LUT-2D), para un total de $2 * N$ trayectorias. A partir de la propiedad biyectiva sabemos que un cambio de plaintext recupera al menos una nueva trayectoria caótica por índice afectado, por lo tanto, la representación LUT-1D es suficiente para recuperar diferentes trayectorias. El número de lugares necesarios en la LUT-1D es ahora 510 (que puede ser mayor, pero no más pequeño, para evitar el efecto envolvente), que representa el número máximo que se obtiene mediante la adición de dos números de tamaño de byte.

Hemos resuelto el requisito de memoria de la LUT, pero todavía permanece el problema de afectar al menos $N/2$ índices de P' bajo un cambio del plaintext. Para

lograr esto, consideremos de nuevo el hecho de que un cambio de bit en el plaintext o permutación de byte modifica al menos dos índices, uno en la primera mitad y la otra en la segunda mitad de P' (perturbando P_1 afecta P'_1 y P'_N). Podemos propagar este cambio de la posición del índice actualmente perturbado $i \leq N/2$ hasta el Índice N^{th} mediante la inclusión de información del índice anterior en forma de retroalimentación al azar de la siguiente manera:

$$P' = T(P) = \left\{ \begin{array}{c} P'_1 = (P'_{1,1}, P'_{1,2}) \\ P'_2 = (P'_{2,1}, P'_{2,2}) \\ \vdots \\ P'_{N/2} = (P'_{N/2,1}, P'_{N/2,2}) \\ P'_{N/2+1} = (P'_{N/2+1,1}, 0) \\ P'_{N/2+2} = (P'_{N/2+2,1}, 0) \\ \vdots \\ P'_N = (P'_{N,1}, 0) \end{array} \right\} = \left\{ \begin{array}{c} V(F + P_1 + P_N), V(LUT(P'_{1,1}) + P_1) \\ V(LUT(P'_{1,2}) + P_2 + P_{N-1}), V(LUT(P'_{2,1}) + P_2) \\ \vdots \\ V(LUT(P'_{N/2-1,2}) + P_{N/2} + P_{N/2+1}), V(LUT(P'_{N/2,1}) + P_{N/2}) \\ V(LUT(P'_{N/2,2}) + P_{N/2+1}) \\ V(LUT(P'_{N/2+1}) + P_{N/2+2}) \\ \vdots \\ V(LUT(P'_{N-1}) + P_N) \end{array} \right\} \tag{5.8}$$

Figura 5.1: Perturbación del plaintext usando una variable de retroalimentación

Donde F es llamado la retroalimentación entre iteración, y representa la trayectoria caótica señalado por el último índice calculado en la iteración anterior (cualquier cambio de plaintext es transmitido a las futuras iteraciones); $LUT(P'_{i,j})$ representa la retroalimentación dentro de la iteración, que es la trayectoria caótica señalado por el último componente $P'_{i,j}$ del vector calculado; y la transformación $V(U) \equiv (U mod L+1)$ confines de la adición de la trayectoria caótica y el correspondiente byte(s) del plaintext para el intervalo $[0, L]$, donde $L \geq 510$ es el tamaño de la LUT. El operador módulo no afecta a la seguridad del sistema ya que $[(B = V(LUT(A) + P_1 + P_{N-i+1})), V(LUT(B) + P_1)]$ no puede producir la misma salida para diferentes valores de P_i y P_{N-i+1} y trayectorias fijas A y B (el mismo índice no se puede alcanzar por el efecto envolvente cuando $L \geq 510$).

La adición de la trayectoria caótica $LUT(S)$ para $0 \leq S \leq L$ como una retroalimentación aleatoria tiene dos funciones: a) que perturba la información real del plaintext por variable aleatoria no controlable para el atacante (el valor $LUT(S)$ no se puede cambiar directamente a la voluntad); y b) que proporciona retroalimentación al azar para futuros índices a través de S, el cual es la posiciń del índice aleatorio calculada previamente, por el cual los cambios de plaintext se propagan durante todo el proceso de cifrado.

Si un simple cambio en P afecta a P'_i, $i \leq N/2$, que se propaga a todos los índices $P'_j, i < j \leq N$ para un número total de $(N - i + 1) > N/2$ índices afectados. Esta es la razón por la que sólo uno de los componentes se utiliza para los índices $P'_m, m > N/2$.

Cada índice aleatorio P' recupera un total de $N + N/2$ trayectorias caóticas de la LUT, que se reducen a $N/2$ trayectorias para el proceso de cifrado (las últimas $N/2$ trayectorias que son sin duda afectadas por cualquier cambio de plaintext):

$$X_i = LUT(P'_i), N/2 < i \leq N \tag{5.8}$$

Una vez que una trayectoria caótica es recuperado de la LUT, es remplazado por una nueva trayectoria desde el siguiente mapa caótica secuencialmente programada. Si n ńdices aleatorios consecutivos apuntan a la misma ubicaciń de la LUT, van a utilizar diferentes trayectorias caóticas de diferentes mapas caťicos (la misma trayectoria nunca se reutiliza). Además, un atacante no puede dirigir su ataque a una ubicación específica de la LUT, porque trayectorias caóticas no controlables están involucradas en el cálculo de cada índice.

La suma de una trayectoria caótica de 32 o 64 bits $0 < LUT(P'_{i,j}) < 2^{32,64}$ a bytes aleatorios de los plaintexts $0 \leq P_i \leq 255$ en la Ec.5.8 no es accidental; nuestro objetivo es índices aleatorios P' distribuidos uniformemente. Para ello primero vamos a analizar la distribución de la $LUT(P'_{i,j})$ (trayectorias involucradas enn el cálculo de P') usando 12 mapas caóticas para los diferentes formatos de de plaintext multimedia: audio sin comprimir, imagen sin comprimir y vídeo comprimido. Independientemente de la distribución del plaintext P_i, la $LUT(P'_{i,j})$ alcanza una distribución uniforme. Cuando se añade $y = LUT(P'_{i,j})$ al plaintext $p = P_i$, la nueva distribución se convierte en la suma de dos variables aleatorias independientes representados por la convolución de sus funciones de distribución correspondientes:

$$f_{Y+P}(a) = \int_{-\infty}^{\infty} f_Y(a-p) f_P(p) dp \tag{5.9}$$

Dado que la magnitud de la $LUT(P'_{i,j}) >> P_i$, $f_P(p)$ actúa como un filtro pasa bajo sobre la distribución dominante $f_Y(y)$ produciendo una distribución uniforme incluso más plano. Cuando la operación módulo se aplica $V(LUT(P'_{i,j}) + plaintext)$, la distribución final de la $f_{Y+P}(a)$ (que representa P' en la Fig.5.1) sigue siendo uniforme (estamos limitando los valores aleatorios $[LUT(P'_{i,j}) + plaintext]$ distribuidos uniformemente a lo largo de la LUT). Índices aleatorios uniformes permiten el acceso justo a la LUT y en consecuencia a los mapas caóticos durante el proceso de encriptación (incrementando la seguridad del sistema). Veremos en la siguiente sección que la selección de la trayectoria aleatoria de la LUT aumenta considerablemente la longitud del ciclo del criptosistema.

Vale la pena señalar que el plaintext P en la Fig.5.1 representa la información del plaintext anterior para que el sistema sea invertible. Desde el plaintext previo se generan las trayectorias caóticas para cifrar el plaintext actual, el esquema sólo requiere de una iteración después del cambio de plaintext para diverger completamente de la secuencia original de cifrado.

Dependiendo de la precisión de la CPU (PR) la LUT puede almacenar trayectorias caóticas en 32 y/o 64 bits. El espacio total de la memoria de la LUT para $L = 510$, plaintext de 128-bit ($N = 16$) y CPU de 32 bits es 2Kb, que es con mucho menor que la requerida para la LUT-2D.

5.2.3. Esquema de degradación caótica consciente

A diferencia de su contraparte continua donde la longitud del ciclo puede ser infinito, los sistemas dinámicos caóticos digitales son de ciclo corto para casi toda trayectoria

caótica [10]. Hasta ahora, hemos hecho uso de un par de trucos para aumentar la longitud del ciclo del sistema de cifrado: a) K mapas están involucrados en el proceso de cifrado, con K desconocido y b) recuperación aleatoria uniforme de las trayectorias de la LUT basado en los datos de entrada y las trayectorias caóticas. Al considerar K mapas con ciclos correspondientes $cl_1 \geq cl_2 \geq \ldots \geq cl_k$, la longitud del ciclo del sistema CL para los mapas evaluados secuencialmente (sin acceso aleatorio a la LUT) con efecto envolvente (después del K^th mapa el esquema continúa con el primer mapa) si se encuentra entre $cl_1 * K \leq CL \leq (cl_1 * cl_2 * \ldots * cl_k) * K$. El CL mínimo se alcanza cuando todos los ciclos tienen la misma magnitud, y el máximo se consigue cuando los ciclos son distintos y primos. CL se puede mejorar aún más por el hecho de que las trayectorias en la LUT se recuperan al azar después de una distribución uniforme (truco b). Para una LUT de longitud L poblada con K mapas ($L > K$) y $TR = L/K$ trayectorias por mapa, hay $\sim D = L^M$ diferentes formas de seleccionar M trayectorias, siempre y cuando $TR < M$. D es de hecho una simplificación del problema real ya que el número de trayectorias por mapa varía ligeramente de iteración a iteración, es decir, si una trayectoria del mapa j se recupera de la LUT, se sustituye inmediatamente por otra trayectoria que puede o no venir del mismo mapa j. D cuenta las diferentes formas en las que puede seleccionar un trajectoria a partir de una posición especificada en la LUT, sin tener en cuenta que el valor real de dichas posiciones cambia con el tiempo, por lo que la estimación D es menor que el valor real.

Bajo este esquema (trayectorias seleccionadas al azar), teniendo cada mapa simple en ciclo no implica que el criptosistema también está en el ciclo; la extracción aleatoria de trayectorias inhibe o al menos difiere del sistema al caer en un cíclo permanente (ciclo que se repite siempre). La tabla III muestra una comparación de la longitud del ciclo entre 4 y 3 secuencialmente y mapas iteradas al azar llamados 0, 1, 2 y 3 (de hecho, es ólo un mapa con 4 condiciones iniciales diferentes). Para los mapas iterados secuencialmente, sólo se necesitan 12 iteraciones para caer en el ciclo, mientras que para los mapas iterados al azar no se detecta ningún ciclo permanente de manera clara, sólo patrones cortos repetitivos no estacionarios. La selección de un mapa aleatorio es una forma natural para extender la longitud de ciclo del sistema. Sin embargo, como el sistema mantiene la iteración durante mucho tiempo, el patrón corto repetititvo se hace más largo y asintóticamente más cerca de un ciclo permanente (esto puede ocurrir cuando el número de mapas es pequeño, mapas ciclados cortos y grandes archivos de plaintext), lo que aumenta la vulnerabilidad del sistema (mismas trayectorias se vuelven a utilizar para la encriptación de diferentes plaintexts).

Para evitar el escenario anterior, perturbamos periódicamente el orden de evaluación y parámetros (y / o variable) y remplaza los primeros k mapas ($MIN_B \leq\leq k \leq K$) en la matriz reordenada final con nuevos parámetros perturbados $\lambda_i = \lambda_i + \delta, \delta \ll \lambda_i$. La trayectoria caótica LUT[0] está recuperado mediante módulo 5 para conseguir k. Un ejemplo del proceso de perturbación para $K = 5$ y $K = 3$ se muestra en la Fig. 6. El ciclo de perturbación se basa en una estimación muy conservadora de $K * cl$ para $1000 \leq cl \leq 5000 (cl \leq 2^{PR} - 1$, donde PR es el bit de precisión del CPU).Con esta opción, la periodicidad de la perturbación pasa a fluctuar entre 8000 y 160000

iterarioness dependiendo de K (véase la Tabla I)

El esquema de perturbación anterior no afecta al rendimiento del sistema de cifrado por las siguientes razones: 1) toda la permutación se realiza gradualmente en $K-1$ iteraciones (una permutaciń por iteración), 2) no se requiere más memoria, 3) la permutación es aplicado durante largos periodos de tiempo, y 4) los valores aleatorios cl al azar y la primera perturbación se toman de las trayectorias caóticas situados en P'_N, que es el último índice calculado en la iteración actual (ver eq.7). Mediante el uso de P'_N garantizamos que cualquier cambio en P afecta el orden en el que se evalúan los mapas (no solo el contenido de la LUT).

5.2.4. Esquema de cifrado/descifrado

Siguiendo la Fig.1, un plaintext P de N-byte transformado en N índices aleatorias P'(Fig.5.1) que generan $N/2$ trayectorias caóticas de $PR-bit$ (Ec.5.8). Sabemos que después de un cambio de plaintext las última N/2 trayectorias caóticas cambiarán de forma independiente de la magnitud y ubicación del cambio, por lo tanto, sólo tenemos en cuenta este conjunto de trayectorias para el proceso de cifrado. El sistema es capaz de cifrar bloques de plaintext de $PL \geq 128$ bits (independientemente de la longitud de la llave B del sistema), con incrementos en múltiplos de 16 bits para producir un número par de índices aleatorios N como se muestra en Fig.5.1. Sin embargo, en nuestras pruebas sólo se considera $PL = B = 128bits$ (N = 16 índices aleatorios) procesados bajo dos diferentes arquitecturas de CPU, $PR = \{32, 64\}$ bits. La ecuación de cifrado para la i^{th} iteración se representa por:

Para arquitectura de 32 bits:

$$\begin{aligned} C_l^{(128,32)} &= C_{l,1} \copyright C_{l,2} \copyright C_{l,3} \copyright C_{l,4} \\ C_{l,1} &= (P_{l,1} + XI_{l,9}) \oplus XI_{l,10} \oplus P'_9 \oplus P'_{10} \\ C_{l,2} &= (P_{l,2} + XI_{l,11}) \oplus XI_{l,12} \oplus P'_{11} \oplus P'_{12} \\ C_{l,3} &= (P_{l,3} + XI_{l,13}) \oplus XI_{l,14} \oplus P'_{13} \oplus P'_{14} \\ C_{l,4} &= (P_{l,4} + XI_{l,15}) \oplus XI_{l,16} \oplus P'_{15} \oplus P'_{16} \end{aligned} \tag{5.10}$$

Para arquitectura de 64 bits:

$$\begin{aligned} C_l^{(128,64)} &= C_{l,1} \copyright C_{l,2} \\ C_{l,1} &= (P_{l,1} + XI_{l,9} \oplus XI_{l,10}) \oplus (XI_{l,11} + XI_{l,12}) \oplus P'_9 \oplus P'_{10} \\ C_{l,2} &= (P_{l,2} + XI_{l,13} \oplus XI_{l,14}) \oplus (XI_{l,15} + XI_{l,16}) \oplus P'_{11} \oplus P'_{12} \end{aligned} \tag{5.11}$$

donde $C^{(128,PR)}$ representa un texto cifrado de 128 bits con una arquitectura de CPU de $PR = 32, 64$ bits, $C_{l,i}$ es un cifrado intermedio de PR bits, $\oplus$ el operador XOR, © el operador de concatenación, $XI_{i,j}$ es la versión entera de las trajectorias caóticas en Ec5.8. Se puede observar en la Ec.5.10 y Ec.5.11 que algunos índices P'_i de la LUT, también están involucrados en el proceso de cifrado. Como las trayectorias caóticas no ocupan todo el intervalo de la unidad (0,1) para $\lambda < 4$ (nuestro caso),los

bits menos significativos se mantienen sin cambios una vez que el mapa caotico está en la trayectoria cíclica; el uso de P_i' mantiene los $log_2(L)$ bits menos significativos (donde L es la longitud de la LUT) cambiantes de acuerdo con una distribución uniforme (hay que recordar que la distribución de P es uniforme). Las Ec.5.10 y Ec.5.11 parecen simples de romper, veremos mas adelante de que es todo lo contrario, su robustez a los ataques es realmente alto.

Las ecuaciones para el descifrado correspondiente se pueden escribir como:

Para arquitectura de 32 bits:

$$\begin{aligned} P_l^{(128,32)} &= P_{l,1} \copyright P_{l,2} \copyright P_{l,3} \copyright P_{l,4} \\ P_{l,1} &= [C_{l,1} \oplus P'_9 \oplus P'_{10} \oplus XI_{l,10} - XI_{l,9}] \\ P_{l,2} &= [C_{l,2} \oplus P'_{11} \oplus P'_{12} \oplus XI_{l,12} - XI_{l,11}] \\ P_{l,3} &= [C_{l,3} \oplus P'_{13} \oplus P'_{14} \oplus XI_{l,14} - XI_{l,13}] \\ P_{l,4} &= [C_{l,4} \oplus P'_{15} \oplus P'_{16} \oplus XI_{l,16} - XI_{l,15}] \end{aligned} \tag{5.12}$$

Para arquitectura de 64 bits:

$$\begin{aligned} P_l^{(128,64)} &= P_{l,1} \copyright P_{l,2} \\ P_{l,1} &= [C_{l,1} \oplus P'_9 \oplus P'_{10} \oplus (XI_{l,11} + XI_{l,12}) - (XI_{l,9} + XI_{l,10})] \\ P_{l,2} &= [C_{l,2} \oplus P'_{11} \oplus P'_{12} \oplus (XI_{l,15} + XI_{l,16}) - (XI_{l,13} + XI_{l,14})] \end{aligned} \tag{5.13}$$

Todas las operaciones de adición en las ecuaciones de cifrado y descifrado se calculan módulo $2^{PR=\{32,64\}}$

Capítulo 6

Pruebas y resultados

El rendimiento del esquema propuesto es analizado a través de las siguientes propiedades del criptosistema: 1)independencias estadística entre plaintext-ciphertext, ciphertext-ciphertext y la distancia media de Hamming entre iteraciones de ciphertext adyacentes, 2) sensibilidad a la llave del sistema y los cambios de plaintext y 3) rendimiento del criptosistema propuesto. El esquema se aplica a varios formatos de plaintext como: audio, imagen y video comprimido con propiedades estadísticas muy diferentes(distribución de probabilidad). Los siguiente ajustes se consideran en el experimento: una llave del sistema de longitud B = 128 bits para el procesamiento de bloques de plaintext PL=128 bits por iteración, RT=25 y $\epsilon = 0{,}0015$ correspondientes al número aleatorio de iteraciones y el factor de acoplamiento caótico en Ec.5.5 respectivamente, $K = 12$ mapas caóticos, LUT de 2160 posiciones (cualquier número mayor que 510), PR = 32 bits (arquitectura de la CPU), y el orden de mapa y el parámetro de perturbación se realiza cada $K * cl = 96000$ iteraciones (suponiendo una duración media del ciclo de $CL = 8000$. RT y ϵ se calculan mediante un generador de números pseudo-aleatorio (PRNG) y se envían al receptor como el primer paquete cifrado; por lo tanto, estas dos variables son independientes a la clave del sistema.

1. Independencia estadística: El criptosistema propuesto produce ciphertext uniformemente distribuida independientemente de la distribución del plaintext, lo que confirma su robustez a ataques estadísticos. La distribución del ciphertext se ve influenciada por la distribución uniforme de P'(índices aleatorios en la Fig.5.1), que permite un acceso justo a todos los K mapas y sus correspondientes $TR = L/K$ trayectorias por mapa. El atacante no sabe si dos trayectorias consecutivas recuperados de la LUT provienen de diferentes mapas, el mismo mapa, y su orden temporal, por lo que es difícil de dirigir un ataque a un mapa predefinido. La independencia estdística entre el plaintext y el ciphertext es analizado a través del coeficiente de correlación, indicando valores despreciables (alrededor de cero) para todos los formatos de medios como muestran en la tabla IV (C1 vs texto plano).

La distancia media de Hamming $\left(\frac{1}{M}\left[\sum_{n=1}^{M} HD(C_n, C_{n+1})\right]\right)$ entre la transformación de plaintext-ciphertext y entre ciphertext adyacentes (comparación de los ciphetexts i e $i+1$) muestran que ~ 50

Capítulo 7

Conclusiones

conclusiones

Capítulo 8

Trabajos futuros

Anexo A

Anexos

A.1. Entrevista para obtener requerimientos

A.2. Diagramas de actividades

A.3. Pantallas del SIADA

A.4. Diagramas de secuencia

Anexo B

Glosario

Bibliografía

[1] Edmar Mota García, Rogelio Hasimoto Beltrán (2006), "Estudio sobre la dinámica del internet en méxico" ,Comunicación Técnica No I-06-04/27-01-2006 (CC/CIMAT)

[2] Manuel José Lucena López (2002), "Criptografía y seguridad en computadoras, España

[3] P. Caballero Gil, C. Hernadez (2002), "Criptografía y seguridad de la información, Editorial RA-MA, España.

[4] Kocarev,L.,Galias,Z.,Lian,S.(2009), "Intelligent Computing Based on chao Studies in Computational Intelligence", vol.184. Springer, Heidelberg

[5] Hirsch, M. W., Smale, S. (1974) , "Differential Equations, Dynamical Systems and Linear Algebra", Academic Press, NewYork.

[6] Feigenbaum M.J., (1980)," The Metric Universal Properties of Period Doubling Bifurcations and the Spectrum for a Route to Turbulence". New York. Acad. Sci. 357, 330 - 336

[7] F.Al-Masalha, R. Hasimoto, A. Khokhar (2010), "Performance Evaluation of Different Encryption Schemes on Portable and Mobile Platforms", 1^{st} International Green Computing Conference (IGCC-2010).

[8] R. Hasimoto Beltrán, "High-performance multimedia encryption system based on chaos, Chaos 18:023110, 2008.

[9] D.J. Bernstein, P. Schwable, "New AES Software Speed Recors". INDOCRYPT-2008, LNCS 5365:322-336,2008.

[10] P. M. Binder, R. V. Jensen. "Simulating chaotic behavior with finite-state machines". Phys. Rev. A 34(5), 4460-4463, 1986.

[11] J. Cernák, "Digital generators of chaos". Phys Lett A 214(3): 151-60, 1996.

[12] G. Alvarez and S. Li, "some basic cryptographic requirements for chaos-based encryption". Int. J. Bifurcation Chaos Appl. Sci. Eng. 16(8):2129-2151, 2006.

[13] S. Li, X. Mou, Z. Ji, J. Zhan, Y. Cai, Z. Ji, and J. Zhang, "On the security of a chaotic encryption scheme: problems with computerized chaos". Comput. Phys. Commun. 153(1):52-58, 2003.

[14] C. Li, S. Li, G. Alvarez, G. Chen, KT. Lo, "Cryptanalysis of a chaotic block cipher with external key and its improved version". Chaos, solitons & Fractals, 2008.

[15] R. Rhouma, S. Belghith, "Cryptanalysis of a spatiotemporal chaotic image/video cryptosystem". Phys. Lett A 372:5790:5794,2008.

[16] G. Alvarez, F. Montoya, M. Romera, G. Pastor, "Cryptanalysis of dynamic look-up table based chaotic cryptosystems". Phys Lett A, 326: 211-218, 2004.

[17] R. Rhouma, E. Solak, S. Belghith, "Cryptanalysis of a new substitution-diffusion based image cipher". Commun Nonlinear Sci Numer Simulat, 15(7), 2010.

[18] E. Solak, R. Rhouma, S. Belghith, "Cryptanalysis of a multi-chaotic streams based image cryptosystem". Optics Communications 283(2), 2010.

[19] B. Cistea, P. Chargé, D. Founier-Prunaret, F. Peyard, J. Mercier, "Behavior of chaotic sequences under finite representation and its cryptofraphic application". IEEE Workshop on Nonlinear Maps and Applications (NOMA'07), Toulouse, France, 2007.

[20] H. Hu, Y. Xu, Z. Zhu, "A method of improving the properties of digital chaotic system". Chaos, Solutions and Fractals,2008.

[21] Z. Elhadj, J.C. Sprott, "The effect of modulating a parameter in the logistic map", Chaos, 2008.

[22] A. Palacios, H. Juárez, "Cryptography with cycling chaos". Phys. Lett A, 303, 2002.

[23] X. Wang, M. Zhan, X. Gong, C.H. Lai, "Construction of a secure cryptosystem based on spatiotemporal chaos and its applications in public channel cyptography. http://arxiv.org/abs/nlin/0502026, 2005

[24] R. Hasimoto-Beltrán, "A generalized chaotic encryption system for multimedia applicatios". Revista mexicana de física 53(5), 2007.

[25] R. Rhouma, S. Meherzi, S. Belghith, "OCML- based colour image encryption". Chaos soliton and Fractals, 2007.

[26] N.K. Pareek, V Patidar, K.K. Sud, "Cryptography using multiple one-dimensional chaotic maps". Commun Nonlinear Set Numer Simulat, 2005.

[27] Y. Mao, G. Chen, S. Lian, "A novel fast image encryption scheme based on 3D chaotic baker map". Int J Biffurcation and chaos, 2004.

[28] J. Fridrich, "Symmetric ciphers based on two-dimensional chaotic map". Int J. Bifurcation Chaos, 1998.

[29] T. Sang, R. Wang, Y. Yan, "Perturbance-based algorithm to expand cycle length chaotic key stream", Electron Lett. 23, 1998.

[30] J. Cernak, "Digital generators of chaos", Phys Lett A, 1996.

[31] X. Tong, M. Cui, "Image encryption scheme based on 3D baker with dynamical compound chaotic sequence cipher generator", Signal Proccessing, 2009.

[32] N.K. Pareek, V. Patidar, K.K. Sud, "Discrete chaotic cryptography using external ker, Phys Leet A, 2003.

[33] M.S Baptista, "Cryptography with chaos". Phys. Lett A,240, 1998.

[34] G.Jakimoski and L. Kocarev, "Analysis of some recently proposed chaos-based encryption algorithms". Phys. Lett A, 291, 2001.

[35] G. Alvarez, F. Montoya, M. Romera, G. Pastor, "Cryptanalysis of an ergodic chaotic cipher. Phys. Lett.A, 311, 2003.

[36] S. Li, G. Chen, K. Wong, X. Mou, Y Cai, "Baptista's chaotic cryptosystems: Problems and countermeasures. Phys. Lett A,332, 2004.

[37] K. Wong, "A fast chaotic cryptographic scheme with dynamic look-up table". Phys. Lett A, 298, 2002.

[38] K. Wong, "A combined chaotic cryptographic and hashing". Phys. Lett. A, 307, 2003.

[39] M. R. K. Ariffin, M. S. M. Noorani, "Modified Baptista type chaotic cryptosystem via matrix secret key". Phys. Lett, 372, 2008.

[40] R. Rhouma, E. Solak, D. Arroyo, S. Li, G. Alvarez, S. Belghith, "Comment on Modified Baptista type chaotic cryptosystem via matrix secret key". Phys. Lett. A, 373, 2009.

[41] F. Huan, Z.H. Guan, "Cryptosystem using chaotic keys". Chaos, Solitons and Fractals, 2005.

[42] D. Formolo, L.P.L Oliveira, "A competitive searching-based chaotic cipher", Int. J. Mod. Phys. 21, 2010.

[43] S. Li. Chen, X. Mou, "On the degradation of digital piecewise linear chaotic map. Int. J. of Bif. Chaos, 15, 2005.

[44] S. El Assad, H Noura, I. Taralova, "Design and analysis of efficient chaotic generators for cryptosystems", Adv. in Electr. and Electron. Eng-IAENG, Special edition of the World Congress on Engineering and Computer Science 2008.