Cifrado de imágenes médicas mediante barajado de alta velocidad y difusión adaptativa de píxeles

Realizado por

Ricardo Carreño Mariño,

Carlos García Martínez,

Pablo Olivencia Moreno,

Antonio Daniel Porcar Aragón

Criptografía

Ingeniería Informática: Ingeniería del Software

20 de diciembre

Curso 2024/25

Resumen

Este proyecto aborda la necesidad de garantizar la seguridad de las imágenes médicas digitales, cuya confidencialidad y protección son fundamentales debido a la naturaleza sensible de los datos que contienen. La propuesta se centra en el desarrollo de un sistema de cifrado innovador y eficiente que combina múltiples técnicas avanzadas para transformar estas imágenes en representaciones cifradas, asegurando que solo puedan ser descifradas con la clave correcta.

El esquema de cifrado propuesto se basa en una red de sustitución y permutación, que integra tres componentes principales: la inserción de datos aleatorios, el mezclado rápido de píxeles y la difusión adaptativa por píxeles. La inserción de datos aleatorios garantiza que incluso utilizando la misma clave, cada ejecución del cifrado genere resultados únicos, mejorando significativamente la resistencia a ataques como el de texto plano elegido. El mezclado rápido de píxeles desorganiza sus posiciones, reduciendo las correlaciones entre píxeles vecinos y mejorando la confusión. Por último, la difusión adaptativa por píxeles extiende los efectos de cambios menores en la imagen original a lo largo de toda la imagen cifrada, asegurando una alta robustez frente a modificaciones.

El sistema desarrollado se basa en una variante que utiliza aritmética modular para optimizar el rendimiento en software. Este diseño ha sido concebido para adaptarse a los entornos tecnológicos actuales, garantizando flexibilidad y facilidad de integración en sistemas médicos existentes.

Se plantea que esta variante ofrece niveles excepcionales de seguridad, incluyendo alta sensibilidad a las claves y resistencia a ataques diferenciales y de texto plano. Además, su diseño asegura que las imágenes cifradas mantendrán su integridad incluso en presencia de ruido o pérdida parcial de datos. Asimismo, se espera que los tiempos de cifrado y descifrado sean competitivos frente a los métodos tradicionales, lo que la hace viable para su aplicación en imágenes de alta resolución y grandes volúmenes de datos.

Este proyecto presenta un avance significativo en la protección de imágenes médicas, proporcionando una solución que combina innovación, flexibilidad y robustez. Su diseño adaptable permite una integración directa con sistemas existentes en el ámbito médico, contribuyendo a fortalecer la privacidad de los pacientes y la seguridad de los datos en un contexto donde la información es cada vez más vulnerable a amenazas externas.

Índice general

1. Introducción

La creciente digitalización en el ámbito médico ha transformado la manera en que se gestionan y almacenan los datos clínicos, especialmente las imágenes médicas, que son herramientas fundamentales para el diagnóstico y tratamiento de enfermedades. Sin embargo, esta evolución también ha generado preocupaciones críticas relacionadas con la seguridad y privacidad de estos datos, dada su naturaleza sensible y su potencial vulnerabilidad frente a accesos no autorizados. La exposición indebida de imágenes médicas puede tener consecuencias graves, no solo en términos de violación de la privacidad de los pacientes, sino también en posibles usos maliciosos, como fraudes en seguros o manipulación de resultados médicos.

En este contexto, la criptografía emerge como una solución esencial para garantizar la protección de las imágenes médicas. Los sistemas de cifrado transforman las imágenes originales en formatos ininteligibles para cualquier individuo o entidad no autorizada, asegurando que solo aquellos con la clave correcta puedan acceder a los datos. A pesar de los avances en este campo, muchos de los esquemas de cifrado existentes presentan limitaciones en términos de eficiencia, robustez frente a ataques y adaptabilidad a diferentes formatos de imágenes. Además, las crecientes capacidades de cómputo y las mejoras en las técnicas de criptanálisis han puesto en evidencia vulnerabilidades en algunos métodos tradicionales, especialmente en aquellos basados en sistemas caóticos con baja entropía o estructuras complejas que ralentizan el proceso de cifrado.

Este trabajo se enmarca en la necesidad de desarrollar soluciones de cifrado específicas para imágenes médicas que sean seguras, eficientes y robustas. El enfoque se centra en diseñar un sistema capaz de proteger estas imágenes frente a ataques como el de texto plano conocido o elegido, así como garantizar su integridad en condiciones adversas, como ruido o pérdida parcial de datos. Asimismo, se busca optimizar el rendimiento del esquema para que pueda ser implementado de manera efectiva tanto en plataformas de hardware como de software, facilitando su integración en entornos clínicos reales donde el tiempo y la precisión son críticos.

En esta línea, el proyecto propone un esquema de cifrado innovador, basado en técnicas avanzadas como la inserción de datos aleatorios, el mezclado rápido de píxeles y la difusión adaptativa por píxeles, complementado con dos implementaciones específicas: una optimizada para hardware y otra para software. Estas características permiten abordar los retos actuales en la protección de imágenes médicas, proporcionando un sistema flexible y robusto que responde a las demandas de seguridad y eficiencia del sector. Este documento detalla el diseño, desarrollo y evaluación del sistema propuesto, posicionándolo como una contribución relevante en el campo de la criptografía aplicada a la medicina.

2. Descripción del problema

El problema principal que aborda el documento está relacionado con la seguridad y privacidad de las imágenes médicas en un contexto de creciente digitalización en el sector sanitario. Las imágenes médicas, fundamentales para el diagnóstico y tratamiento de enfermedades, contienen información altamente sensible y confidencial sobre los pacientes. Su exposición o uso indebido puede tener graves consecuencias, tanto a nivel personal como institucional. Estas imágenes pueden ser blanco de accesos no autorizados, ya sea por ataques cibernéticos o por el mal uso por parte de administradores de bases de datos o personal no autorizado. Esto podría derivar en actividades fraudulentas, como reclamaciones de seguros ilegítimas, explotación comercial o manipulación de información médica, generando riesgos significativos para la seguridad y bienestar de los pacientes.

Además, los sistemas actuales de almacenamiento y transmisión de imágenes médicas están sujetos a desafíos técnicos, como la corrupción de datos debido al ruido, pérdida parcial de la información durante la transmisión o almacenamiento, y el uso de algoritmos de cifrado que no son capaces de resistir los avances en las capacidades de cómputo y las técnicas de criptanálisis. A menudo, los métodos tradicionales de cifrado no están diseñados para adaptarse a la complejidad y diversidad de los formatos de imágenes médicas, lo que limita su eficacia y aplicabilidad. Esto incluye desde imágenes en formato DICOM, ampliamente utilizado en equipos de radiología, hasta otros formatos como NIFTI o MINC, empleados en investigaciones clínicas y sistemas de diagnóstico avanzados.

El documento identifica estas brechas y propone un sistema de cifrado que no solo protege las imágenes médicas frente a accesos no autorizados, sino que también responde a problemas como la adaptabilidad a distintos formatos, la robustez frente a ruido o pérdida de datos, y la eficiencia computacional necesaria para su uso en entornos clínicos. En esencia, el problema radica en la necesidad de un sistema de protección que combine alta seguridad con la capacidad de manejar grandes volúmenes de datos complejos de manera rápida y confiable, minimizando riesgos y asegurando la privacidad de los pacientes en un entorno donde la información es cada vez más vulnerable. Este desafío técnico y ético es el núcleo del problema al que el documento busca dar solución.

3. Solución: Esquema de encripción de imagenes médicas

3.1. Distribución de la clave

El proceso de distribución de la clave en el esquema de cifrado se basa en la generación de números pseudoaleatorios mediante un sistema conocido como **Logistic-Sine System (LSS)**. Este sistema es fundamental para garantizar propiedades como la alta aleatoriedad y la sensibilidad a las condiciones iniciales, aspectos clave para la seguridad criptográfica.

La clave secreta K se utiliza para derivar los estados iniciales y los parámetros de control necesarios en dos rondas de cifrado: una para el *scrambling* (mezcla de posiciones de píxeles) y otra para la difusión adaptativa de píxeles. La estructura de K incluye las siguientes componentes:

- x_0 y r: números decimales obtenidos de secuencias binarias de 52 bits.
- R_1 y R_2: constantes también derivadas de 52 bits.
- d_1 y d_2: números enteros obtenidos de secuencias de 24 bits.

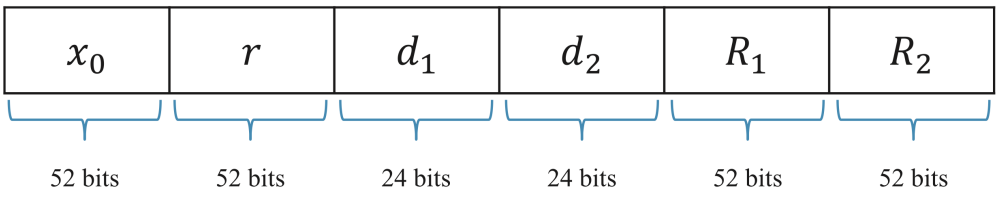

Figura 3.1: Esquema de la clave.

La conversión de secuencias binarias a números flotantes o enteros se realiza mediante las siguientes expresiones:

$$F_N = \sum_{i=1}^{52} Bin_i \times 2^{-i},$$

$$I_N = \sum_{i=1}^{24} Bin_i \times 2^{i-1}.$$

3.1.1. El Logistic-Sine System (LSS) como generador de números pseudoaleatorios

El *Logistic-Sine System* (LSS) es un método innovador utilizado para la generación de números pseudoaleatorios. Este sistema combina las propiedades de dos mapas

caóticos ampliamente estudiados: el mapa logístico y la función seno, logrando así una mayor complejidad en las secuencias generadas. El LSS está definido por la siguiente ecuación:

$$X_{n+1} = (rX_n(1 - X_n) + (4 - r)\sin(\pi X_n)/4) \quad \text{mód } 1,$$

donde X_n representa la variable de iteración y r es el parámetro de control, el cual puede tomar valores en el intervalo $[0, 4]$ y el resultado es un decimal encontrado entre $[0, 1]$. Esta fórmula produce valores acotados dentro del rango $(0, 1)$, lo que asegura un comportamiento caótico altamente dependiente de las condiciones iniciales. Este tipo de dinámica resulta ideal para aplicaciones que requieren alta aleatoriedad, como la criptografía y la simulación.

Comportamiento caótico del LSS

Una de las características más importantes del LSS es su comportamiento caótico. Este se manifiesta en su alta sensibilidad a las pequeñas variaciones en el valor inicial X_0 o en el parámetro de control r producen secuencias completamente diferentes. Este fenómeno es conocido como la dependencia extrema de las condiciones iniciales, un rasgo distintivo de los sistemas caóticos. Además, debido a la combinación de términos no lineales $(rX_n(1 - X_n))$ y el efecto de la función seno $((4 - r)\sin(\pi X_n)/4)$, el sistema logra complejidad adicional, dificultando la predicción de sus resultados. (Hua y cols., 2019)

Ciclo y calidad de los números generados

El LSS garantiza que los números generados tengan ciclos largos antes de repetir valores, lo que es crucial para aplicaciones que requieren secuencias extensas sin patrones detectables. Este atributo, conocido como largo periodo de periodicidad, asegura que las secuencias sean adecuadas para escenarios donde la repetición podría comprometer la integridad del proceso.

Además, la calidad de los números pseudoaleatorios generados se mide por su entropía y uniformidad dentro del intervalo $(0, 1)$. Al cumplir con estos criterios, el LSS minimiza la aparición de correlaciones no deseadas, mejorando la utilidad de las secuencias en tareas que dependen de una distribución aparentemente aleatoria.

Transformación y eficiencia computacional

Los valores generados por el LSS, que están restringidos al intervalo $(0, 1)$, pueden transformarse fácilmente para cumplir con distintos formatos o rangos requeridos por aplicaciones específicas. Por ejemplo, se pueden convertir en números enteros dentro de un rango dado, o incluso a representaciones binarias, según las necesidades del sistema.

Otra ventaja significativa del LSS es su eficiencia computacional. A pesar de la complejidad matemática de su definición, su implementación en hardware o software

es relativamente sencilla y rápida. Esto lo hace ideal para dispositivos con recursos limitados, como sistemas embebidos, donde la eficiencia y la seguridad son requisitos fundamentales.

Importancia del parámetro r

El parámetro r juega un papel crucial en el comportamiento del LSS. Cuando r se encuentra en ciertas regiones del intervalo $[0, 4]$, el sistema muestra un comportamiento caótico ideal para la generación de números pseudoaleatorios. Sin embargo, si r se selecciona fuera de estos valores, el sistema puede entrar en regiones de estabilidad donde la calidad de la aleatoriedad se ve comprometida. Por esta razón, la elección adecuada de r es un factor determinante en el diseño y uso del LSS.

3.1.2. Cálculo de estados iniciales

Los estados iniciales para las dos rondas de cifrado se calculan de la siguiente manera:

$$X_i^0 = d_i \times (x_0 + R_i) \mod 1,$$
$$r_i = d_i \times (r + R_i) \mod 4,$$

donde $i = 1, 2, ...$ corresponde a cada ronda. Estos estados iniciales únicos garantizan que las operaciones de mezcla y difusión dependan directamente de la clave secreta K, asegurando una alta sensibilidad a cualquier cambio en sus valores.

3.2. Inserción de Datos Aleatorios

La *inserción de datos aleatorios* es la primera fase del proceso una operación que tiene como objetivo incrementar la seguridad del proceso de cifrado al agregar ruido a la imagen original. Este ruido, generado de manera aleatoria, se inserta en los bordes de la imagen, lo que introduce variabilidad en cada ejecución del cifrado, incluso si se utiliza la misma imagen y la misma clave secreta. De esta manera, la imagen cifrada resultante será diferente en cada caso, lo que dificulta su análisis y recuperación sin la clave adecuada.

La operación se realiza en una imagen P de tamaño $H \times W$, donde H representa la altura (height) y W la anchura (width). Se generan dos vectores aleatorios:

- R, de tamaño $2 \times W$.
- O, de tamaño $(H + 2) \times 2$.

Los valores de estos vectores siguen el mismo formato de datos que los píxeles de la imagen original. La inserción de los datos aleatorios se lleva a cabo de la siguiente forma:

1. Las dos filas del vector R se insertan en la parte superior e inferior de la imagen.

2. Las dos columnas del vector O se insertan en el lado izquierdo y derecho de la imagen.

Este proceso introduce ruido en los bordes de la imagen, lo que influye en las fases posteriores de mezcla y difusión. La principal ventaja de este enfoque es que, dado que los datos aleatorios cambian en cada ejecución, la imagen cifrada varía aún cuando se utilicen los mismos parámetros (como la clave secreta). Esto incrementa la seguridad, ya que cada operación de cifrado produce un resultado único.

Un aspecto clave de este enfoque es que tanto la mezcla como la difusión, influenciadas por los datos aleatorios, son operaciones reversibles. Esto significa que, con la clave adecuada, es posible recuperar la imagen original de manera exacta, sin que los datos aleatorios insertados persistan en la imagen descifrada. Así, aunque la inserción de datos aleatorios aumenta la complejidad del cifrado, no afecta la capacidad de recuperar la imagen original, manteniendo la reversibilidad del proceso.

3.3. *High-Speed Scrambling*

El *high-speed scrambling* (barajado de alta velocidad) es una técnica fundamental en la criptografía de imágenes que tiene como objetivo alterar de forma eficiente las posiciones de los píxeles dentro de la imagen. Este proceso permite reducir la fuerte correlación entre los píxeles adyacentes, lo que mejora la seguridad del cifrado. El *high-speed scrambling* utiliza el Sistema Logístico-Seno (LSS), previamente explicado para generar las matrices necesarias en la encripción del sistema.

3.3.1. Generación de la matriz de barajado S

Supongamos que la imagen a cifrar tiene un tamaño de $M \times N$. Primero, se generan dos vectores aleatorios, A de longitud M y B de longitud N, utilizando el generador de números pseudoaleatorios basado en el LSS. Estos vectores se generan a partir del estado inicial (X_0, r_0) derivado de la clave secreta.

Una vez generados, los vectores A y B son ordenados para obtener dos vectores de índice, I y J. Estos índices se utilizan para reordenar las filas de una matriz de tamaño $M \times N$, y a continuación, las filas se desplazan según los valores de I y J, generando finalmente la matriz de barajado S.

3.3.2. Cifrado de la imagen utilizando S

Una vez que se ha generado la matriz S, los píxeles de la imagen se reordenan utilizando las columnas de la matriz S. Cada columna de S determina cómo deben moverse los píxeles de la imagen. Este proceso se describe mediante una bijección que mapea las posiciones de los píxeles de la imagen original P a sus nuevas posiciones en la imagen cifrada T.

El algoritmo de barajado de alta velocidad sigue los siguientes pasos:

1. Inicializar el índice de columna $j = 1$.
2. Encontrar los píxeles de la imagen P con las posiciones $(1, S_{1,j}), (2, S_{2,j}), \ldots, (M, S_{M,j})$.
3. Conectar estos píxeles en un círculo y desplazarlos $S_{1,j}$ celdas hacia arriba.
4. Repetir los pasos 2 y 3 para cada columna $j = 2, \ldots, N$.

Este proceso asegura que los píxeles estén completamente desordenados, lo que dificulta la tarea de descifrar la imagen sin la clave adecuada.

3.3.3. Relación con el LSS

La utilización del LSS en la generación de las matrices de barajado aporta un nivel significativo de aleatoriedad al proceso de cifrado. Como se explicó anteriormente, el LSS genera secuencias caóticas que dependen de las condiciones iniciales, lo que garantiza que, incluso si se utiliza la misma clave, el cifrado de la imagen será diferente en cada ejecución. Esto refuerza la seguridad del proceso, asegurando que no se repitan patrones en las imágenes cifradas, incluso cuando se cifren imágenes idénticas con la misma clave secreta.

3.4. Difusión Adaptativa de Píxeles

La difusión adaptativa de píxeles es una técnica esencial para fortalecer la seguridad de los esquemas de cifrado de imágenes. Este método asegura que cualquier cambio pequeño en la imagen original (texto plano) se propague de manera significativa a lo largo de todos los píxeles de la imagen cifrada (texto cifrado). Esto se logra mediante la combinación de valores pseudoaleatorios y valores de píxeles vecinos, resultando en una propagación no lineal de las variaciones iniciales.

3.4.1. Descripción del Algoritmo

El proceso de difusión adaptativa de píxeles se basa en modificar el valor actual de un píxel utilizando el valor del píxel anterior y un valor pseudoaleatorio generado. Este enfoque permite que los cambios en cualquier parte de la imagen original afecten a todos los píxeles de la imagen cifrada, distribuyendo uniformemente cualquier variación o alteración.

El método incluye dos implementaciones específicas:

- **XOR a Nivel de Bits (BX):** Utiliza la operación XOR bitwise para modificar los valores de los píxeles. Este método es altamente eficiente en plataformas de hardware debido a la simplicidad y rapidez de la operación XOR en componentes electrónicos.
- **Aritmética Módulo (MA):** Emplea operaciones de suma y módulo para calcular los valores de los píxeles. Esta implementación está diseñada para plataformas

de software, donde se logra un rendimiento óptimo gracias a la eficiencia computacional de estas operaciones en sistemas basados en CPU.

La fórmula general para la difusión adaptativa de píxeles en el esquema basado en XOR es:

$$C_{i,j} = \begin{cases} T_{i,j} \oplus Q_{i,j} & \text{si } i = 1, j = 1 \\ T_{i,j} \oplus C_{i,j-1} \oplus Q_{i,j} & \text{si } i = 1, j \neq 1 \\ T_{i,j} \oplus C_{i-1,j} \oplus Q_{i,j} & \text{si } i \neq 1 \end{cases}$$

En el esquema basado en la aritmética módulo, la fórmula se expresa como:

$$C_{i,j} = \begin{cases} (T_{i,j} + Q_{i,j}) \mod F & \text{si } i = 1, j = 1 \\ (T_{i,j} + C_{i,j-1} + Q_{i,j}) \mod F & \text{si } i = 1, j \neq 1 \\ (T_{i,j} + C_{i-1,j} + Q_{i,j}) \mod F & \text{si } i \neq 1 \end{cases}$$

donde $T_{i,j}$ representa el valor del píxel después del reordenamiento, $Q_{i,j}$ es un valor pseudoaleatorio generado, $\oplus$ denota la operación XOR, y F indica el número total de niveles de intensidad, que típicamente es 256 para imágenes de 8 bits.

3.4.2. Proceso de Descifrado y Limitaciones Prácticas

La operación inversa de la ecuación de cifrado se define de acuerdo con las siguientes condiciones:

$$T_{i,j} = \begin{cases} C_{i,j} \oplus Q_{i,j}, & \text{si } i = 1, j = 1, \\ C_{i,j} \oplus C_{i,j-1} \oplus Q_{i,j}, & \text{si } i = 1, j \neq 1, \\ C_{i,j} \oplus C_{i-1,j} \oplus Q_{i,j}, & \text{si } i \neq 1. \end{cases}$$

$$T_{i,j} = \begin{cases} (C_{i,j} - T_{M,N} - Q_{i,j}) \mod F, & \text{si } i = 1, j = 1, \\ (C_{i,j} - C_{i,j-1} - Q_{i,j}) \mod F, & \text{si } i = 1, j \neq 1, \\ (C_{i,j} - C_{i-1,j} - Q_{i,j}) \mod F, & \text{si } i \neq 1. \end{cases}$$

El orden de las operaciones en el proceso de descifrado es opuesto al proceso de cifrado. En teoría, MIE-BX puede cifrar imágenes de cualquier tamaño. Sin embargo, en aplicaciones prácticas, debido a las limitaciones de almacenamiento en implementaciones de hardware, cuando el tamaño de la imagen es demasiado grande y supera la capacidad de almacenamiento, MIE-BX no puede cifrar directamente la imagen. En estos casos, MIE-BX divide primero la imagen en bloques más pequeños y luego cifra cada uno de estos bloques por separado.

3.4.3. Propiedades y Beneficios

El diseño del algoritmo de difusión adaptativa de píxeles ofrece varias ventajas significativas:

- **Propagación Total de Cambios:** Cualquier modificación en un píxel de la imagen original se extiende a todos los píxeles de la imagen cifrada, asegurando un alto grado de confusión y difusión.
- **Versatilidad:** El algoritmo puede aplicarse a imágenes de diferentes profundidades de color y formatos, lo que lo hace adecuado para una amplia variedad de aplicaciones en cifrado de imágenes médicas y no médicas.
- **Optimización para Hardware y Software:** Las dos implementaciones propuestas permiten adaptarse a diferentes entornos, maximizando la eficiencia en función de los recursos disponibles.
- **Robustez:** La estructura del algoritmo permite recuperar la imagen original incluso en escenarios donde la imagen cifrada ha sufrido pérdida de datos o ruido.

3.4.4. Importancia en Aplicaciones Prácticas

La difusión adaptativa de píxeles desempeña un papel crucial en la protección de imágenes médicas, donde la privacidad de los datos es de suma importancia. En estos contextos, la capacidad de cifrar imágenes de alta resolución y formatos variados con un nivel elevado de seguridad es esencial para cumplir con las regulaciones de privacidad y garantizar la integridad de los datos. Además, la robustez del esquema frente a ruido y pérdida de datos lo hace ideal para transmisiones en redes inseguras o almacenamiento en dispositivos con capacidad limitada.

3.5. Evaluación de Seguridad

3.5.1. Seguridad de la Clave

El espacio de claves de MIE-BX y MIE-MA es 2^{256}, proporcionando una capacidad adecuada para resistir ataques de fuerza bruta. Además, las claves deben ser extremadamente sensibles. Si las claves no son lo suficientemente sensibles, claves ligeramente diferentes podrían reconstruir correctamente la imagen original, reduciendo el espacio de claves efectivo.

La sensibilidad de las claves se evalúa utilizando la tasa de cambio de bits (NBCR, por sus siglas en inglés), definida como:

$$\mathrm{NBCR}(B_1, B_2) = \frac{\mathrm{Ham}(B_1, B_2)}{\mathrm{Len}},$$

donde $\mathrm{Ham}(B_1, B_2)$ es la distancia de Hamming entre dos imágenes B_1 y B_2 (la distancia de Hamming mide el número de posiciones en las que difieren dos cadenas de

igual longitud. Por ejemplo, para las cadenas `1010` y `1001`, la distancia de Hamming es 2, ya que difieren en dos posiciones), y Len es el número total de bits de las imágenes. Un NBCR cercano al 50 % indica que las imágenes son completamente diferentes. Los experimentos muestran que al cambiar cualquier bit en una clave secreta de 256 bits, las imágenes cifradas y descifradas resultantes son completamente diferentes, demostrando la alta sensibilidad de las claves de MIE-BX y MIE-MA.

3.5.2. Resistencia a Ataques por Texto Plano Escogido

Un algoritmo de cifrado con un alto nivel de seguridad debe resistir ataques por texto plano escogido. En MIE-BX y MIE-MA, se inserta ruido aleatorio en los bordes de la imagen antes de realizar el cifrado, garantizando que al cifrar una misma imagen dos veces con la misma clave, los cifrados resultantes sean completamente diferentes. Esto dificulta que los atacantes encuentren correlaciones entre textos planos y cifrados, resistiendo eficazmente ataques por texto plano escogido y conocido.

3.5.3. Resistencia a Ataques Diferenciales

La resistencia a ataques diferenciales se mide mediante la tasa de cambio de píxeles (NPCR) y la intensidad media de cambio uniforme (UACI), que evalúan cómo las diferencias en los textos planos afectan a los cifrados. Los experimentos realizados con 20 imágenes médicas muestran que los valores de NPCR y UACI de MIE-BX y MIE-MA están muy cerca de los valores esperados (99.9985 % y 33.3338 %, respectivamente), indicando una fuerte resistencia a ataques diferenciales.

4. Demostración en código

4.1. Descripción del Funcionamiento del Sistema de Cifrado

El sistema de cifrado ha sido implementado en Python, utilizando Streamlit para la interfaz gráfica y varias bibliotecas para las operaciones de cifrado y manipulación de imágenes. El archivo principal del programa, `app.py`, gestiona el flujo principal de la aplicación y hace uso de módulos auxiliares para tareas específicas como el cifrado, descifrado y la anonimización de datos médicos.

4.1.1. Descripción General

El sistema está diseñado para procesar imágenes médicas en formato DICOM, realizar su cifrado mediante scrambling y difusión adaptativa, y permitir su descifrado con la clave correcta. La estructura modular y el uso de `Streamlit` facilitan su uso interactivo. Los pasos principales son los siguientes:

1. Carga de la imagen DICOM y normalización de sus datos.
2. Generación de claves criptográficas mediante un generador basado en hash256.
3. Cálculo de las claves de ronda.
4. Aplicación del cifrado mediante técnicas de scrambling y difusión adaptativa.
5. Descifrado de la imagen cifrada utilizando las claves correctas.
6. Visualización de las imágenes y resultados.

4.1.2. Estructura del Código

A continuación, se describen las principales partes del código.

Importaciones Principales

El archivo comienza con las importaciones necesarias, que incluyen:

- `numpy` y `matplotlib.pyplot`: Para manipulación de datos y visualización.
- `streamlit`: Para crear una interfaz gráfica interactiva.
- `pydicom`: Para procesar imágenes médicas en formato DICOM.
- Módulos personalizados como `padding`, `cypher2`, `key_gen`, y `anonimize`, que encapsulan funcionalidades específicas.

```
import numpy as np
import matplotlib.pyplot as plt
import streamlit as st
import pydicom
from util.padding import insertar_datos_aleatorios, quitar_datos_aleatorios
from util.v2.cypher2 import encrypt_image, decrypt_image
from util.key_gen import generate_round_keys
from util.anonimize import save_dicom
```

Cifrado de Imágenes

La función principal para el cifrado se define utilizando un decorador de `Streamlit` para mejorar el rendimiento mediante caché. Los pasos son:

1. Se genera una clave utilizando `generate_round_keys`.
2. Se aplican técnicas de scrambling y difusión mediante la función `encrypt_image`.

A partir de la clave generada, obtenemos las matrices de scrambling y difusión para realizar el cifrado.

```
@st.cache_data
def cifrar_imagen(image, clave_cifrado, num_rounds):
    claves = generate_round_keys(clave_cifrado)
    imagen_cifrada, params = encrypt_image(image, claves, num_rounds)
    return imagen_cifrada, params
```

Descifrado de Imágenes

La función para el descifrado invierte el proceso de cifrado utilizando la clave introducida, volviendo a generar las claves de ronda y matrices de scrambling y difusión.

```
@st.cache_data
def descifrar_imagen(image, clave_descifrado, params, num_rounds):
    claves = generate_round_keys(clave_descifrado)
    imagen_descifrada = decrypt_image(image, claves, params, num_rounds)
    return imagen_descifrada
```

Interfaz Gráfica

La interfaz gráfica se construye con `Streamlit`, permitiendo la carga de imágenes y visualización de resultados de forma interactiva.

```
st.title("Cifrado y Descifrado de Imágenes Médicas")
image_file = st.file_uploader("Sube una imagen DICOM")
if image_file is not None:
    ds = pydicom.dcmread(image_file)
    image = ds.pixel_array
```

```
...
st.image(image, caption="Imagen Original")
```

4.1.3. Ventajas del Sistema

El sistema de cifrado implementado presenta varias ventajas:

- **Interactividad**: La interfaz gráfica basada en Streamlit es fácil de usar y entender.
- **Modularidad**: La separación en módulos permite realizar mejoras o adaptaciones fácilmente.
- **Seguridad**: Utiliza técnicas avanzadas como scrambling y difusión adaptativa, garantizando la confidencialidad de los datos.

4.1.4. Resultados Esperados

El sistema produce una imagen cifrada que es completamente irreconocible sin la clave correcta. Una vez descifrada, la imagen original se recupera con total precisión, asegurando tanto la integridad como la confidencialidad de los datos médicos.

El repositorio del proyecto puede encontrarse en Moreno (2024) y la aplicación está abierta para su uso en Grupo14 (2024)

Mencionar que la imagen en formato DICOM de la que disponemos es bastante pesada, y la aplicación no dispone de muchos recursos, por lo que el proceso puede ser lento. Su ejecución en local puede aumentar drásticamente el rendimiento; si bien el algoritmo presenta bastante espacio para su mejora mediante la paralelización de procesos. Actualmente, la generación de claves, matrices de scrambling y difusión se realizan de forma secuencial, así como la difusión, que es en gran parte paralelizable.

5. Tabla de tiempos

En esta sección se muestran las tareas realizadas con sus respectivas duraciones.

Description	User	Duration (h)
Explicacion de codigo + finalizacion del documento	Ricardo Carreño Mariño	0:18:15
presentacion	Carlos García Martínez	0:17:06
Limpieza y arreglo de la documentacion + implementacion	Ricardo Carreño Mariño	1:06:10
presentacion	Carlos García Martínez	1:06:26
presentacion	Carlos García Martínez	0:41:02
Clase 3/12	Pablo Olivencia Moreno	1:31:25
clase 3/12/2024	Carlos García Martínez	1:50:00
Trabajo clase 3/12: Presentación y ajuste en el overleaf (detectamos fallo en el paper)	Antonio Daniel Porcar Aragón	1:50:00
Desarrollo de la app streamlit: hacer que pida imagen cifrada y su clave para correcto descifrado	Antonio Daniel Porcar Aragón	1:30:00
Desarrollo de la app streamlit: subir imagen introduciendo clave y da imagen cifrada, y si pones la de descifrado te descifra, si es incorrecta no lo hace bien	Antonio Daniel Porcar Aragón	2:30:00
memoria	Carlos García Martínez	0:53:00
clase 28/11/2024	Carlos García Martínez	1:50:00
Trabajo clase 28/11: Ajustar algoritmo para que funcione exactamente igual que en el paper	Antonio Daniel Porcar Aragón	1:50:00
clase 26/11/2024	Carlos García Martínez	1:29:23
Clase 26/11	Ricardo Carreño Mariño	1:30:30
Trabajo clase 26/11	Antonio Daniel Porcar Aragón	1:31:16
Clase 26/11	Pablo Olivencia Moreno	1:40:20

Figura 5.1: Resumen de tiempos de ejecución.

Los documentos originales pueden consultarse en Mariño (2024).

6. Referencias

Grupo14. (2024). *Aplicación trabajo criptografía.* https://criptog-14.streamlit.app. (Aplicación desplegada del proyecto)

Hua, Z., Jin, L., Huang, H., y Huang, Y. (2019). 2d logistic-modulated-sine-coupling-logistic chaotic map for image encryption. *IEEE Transactions on Industrial Electronics*, *67*(6), 5039–5049. Descargado de https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8977567 doi: 10.1109/TIE.2019.2959128

Mariño, R. C. (2024). *Documentos originales en carpeta compartida.* https://uses0-my.sharepoint.com/:f:/g/personal/riccarmar_alum_us_es/EjVzNJx9-x9KvqMtQUNPcwwBXZWNr_sIdRBl1fsbgcDhbA?e=eFjdvo. (Acceso a documentos originales de tiempos.)

Moreno, P. O. (2024). *Repositorio trabajo criptografía.* https://github.com/pabolimor99/CriptoG-14.git. (Repositorio del proyecto)