Artículo original/ Original article
ISSN 2953-6553

Protección de datos mediante el uso de técnicas de criptografía para garantizar la confidencialidad

Data protection through the use of cryptographic techniques to guarantee confidentiality

Marcelo Monteros-Guerrero [1] 0000-0002-8095-8109, Bryam Durazno-Chumbay [1] 0009-0002-3137-8842, Diego Cherres-Yuqilima [1] 0009-0008-4318-8564
ruben.monteros@tecazuay.edu.ec, bryam.durazno.est@tecazuay.edu.ec, diego.cherres.est@tecazuay.edu.ec

[1] Instituto Superior Universitario Tecnológico del Azuay, Cuenca, Ecuador

DOI 10.36500/atenas.2.003

Resumen

Resumen: En la era de la digitalización, es fundamental garantizar la confidencialidad de la información en todas las etapas de su ciclo de vida, desde su creación hasta su uso y destrucción. Es importante considerar que gran parte de la información es confidencial durante un período de tiempo determinado, por lo que se debe garantizar su acceso y uso restringido exclusivamente al personal autorizado.
En este contexto, este trabajo se enfoca en la propuesta de implementar un software que utiliza una técnica criptográfica probada para asegurar la confidencialidad de los datos. Esta solución ofrece una alternativa para la transferencia segura de datos o información entre dispositivos, permitiendo que dichos datos sean transportados de manera confidencial, sin importar el medio utilizado.
Después de un análisis exhaustivo, se ha seleccionado el algoritmo criptográfico AES. Este algoritmo ha demostrado ser uno de los más seguros y ha sido implementado utilizando el lenguaje Python con la librería cryptodome. La elección de este algoritmo se basa en investigaciones que respaldan su fiabilidad y robustez.
Finalmente, este trabajo aborda la necesidad imperante de proteger la confidencialidad de la información en un entorno digital. Propone la implementación de un software que utiliza el algoritmo criptográfico AES para garantizar la seguridad de los datos, permitiendo su transporte confidencial entre dispositivos.

Abstract

Abstract: In the era of digitization, it is essential to guarantee the confidentiality of information at all stages of its life cycle, from its creation to its use and destruction. It is important to consider that much of the information is confidential for a certain period of time, so its access and restricted use must be guaranteed exclusively to authorized personnel.
In this context, this work focuses on the proposal to implement software that uses a proven cryptographic technique to ensure data confidentiality. This solution offers an alternative for the secure transfer of data or information between devices, allowing such data to be transported confidentially, regardless of the medium used.
After extensive analysis, the AES cryptographic algorithm has been selected. This algorithm has proven to be one of the most secure and has been implemented using the Python language with the cryptodome library. The choice of this algorithm is based on research supporting its reliability and robustness.
Finally, this paper addresses the prevailing need to protect the confidentiality of information in a digital environment. It proposes the implementation of a software that uses the AES cryptographic algorithm to guarantee the security of the data, allowing its confidential transport between devices.

Palabras Claves – AES, Criptografía, confidencialidad.
Keywords – AES, Cryptography, confidentiality.

Recibido: 2023-05-05, Aprobado tras revisión: 2023-11-10

I. INTRODUCCIÓN

En la actualidad, "la información es un activo extremadamente valioso para las empresas, ya sea interna o de los clientes" (Trevenque, 2018). Sin embargo, en la era digital en la que nos encontramos; los datos, tanto personales como empresariales están expuestos más que nunca a la exfiltración o filtración, y los atacantes pueden ser empleados como personas externas a las organizaciones.

Por otra parte, la Ley Orgánica de Protección de Datos Personales (LOPD) tiene como objetivo proteger el derecho constitucional de las personas a la protección de sus datos personales (Asamblea Nacional, 2021). Si una empresa sufre una violación de seguridad y no ha tomado medidas adecuadas para proteger su información, puede ser objeto de sanciones. Además, es obligada a informar a sus clientes sobre la violación de seguridad (Naciones Unidas, 2012). Esto puede resultar en pérdidas económicas y de imagen.

Con lo antes indicado, la seguridad de la información se ha convertido en un elemento clave para el funcionamiento de las empresas, puesto que todas manejan datos para realizar sus actividades y necesitan garantizar su protección e integridad (AyudaLey, 2020). Para garantizar la seguridad de la información, es esencial aplicar los tres pilares fundamentales de la seguridad de la información: confidencialidad, integridad y disponibilidad. En este sentido, en el presente artículo se propone el uso de un algoritmo criptográfico para garantizar la confidencialidad de los datos almacenados en un medio físico, como puede ser un disco duro o flash memory, lo cual facilita el manejo y transporte de la información confidencial de forma segura dificultando el acceso no autorizado. Este software fue realizado utilizando Visual Studio Code y Python como el entorno de desarrollo integrado y el lenguaje de programación con sus respectivas librerías.

II. MARCO TEÓRICO

La criptografía es una técnica que se utiliza para proteger la información de accesos no autorizados y garantizar la confidencialidad. Consiste en la transformación de datos en un formato ilegible para quienes no tienen la clave de descifrado, lo que dificulta su lectura y comprensión (Sáez, 2022). “Esta técnica se ha utilizado durante siglos para proteger información importante en distintos ámbitos, como la política, la guerra, la diplomacia y el comercio” (Velasco, 2014). Actualmente, se aplica en el mundo digital para proteger la privacidad y la seguridad de la información que se intercambia en línea. Se analiza los siguientes tipos de cifrado simétrico:

A. Cifrado AES

Definición:

El cifrado AES es un algoritmo simétrico que utiliza bloques de datos de 128 bits y claves de cifrado con longitudes variables de 128, 192 o 256 bits. Implementa un conjunto de operaciones criptográficas, como sustituciones no lineales, permutaciones y mezclas de datos, para garantizar un nivel elevado de seguridad y resistencia frente a ataques criptoanalíticos. Según el Instituto Nacional de Normas y Tecnología (NIST), "El algoritmo AES especifica un cifrado de bloque simétrico, que opera en bloques de datos de 128 bits, utilizando claves de cifrado con longitudes de 128, 192 o 256 bits" (NIST, 2023).

Diseño:

El diseño del AES se fundamenta en una red de sustitución-permutación (SPN) ampliamente adoptada en algoritmos de cifrado. Esta estructura combina las operaciones de sustitución y permutación para incrementar la seguridad y resistencia ante ataques criptoanalíticos. La red de sustitución-permutación del AES se compone de varias rondas de transformaciones criptográficas. En cada ronda, se aplican las siguientes operaciones:

- **Sustitución de bytes:** Se reemplaza cada byte del bloque de datos utilizando una tabla de sustitución llamada S-Box, lo cual aporta no linealidad al cifrado.
- **Permutación de filas:** Se reorganizan las posiciones de las filas en el bloque de datos para dispersar la información y evitar patrones predecibles.
- **Mezcla de columnas:** Se mezclan linealmente las columnas del bloque de datos mediante una operación matemática llamada mezcla de columnas, lo cual aumenta la difusión de los datos en el cifrado.
- **Adición de clave:** En cada ronda, se realiza una operación XOR entre el bloque de datos y una subclave derivada de la clave original, lo cual mezcla la clave con los datos y añade una capa de seguridad adicional (Stallings, 2014).

Estas operaciones se repiten en varias rondas, dependiendo de la longitud de la clave utilizada 10 rondas para claves de 128 bits, 12 rondas para claves de 192 bits y 14 rondas para claves de 256 bits, lo que aumenta la complejidad y la seguridad del cifrado. "El diseño del AES destaca por su elegancia y simplicidad, lo que facilita su implementación eficiente en hardware y software." (Rijmen, 1999)

Fortalezas del cifrado AES:

- Seguridad comprobada: "AES ha resistido el escrutinio de la comunidad criptográfica y ha demostrado ser seguro contra los ataques más avanzados conocidos hasta la fecha." (NIST, 2023).
- Amplia adopción: "AES es ampliamente utilizado en todo el mundo y se ha convertido en un estándar de facto para el cifrado de datos." (Alliance, 2016).
- Eficiencia y rendimiento: "AES se ha optimizado y se puede implementar de manera eficiente en hardware y software, lo que lo hace adecuado para una amplia gama de dispositivos y aplicaciones." (NIST, 2023) .

Limitaciones del cifrado AES:

- Vulnerabilidad potencial a ataques cuánticos: "AES, al igual que otros algoritmos criptográficos de clave simétrica, podría verse comprometido por computadoras cuánticas en el futuro si no se utilizan claves lo suficientemente largas." (Kumar, 2022).
- Implementación incorrecta: "Los ataques a AES no siempre se dirigen directamente al algoritmo en sí, sino a la implementación del algoritmo en sistemas específicos. Una implementación incorrecta o débil puede introducir vulnerabilidades." (Institute, 2002)
- Gestión de claves: "La seguridad de AES depende en gran medida de la fortaleza y la gestión adecuada de las claves utilizadas. Una clave débil o una gestión inadecuada de las claves puede debilitar la seguridad del cifrado." (National Cyber Security, 2021).

B. Cifrado 3DES

Definición

El cifrado 3DES (Triple Data Encryption Standard) es un algoritmo de cifrado simétrico basado en el estándar DES. Utiliza tres rondas de cifrado DES para mayor seguridad. Cada ronda utiliza una clave de cifrado de 56 bits, al igual que el DES. Se aplican operaciones de sustitución, permutación y mezcla para transformar y difundir los datos de forma no lineal, agregando complejidad al cifrado. (Barker & Mouha, 2017)

Diseño

El cifrado 3DES se basa en la combinación de tres rondas de cifrado del estándar DES y utiliza permutaciones como parte de su proceso de cifrado. Estas permutaciones reorganizan los datos antes y después de las etapas de sustitución y mezcla, dispersando la información y añadiendo complejidad. La permutación de los datos se logra mediante la reordenación de las posiciones de los bits en el bloque de datos, evitando patrones predecibles (Barker & Mouha, 2017). El diseño del 3DES también incluye permutaciones en las etapas de generación de claves, asegurando una efectiva mezcla y distribución de

las claves. La combinación de las operaciones de sustitución, mezcla y permutación fortalece la seguridad del 3DES y dificulta la recuperación de la información sin las claves adecuadas. Sin embargo, debido a su velocidad de procesamiento lenta y la existencia de algoritmos más eficientes como el AES, se está reevaluando su uso continuo (Barker & Mouha, 2017).

Fortalezas/limitaciones

La fortaleza del cifrado 3DES radica en la repetición del algoritmo DES tres veces consecutivas. Esto incrementa significativamente la longitud de clave efectiva a 168 bits y proporciona una mayor resistencia a los ataques criptoanalíticos (Miranda & Medina, 2015). La complejidad resultante hace que sea extremadamente difícil para los atacantes encontrar debilidades o descifrar los datos cifrados.

Fortaleza

- Es un algoritmo de cifrado muy seguro.
- Es ampliamente compatible con hardware y software.
- Es relativamente fácil de implementar.

Limitaciones

- Es más lento que otros algoritmos de encriptación.
- No es tan eficiente como otros algoritmos de encriptación.
- No se considera tan seguro como algunos algoritmos de cifrado más nuevos, como AES.

Actualmente, se está reconsiderando el uso continuo del 3DES debido a su velocidad de procesamiento comparativamente más lenta y la aparición de algoritmos más modernos y eficientes, como el AES. (Barker & Mouha, 2017)

C. Cifrado DES

Definición

El DES es un sistema criptográfico basado en la estructura de red Feistel (Anshel, 2012), que opera con bloques de información de 64 bits. Utiliza una clave de 64 bits, donde cada octavo bit se reserva para la paridad. El proceso de cifrado del DES implica una permutación inicial del texto en claro, seguido de 16 rondas de una función de tipo Feistel utilizando subclaves generadas. Para el descifrado, se sigue un procedimiento similar, pero con las subclaves en orden descendente. En resumen, el DES utiliza la estructura de red Feistel y opera sobre bloques de 64 bits con clave de 64 bits, realizando permutaciones y rondas de funciones para cifrar y descifrar los datos. (Bernal & Alejandro, 2017)

Diseño:

El cifrado DES sigue un diseño basado en la estructura de red Feistel. En resumen, se puede describir de la siguiente manera:

- Permutaciones iniciales: Se aplica una permutación inicial al bloque de datos de entrada de 64 bits, reordenando los bits según una tabla predefinida.

- División en bloques: El bloque de datos se divide en dos partes iguales de 32 bits cada una, conocidas como "parte izquierda" y "parte derecha".

- Rondas de Feistel: Se llevan a cabo 16 rondas de operaciones en las partes izquierda y derecha del bloque de datos. Cada ronda incluye las siguientes operaciones:

a. Expansión: La parte derecha se expande de 32 a 48 bits mediante una tabla de expansión que duplica algunos bits.

b. XOR con la subclave: Se realiza una operación XOR entre la expansión y una subclave de 48 bits generada a partir de la clave de cifrado principal.

c. Sustitución: Los 48 bits resultantes se dividen en ocho grupos de 6 bits, y cada grupo se sustituye por 4 bits utilizando ocho S-Boxes (tablas de sustitución no lineal).

d. Permutación: Los 32 bits resultantes de la sustitución se permutan utilizando una tabla de permutación fija llamada P-Box.

e. XOR y cambio de bloques: La parte izquierda se somete a una operación XOR con la parte derecha, y luego se intercambian las partes izquierda y derecha para prepararse para la siguiente ronda.

f. Permutación final: Después de las 16 rondas de Feistel, se realiza una permutación final para reordenar los bits del bloque de datos cifrado.

g. Salida: El resultado final del cifrado DES es un bloque de datos cifrado de 64 bits (Stallings, 2014).

Fortalezas del cifrado DES:

- Amplia adopción: El cifrado DES ha sido ampliamente adoptado y utilizado en una variedad de aplicaciones y sistemas durante varias décadas.

- Diseño basado en Feistel: El cifrado DES sigue un diseño basado en la estructura de red Feistel, que ha demostrado ser resistente a diversos ataques criptográficos.

- Longitud de clave efectiva: Aunque la longitud de clave del DES es de 56 bits, la repetición del algoritmo en tres rondas aumenta la longitud de clave efectiva a 168 bits, lo que proporciona un nivel razonable de seguridad.

- Resistencia a ataques conocidos: A lo largo de los años, el cifrado DES ha demostrado ser resistente a muchos ataques criptoanalíticos conocidos, lo que ha contribuido a su reputación como un algoritmo seguro (Stallings, 2014).

Limitaciones del cifrado DES:

- Longitud de clave corta: La longitud de clave de 56 bits del DES se considera insuficiente en la actualidad, ya que los avances en tecnología y computación han permitido técnicas de fuerza bruta más eficientes para descifrar claves de esta longitud.

- Vulnerabilidad a ataques de fuerza bruta: Aunque el cifrado DES es resistente a muchos ataques conocidos, es vulnerable a los ataques de fuerza bruta, donde todas las combinaciones posibles de claves se prueban exhaustivamente hasta encontrar la clave correcta.
- Ritmo de procesamiento lento: Comparado con algoritmos más modernos, como AES, el DES tiene un ritmo de procesamiento relativamente lento, lo que puede limitar su uso en aplicaciones que requieren un cifrado rápido y eficiente.
- Avances en criptoanálisis: A medida que avanza el criptoanálisis y se descubren nuevos métodos y técnicas, el cifrado DES puede verse más vulnerable a ataques más sofisticados y avanzados (Stallings, 2014).

D. *Definición del problema*

Una vez analizado los distintos algoritmos de cifrado, se deduce la aplicación del algoritmo AES. Esto debido varias razones científicas y técnicas que lo hacen altamente seguro y confiable. Aquí están algunas de las razones principales:

1. Resistencia a ataques criptoanalíticos: AES ha sido diseñado para resistir ataques criptoanalíticos conocidos, como el ataque de texto claro elegido, el ataque de texto cifrado elegido y el ataque diferencial. Estos ataques intentan explotar debilidades en el algoritmo para descifrar el mensaje sin conocer la clave (Gordillo, 2022).

2. Comprobada seguridad: AES ha sido ampliamente estudiado y analizado por expertos en criptografía de todo el mundo. Ha pasado por rigurosas pruebas y revisiones públicas y ha demostrado ser seguro en términos de confidencialidad de datos (De la Torre, 2016)

3. Eficiencia y rendimiento: AES es un algoritmo de cifrado eficiente en términos de tiempo de ejecución y uso de recursos computacionales. Puede cifrar y descifrar datos de manera rápida y

efectiva, lo que lo hace adecuado para su uso en una amplia gama de aplicaciones, incluyendo la protección de datos en tiempo real (HID Global Corporation, 2021).

4. Amplia adopción: AES es un estándar ampliamente aceptado y utilizado tanto en el ámbito gubernamental como en el comercial. Su implementación está disponible en una variedad de plataformas y lenguajes de programación, lo que facilita su integración en sistemas y aplicaciones (De la Torre, 2016).

5. Tamaño de clave flexible: AES admite claves de diferentes tamaños, lo que permite adaptarse a requisitos específicos de seguridad. Puede utilizar claves de 128, 192 o 256 bits, lo que proporciona una mayor flexibilidad para garantizar la confidencialidad de los datos. (Gordillo, 2022)

En general, considerando lo analizado el algoritmo AES tiene solidez matemática, seguridad probada y eficiencia en el rendimiento, lo que lo convierte en una elección confiable a ser utilizada en el proyecto de cifra de datos almacenados.

E. Comparación de los tres tipos de cifrado

Considerando lo analizado en la Tabla 1 se expone la comparación de los tres tipos de algoritmos de cifrado. Proporciona las siguientes informaciones:

- Factores
- AES
- 3DES
- DES

Tabla 1.

Comparación ente los cifrados AES, 3DES y DES.

FACTORES	AES	3DES	DES
Longitud de clave	128, 192 y 256 bits	3 K. de 56 bits c/u	56 bits
Tamaño de bloque	128, 192 y 256 bits	64 bits	64 bits
Año de Desarrollo	2000	1978	1977
Resiste criptoanálisis	SI	NO	NO
Seguridad	Seguro	Mediana	Inseguro
Rondas	10 (128 b), 12(192 b), 14(256b)	48	8
Rendimiento	4.174/6.452	3.45/5665	4.01/6.347
Tipo de clave	Sola	Sola - partida en 3	Sola

Nota: Después de exponer las características principales de los 3 tipos de cifrado, se puede deducir que el mejor es AES.

III. METODOLOGÍA Y CÁLCULOS

Luego de analizar los tipos de cifrados para la protección de la información y garantizar la seguridad digital se concluyó en implementar la técnica criptográfica mediante el cifrado AES. La misma garantizará un proceso eficiente y eficaz que cumple con los requerimientos para el proyecto y brinda una solución de calidad en términos de seguridad y protección de la información. Mismos que se describen técnicamente a continuación:

1) Cifrado de datos utilizando el algoritmo criptográfico AES en Python

Este algoritmo criptográfico se desarrolló en Visual Studio Code como el entorno de desarrollo integrado y el lenguaje de programación Python con sus respectivas librerías. Inicialmente utilizamos la biblioteca **os,** es una biblioteca incorporada de Python que permite la manipulación de archivos,

directorios y variables de entorno. En este caso **os.path.join()** es útil para trabajar con rutas de archivos y directorios de una forma segura en diferentes sistemas operativos. Asegura que la ruta resultante sea válida y que esté formateada de manera consistente (Python, 2023).

Seguidamente la importación de AES desde Crypto.Cipher es una forma de acceder a la implementación del algoritmo de cifrado AES en Python utilizando la biblioteca PyCryptodome (o Crypto). La biblioteca también admite diferentes modos de operación, como CBC (cadena de bloques de cifrado) y GCM (Modo Galois/Contador) (KeepCoding , 2023). La figura 1 “Modo de operación GCM” si indica el código para cargar el archivo.

Figura 1.

Modo de operación GCM

```
nombre_Ar_Cf=nombre_Arch_or
ruta_archivo_original = os.path.join(file_path, nombre_Arch_or)
nonce = get_random_bytes(12)
cipher = AES.new(key, AES.MODE_GCM, nonce=nonce)
with open(ruta_archivo_original, 'rb') as file:
```

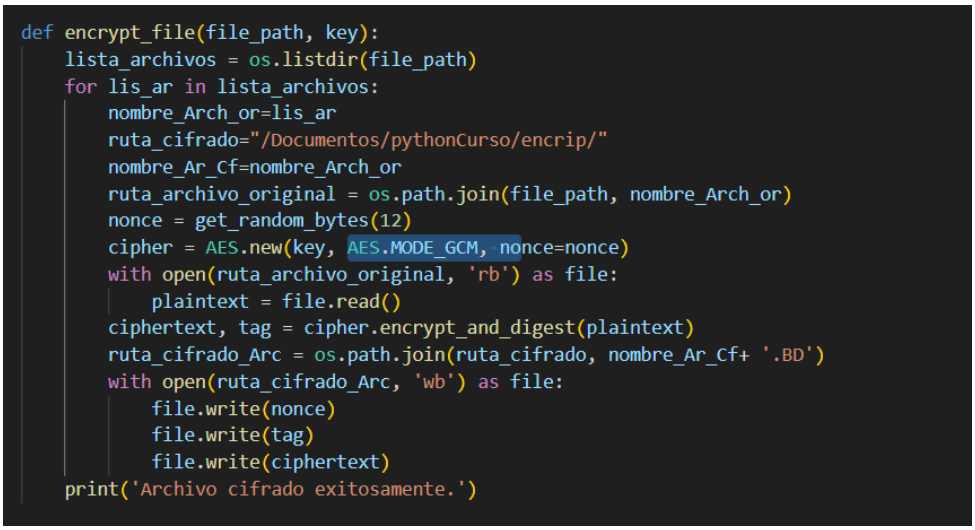
Nota: Modo de operación GCM” si indica el código para cargar el archivo. 2023

La biblioteca os y Cryptodome es ampliamente utilizada en la industria y cuenta con una buena reputación en términos de seguridad. Sin embargo, como cualquier biblioteca de seguridad, es importante usarla correctamente y seguir las mejores prácticas de seguridad al implementar soluciones de seguridad en el código de Python (Moreno, 2023).

La figura 2 “Código de cifrado AES” indica el código utilizado en la implementación.

Figura 2.

Código de cifrado AES



```python
def encrypt_file(file_path, key):
    lista_archivos = os.listdir(file_path)
    for lis_ar in lista_archivos:
        nombre_Arch_or=lis_ar
        ruta_cifrado="/Documentos/pythonCurso/encrip/"
        nombre_Ar_Cf=nombre_Arch_or
        ruta_archivo_original = os.path.join(file_path, nombre_Arch_or)
        nonce = get_random_bytes(12)
        cipher = AES.new(key, AES.MODE_GCM, nonce=nonce)
        with open(ruta_archivo_original, 'rb') as file:
            plaintext = file.read()
        ciphertext, tag = cipher.encrypt_and_digest(plaintext)
        ruta_cifrado_Arc = os.path.join(ruta_cifrado, nombre_Ar_Cf+ '.BD')
        with open(ruta_cifrado_Arc, 'wb') as file:
            file.write(nonce)
            file.write(tag)
            file.write(ciphertext)
    print('Archivo cifrado exitosamente.')
```

Nota: "Código de cifrado AES" indica el código utilizado en la implementación. 2023

Este código realiza la encriptación de archivos, se detalla una explicación sencilla de lo que hace. La función tomará dos argumentos: file_path, que es una cadena que representa la ruta del directorio que contiene los archivos a cifrar, y key, que es la clave secreta utilizada para el cifrado, posteriormente obtiene una lista de todos los archivos en el directorio especificado usando el os.listdir()método. Luego itera sobre cada archivo y realiza el proceso de cifrado en cada archivo:

1. Carga el contenido del archivo original usando el open()método y lo lee como bytes.
2. Genera un nonce único aleatorio de 12 bytes.
3. Crea un nuevo cifrado AES con la clave y el nonce especificados.
4. Cifra el texto sin formato usando el encrypt_and_digest()método de cifrado, que devuelve el texto cifrado y una etiqueta que se puede usar para verificar la autenticidad del mensaje.
5. Escribe el nonce, la etiqueta y el texto cifrado en un nuevo archivo con .BDextensión.

Cuando se procesan todos los archivos, la función imprime un mensaje que indica la finalización exitosa del cifrado.

2) Descifrado de datos utilizando el algoritmo criptográfico AES en Python

En la figura 3 **"**Código de Descifrado AES**"** se muestra el código de descifrado

Figura 3

Código de Descifrado AES

```python
def decrypt_file(file_path2, key):
    lista_archivos_des = os.listdir(file_path2)
    for lis_ar_des in lista_archivos_des:
        nombre_Arch_or_des=lis_ar_des
        ruta_descifrado="/Documentos/pythonCurso/decrip/"
        nombre_Ar_des=nombre_Arch_or_des
        ruta_archivo_original_des = os.path.join(file_path2, nombre_Arch_or_des)
        with open(ruta_archivo_original_des, 'rb') as file:
            nonce = file.read(12)
            tag = file.read(16)
            ciphertext = file.read()
        cipher = AES.new(key, AES.MODE_GCM, nonce=nonce)
        plaintext = cipher.decrypt_and_verify(ciphertext, tag)
        ruta_cifrado_Arc_des = os.path.join(ruta_descifrado, nombre_Ar_des[:-3])
        with open(ruta_cifrado_Arc_des, 'wb') as file:
            file.write(plaintext)
    print('Archivo descifrado exitosamente.')
```

Nota: En la figura 3 **"**Código de Descifrado AES**"** se muestra el código de descifrado

Este código realiza la desencriptación de archivos en un directorio específico utilizando el algoritmo de encriptación. Se detalla una explicación sencilla de lo que hace:

La función primero obtiene una lista de todos los archivos en el directorio especificado que tienen la .BDextensión usando el os.listdir()método. Luego itera sobre cada archivo y realiza el proceso de descifrado en cada archivo:

1. Carga el contenido del archivo cifrado utilizando el open()método y lee el nonce, la etiqueta y el texto cifrado como bytes.
2. Crea un nuevo cifrado AES con la clave y el nonce especificados.

3. Descifra el texto cifrado usando el decrypt_and_verify()método de cifrado, que devuelve el texto sin formato y verifica la autenticidad del mensaje usando la etiqueta.
4. Escribe el texto sin formato en un archivo nuevo con el nombre del archivo original sin la .BDextensión.

Cuando se procesan todos los archivos, la función imprime un mensaje que indica la finalización exitosa del descifrado. En la figura 4 "Código de la clave y directorio" se visualiza el uso de las claves y directorio a ser usado en la cifra y descifrado.

Figura 4

Código de la clave y directorio

```
# La clave debe tener 16, 24 o 32 bytes
key = b'clave_secreta_de_16_24_o_32_byte'
file_to_encrypt ='/Documentos/pruebascr_cri/'
encrypted= "/Documentos/pythonCurso/encrip/"
# Cifrar el archivo
encrypt_file(file_to_encrypt,key)
# Descifrar el archivo cifrado
decrypt_file(encrypted,key)
```

Nota: "Código de la clave y directorio" se visualiza el uso de las claves y directorio a ser usado en la cifra y descifrado. 2023

Este código toma una clave secreta y una ruta de directorio, y encripta los archivos en el directorio especificado utilizando la clave, guardando los archivos encriptados en otro directorio.

IV. RESULTADOS Y DISCUSIÓN

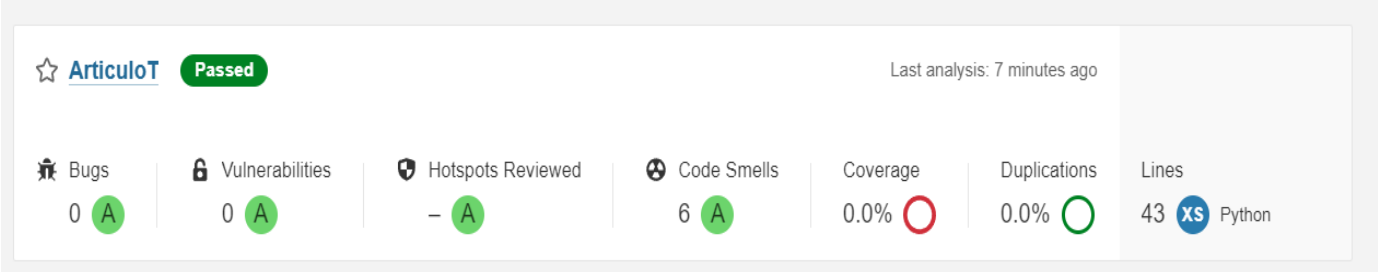
Durante la evaluación del código fuente mediante el método AES, se realiza un minucioso análisis para garantizar la correcta implementación y funcionamiento de este algoritmo de encriptación. En este proceso de evaluación, se somete el código fuente a diversas situaciones y casos de prueba con el fin de evaluar su desempeño, seguridad y la exactitud de los resultados obtenidos.

Se verifican aspectos como la generación y gestión adecuada de claves, la correcta aplicación del algoritmo AES en los datos de entrada, así como la descodificación precisa de los datos cifrados. Es relevante resaltar que el método AES es ampliamente utilizado en aplicaciones y sistemas que requieren un alto nivel de seguridad en la protección de datos sensibles. Por lo tanto, es fundamental asegurar que el código fuente implemente correctamente este algoritmo y cumpla con los estándares de seguridad establecidos. En las pruebas del código fuente, se emplean herramientas y técnicas específicas, como la generación de vectores de prueba, para abarcar una amplia gama de escenarios y situaciones de encriptación. Se evalúan aspectos como la confidencialidad, integridad y autenticidad de los datos cifrados, garantizando así que el algoritmo AES funcione de manera adecuada en todos estos aspectos.

Por otro lado, el código fue sometido a un análisis de vulnerabilidades utilizando la herramienta SonarQube, y el resultado ha sido positivo. La herramienta no ha detectado vulnerabilidades, lo que garantiza significativamente su calidad y seguridad, En la figura 5 "Resultados de SonarQube" se visualiza los resultados del análisis de vulnerabilidades.

Figura 4

Resultados de SonarQube.



Nota: "Resultados de SonarQube" se visualiza los resultados del análisis de vulnerabilidades. 2023

Finalmente, se llevan a cabo pruebas de rendimiento para evaluar la eficiencia del código fuente y su capacidad para manejar grandes volúmenes de datos. Esto implica medir el tiempo de encriptación

y desencriptación, así como el impacto en los recursos del sistema durante el proceso. En la Tabla 2 se detalla las pruebas de rendimiento del código al momento de cifrar y descifrar.

Tabla 2.

Rendimiento del cifrado y descifrado.

FACTORES	ESPECIFICACIONES
Cifrado	
Total, archivos	312
Tamaño de los archivos	462MB
Tiempo de cifrado	2 segundos
Descifrado	
Total, archivos	312
Tamaño de los archivos	462MB
Tiempo de Descifrado	2 segundos

Nota: Pruebas de rendimiento del código al momento de cifrar y descifrar

V. CONCLUSIONES

El algoritmo de cifrado funciona en el tiempo y con la eficacia esperada; sin embargo, se deberá continuar con la investigación de nuevos algoritmos que permitan mayor seguridad al intento de descifrado por fuerza bruta. Ya que esto depende de la actual capacidad de cómputo existente en la industria.

Adicional, existe el cifrado persistente por hardware embebido en las tarjetas madre de las PC, lo cual contribuye a la confidencialidad de la información lo cual es un nuevo ámbito para investigar.

REFERENCIAS BIBLIOGRÁFICAS

Bernal, T., Alejandro, D. (21 de septiembre de 2017). *Diseño e implementación de un cifrado DES extendido trenzado.* https://repository.libertadores.edu.co/bitstream/handle/11371/1438/tellezdiego2017.pdf?sequence=1&isAllowed=y

National Cyber Security. (21 de septiembre de 2021). *Password Guidance. Encryption Algorithms.* https://teampassword.com/blog/what-is-password-encryption-and-how-much-is-enough

Alliance, C. S. (2016). Las principales amenazas de la computación en la nube.

Anshel, I. (23 de mayo de 2012). *The Artin-Feistel Symmetric Cipher.* https://veridify.com/wp-content/uploads/2014/05/ArtinFeistelPaper.pdf

Asamblea Nacional. (2021). *Ley Orgánica de Protección de Datos*.https://www.finanzaspopulares.gob.ec/wpcontent/uploads/2021/07/ley_organica_de_proteccion_de_datos_personales.pdf

AyudaLey.(2020).*CategoríasdedatospersonalesenelRGPD*.https://ayudaleyprotecciondatos.es/category/lopdgdd-rgpd/

Barker,E.,yMouha,N.(2017).*RecommendationfortheTripleData*.https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-67r2.pdf

DelaTorre,J.(3deoctubrede2016).*Cifradodeclaveprivada*.https://core.ac.uk/download/pdf/84137053.pdf

Gordillo,E.(2022).*TécnicascriptográficasligerasparadispositivosIoT*.CentroUniversitariodelaDefensa.

HIDGlobalCorporation.(09dediciembrede2021).*Tresrazonesesenciales*.https://www.hidglobal.com/sites/default/files/resource_files/pacs-three-reasons-to-upgrade-eb-es.pdf

Institute,S.(2002).¿QuéeselcifradoAESycómofunciona?https://ciberseguridad.com/guias/prevencion-proteccion/criptografia/cifrado-aes/

KeepCoding.(14demarzode2023).*Modosdeoperacióndelcifradoporbloques*.https://keepcoding.io/blog/modos-de-operacion-del-cifrado-por-bloques/

Kumar,M.(5dejuliode2022).*Computacióncuánticaycriptografíaposcuántica*.https://dynatec.es/2022/05/07/criptografia-poscuantica-la-herramienta-para-combatir-los-ciberataques-informaticos-cuanticos/

Lofton.(09deseptiembrede2020).*Importanciadelaconfidencialidaddelainformación*.https://loftonsc.com/blog/juridico/juridico-laboral/confidencialidad-de-la-informacion/

Miranda,H.,yMedina,Y.T.(Juniode2015).*ComparacióndeAlgoritmosBasadosenlaCriptografía*.https://dialnet.unirioja.es/descarga/articulo/5286657.pdf

Moreno,R.(14deabrilde2023).*¿Cuálessonlasmejoreslibreríasenpythonpararealizaractividadesdeciberseguridadyciberdefensa?*.https://es.quora.com/Cu%C3%A1les-son-las-mejores-librer%C3%ADas-en-python-para-realizar-actividades-de-ciberseguridad-y-ciberdefensa

NacionesUnidas.(2012).*Organizaciónderechoshumanos*.https://www.ohchr.org/sites/default/files/Documents/Publications/HR.PUB.12.2_sp.pdf

NIST.(09demayode2023).*AdvancedEncryptionStandard(AES)*.https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.197-upd1.pdf

Python.(2023).*Interfacesmisceláneasdelsistemaoperativo*.https://docs.python.org/es/3.10/library/os.html

Python.(2023).*Manipulacionescomunesdenombrederuta*.https://docs.python.org/es/3.10/library/os.path.html

Rijmen,D.&.(1999).*AESProposal:Rijndael*.https://csrc.nist.gov/csrc/media/projects/cryptographic-standards-and-guidelines/documents/aes-development/rijndael-ammended.pdf

Sáez,J.(8deagostode2022).IEBSchool.*Quéeslacriptografíayparaquésirve*.https://www.iebschool.com/blog/que-es-la-criptografia-y-para-que-sirve-finanzas/

Stallings,W.(28deagostode2014).PrinciplesandPracticehttps://uomustansiriyah.edu.iq/media/lectures/6/6_2017_03_17!10_56_57_PM.pdf

Trevenque.(04deabrilde2018).¿Cómopuedeayudartenuestrodepartamentodeseguridad?https://www.trevenque.es/seguridad/departamento-seguridad/

Velasco,J.J.(20demayode2014).Brevehistoriadelacriptografía:https://www.eldiario.es/turing/criptografia/breve-historia-criptografia_1_4878763.html